



HAL
open science

Complexité de l'évaluation de plusieurs formes bilinéaires et des principaux calculs matriciels

Jean-Claude Lafon

► **To cite this version:**

Jean-Claude Lafon. Complexité de l'évaluation de plusieurs formes bilinéaires et des principaux calculs matriciels. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1976. tel-00287014

HAL Id: tel-00287014

<https://theses.hal.science/tel-00287014>

Submitted on 10 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

Université Scientifique et Médicale de Grenoble
Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR ES SCIENCES MATHÉMATIQUES

par

Jean-Claude LAFON



COMPLEXITE
DE L'EVALUATION DE PLUSIEURS FORMES BILINEAIRES
ET
DES PRINCIPAUX CALCULS MATRICIELS



Thèse soutenue le 29 novembre 1976 devant la Commission d'Examen

Président : B. MALGRANGE

Examineurs : J. CEA
N. GASTINEL
F. ROBERT
J. VUILLEMIN

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Monsieur Michel SOUTIF : Président

Monsieur Gabriel CAU : Vice-Président

MEMBRES DU CORPS ENSEIGNANTS DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des Fluides
	ARNAUD Paul	Chimie
	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
	BEAUDOING André	Clinique de Pédiatrie et Puériculture
	BERNARD Alain	Mathématiques Pures
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques Pures
	BEZES Henri	Pathologie Chirurgicale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLINET Louis	Informatique (I.U.T. B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique Ophtalmologique
	BONNET-EYMARD Joseph	Clinique Gastro-entérologique
Mme	BONNIER Marie-Jeanne	Chimie Générale
MM.	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques Appliquées
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique Rhumatologique et Hydrologique
	CALAS François	Anatomie
	CARLIER Georges	Biologie Végétale
	CARRAZ Gilbert	Biologie Animale et Pharmacodynamie
	CAU Gabriel	Médecine Légale et Toxicologie
	CAUQUIS Georges	Chimie Organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Clinique Oto-Rhino-Laryngologique
	CHATEAU Robert	Clinique de Neurologie
	CHIBON Pierre	Biologie Animale
	COEUR André	Pharmacie Chimique et Chimie Analytique
	CONTAMIN Robert	Clinique Gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière Médicale
MM.	DEBELMAS Jacques	Géologie Générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumo-Physiologie
	DEPORTES Charles	Chimie Minérale
	DESRE Pierre	Métallurgie

MM.	DESSAUX Georges	Physiologie Animale
	DODU Jacques	Mécanique Appliquée (I.U.T. A)
	DOLIQUE Jean-Michel	Physique des Plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	GAGNAIRE Didier	Chimie Physique
	GALLISSOT François	Mathématiques Pures
	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
	KUNTZMANN Jean	Mathématiques Appliquées
	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-Jean	Mathématiques Appliquées
	LEDRU Jean	Clinique Médicale B
	LLIBOUTRY Louis	Géophysique
	LOISEAUX Pierre	Sciences Nucléaires
	LONGEQUEUE Jean-Pierre	Physique Nucléaires
	LOUP Jean	Géographie
Melle	LUTZ Elisabeth	Mathématiques Pures
	MALGRANGE Bernard	Mathématiques Pures
	BOUTET DE MONVEL Louis	Mathématiques Pures
	MALINAS Yves	Clinique Obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MAZARE Yves	Clinique Médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MICOUD Max	Clinique Maladies Infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie Nucléaire
	MULLER Jean-Michel	Thérapeutique (Néphrologie)
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques Pures
	PEBAY-PEYROULA Jean-Claude	Physique
	RASSAT André	Chimie Systématique
	RENARD Michel	Thermodynamique
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-Chirurgie
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction Mécanique (I.U.T. A)
	SOUTIF Michel	Physique Générale
	TANCHE Maurice	Physiologie

MM.	TRAYNARD Philippe	Chimie Générale
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique Nucléaire
	VAUQUOIS Bernard	Calcul Electronique
Mme	VERAIN Alice	Pharmacie Galénique
MM.	VERAIN André	Physique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCCOZ Jean	Physique Nucléaire Théorique

PROFESSEURS ASSOCIES

MM.	CLARK Gilbert	Spectrométrie Physique
	CRABBE Pierre	CERMO
	ENGLMAN Robert	Spectrométrie Physique
	HOLTZBERG Frédéric	Basses Températures
	ROST Ernest	Sciences Nucléaires

PROFESSEURS SANS CHAIRE

Melle	AGNIUS-DELDORD Claudine	Physique Pharmaceutique
	ALARY Josette	Chimie Analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques Appliquées
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique (I.U.T. A)
	BUISSON René	Physique (I.U.T. A)
	CONTE René	Physique (I.U.T. A)
	DEPASSEL Roger	Mécanique des Fluides
	GAUTHIER Yves	Sciences Biologiques
	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.	KUHN Gérard	Physique (I.U.T. A)
	LE ROY Philippe	Mécanique (I.U.T. A)
	LUU DUC Cuong	Chimie Organique
	MAYNARD Roger	Physique du Solide
	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et Minéralogie
	PSISTER Jean-Claude	Physique du Solide
Melle	PIERY Yvette	Physiologie Animale
MM.	RAYNAUD Hervé	M.I.A.G.
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie Générale
	RICHARD Lucien	Biologie Végétale
Mme	RINAUDO Marguerite	Chimie Macromoléculaire
MM.	ROBERT André	Chimie Papetière

MM.	SARRAZIN Roger	Anatomie et Chirurgie
	SARROT-REYNAUD Jean	Géologie
	SIROT Louis	Chirurgie Générale
Mme	SOUTIF Jeanne	Physique Générale
MM.	STREGLITZ Paul	Anesthésiologie
	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques Appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	AMBLARD Pierre	Dermatologie
	ARMAND Gilbert	Géographie
	ARMAND Yves	Chimie (I.U.T. A)
	BACHELOT Yvan	Endocrinologie
	BARGE Michel	Neuro chirurgie
	BARJOLLE Michel	MIAG
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamie
MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (C.U.S.)
MM.	BRODEAU François	Mathématiques (I.U.T. B)
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et Organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie Papetière
	CHIAVERINA Jean	Biologie Appliquée (EFP)
	COHEN-ADDAD Jean-Pierre	Spectrométrie Physique
	COLOMB Maurice	Biochimie Médicale
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	CORDONNIER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	CYROT Michel	Physique du solide
	DELOBEL Claude	M.I.A.G.
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie Végétale
	DUSSAUD René	Mathématiques (C.U.S.)
Mme	ETERRADOSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine Légale
	FAURE Gilbert	Urologie
	FONTAINE Jean-Marc	Mathématiques Pures
	GAUTIER Robert	Chirurgie Générale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GROS Yves	Physique (I.U.T. A)
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	KOLODIE Lucien	Hématologie
	KRAKOWIAK Sacha	Mathématiques Appliquées
	LE NOC Pierre	Bactériologie-virologie
	LEROY Philippe	I.U.T. A
	MACHE Régis	Physiologie Végétale
	MAGNIN Robert	Hygiène et Médecine Préventive
	MALLION Jean-Michel	Médecine du Travail
	MARECHAL Jean	Mécanique (I.U.T. A)
	MARTIN-BOUYER Michel	Chimie (C.U.S.)

M.	MICHOULIER Jean	Physique (I.U.T. A)
Mme	MINIER Colette	Physique (I.U.T. A)
MM.	NEGRE Robert	Mécanique (I.U.T. A)
	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique (I.U.T.A)
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (I.U.T. B)
	PEFFEN René	Métallurgie (I.U.T. A)
	PERRET Jean	Neurologie
	PERRIER Guy	Géophysique - Glaciologie
	PHELIP Xavier	Rhumatologie
	RACHAIL Michel	Médecine Interne
	RACINET Claude	Gynécologie et Obstétrique
	RAMBAUD André	Hygiène et Hydrologie
	RAMBAUD Pierre	Pédiatrie
Mme	RENAUDET Jacqueline	Bactériologie
MM.	ROBERT Jean-Bernard	Chimie-Physique
	ROMIER Guy	Mathématiques (I.U.T. B)
	SHOM Jean-Claude	Chimie générale
	STOEBNER Pierre	Anatomie pathologique
	VROUSOS Constantin	Radiologie

MAITRE DE CONFERENCES ASSOCIES

M.	COLE Antony	Sciences Nucléaires
----	-------------	---------------------

CHARGE DE FONCTIONS DE MAITRE DE CONFERENCES

M.	JUNIEN-LAVILLAVROY Paul	O.R.L.
----	-------------------------	--------

Fait à SAINT MARTIN D'HERES,
DECEMBRE 1975.

Président : M. NEEL Louis
Vice-Présidents : M. BENOIT Jean
M. BONNETAIN Lucien

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSION Jean	Electrochimie
	BLOCH Daniel	Physique du solide
	BONNETAIN Lucien	Chimie Minérale
	BONNIER Etienne	Electrochimie et Electrometallurgie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	COUMES André	Radioélectricité
	FELICI Noël	Electrostatique
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie-Physique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	POLOUJADOFF Michel	Electrotechnique
	SILBERT Robert	Mécanique des Fluides

PROFESSEURS ASSOCIES

MM.	RUPPERSBERG Albert, Henner	Chimie
	ROUXEL Roland	Automatique

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BOUVARD Maurice	Génie Mécanique
	COHEN Joseph	Electrotechnique
	DURAND Francis	Métallurgie
	FOULARD Claude	Automatique
	LACOUME Jean-Louis	Géophysique
	LANCIA Roland	Electronique
	VEILLON Gérard	Informatique Fondamentale & Appliquée
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Mathématiques Appliquées
	BOUDOURIS Georges	Radioélectricité
	CHARTIER Germain	Electronique
	GUYOT Pierre	Chimie Minérale
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du solide
	MORET Roger	Electrotechnique Nucléaire
	PIERRARD Jean-Marie	Hydraulique
	ROBERT François	Analyse Numérique
	SABONNADIÈRE Jean-Claude	Informatique Fondamentale & Appliquée
Mme	SAUCIER Gabrièle	Informatique Fondamentale & Appliquée

MAITRE DE CONFERENCES ASSOCIE

M. LANDAU Ioan Automatique

CHERCHEURS DU C.N.R.S. (Directeurs et Maîtres de Recherche)

MM.	FRUCHART Robert	Directeur de Recherche
	ANSARA Ibrahim	Maître de Recherche
	CARRE René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Les travaux qui font l'objet de cette thèse ont été effectués à l'Institut de Mathématiques Appliquées de Grenoble. Je tiens à exprimer ma reconnaissance à tous ceux qui ont participé à la création et au développement de cet Institut. Ma profonde gratitude va à Monsieur le Professeur N. GASTINEL qui, en m'accueillant au sein de l'équipe d'analyse numérique m'a permis de travailler dans un cadre si favorable.

Monsieur le Professeur B. MALGRANGE a bien voulu s'intéresser à ce travail et accepter de présider le jury. Je le prie de trouver ici l'expression de ma respectueuse gratitude.

Monsieur le Professeur N. GASTINEL m'a proposé le sujet de ce travail. Pour son appui constant, pour ses encouragements, et pour tout le temps qu'il a bien voulu me consacrer, je lui exprime ma très vive reconnaissance.

Monsieur le Professeur J. CEA a continuellement suivi l'élaboration de cette thèse. Je le remercie aussi d'avoir accepté de venir à Grenoble faire partie du jury.

Je suis très reconnaissant à Monsieur VUILLEMIN, Maître de Conférences à Orsay, de s'être intéressé à ce travail et de s'être déplacé pour participer au jury.

J'exprime aussi ma gratitude à Monsieur le Professeur DEMAZURE pour l'attention et les suggestions apportées à cette étude.

A Monsieur le Professeur ROBERT, j'adresse tous mes remerciements pour avoir suivi avec intérêt, en maintes occasions, le déroulement de ce travail.

J'adresse enfin mes vifs remerciements à Madame Meyrieux pour le soin apporté à la dactylographie de ce document ainsi qu'à l'équipe du Service de Reproduction pour sa réalisation matérielle.

PLAN

	pages
 <u>PARTIE A</u> : GENERALITES	
I	Fondements théoriques de la complexité..... A 4
II	Discussion des principaux critères de coûts.... A 7
	Références..... A21
 <u>PARTIE B</u> : COMPLEXITE DU CALCUL DE PLUSIEURS FORMES BILINEAIRES	
	Plan détaillé de la partie B..... B 1
	Chapitre BI : Classe des algorithmes utilisés, Critère d'optimalité..... B 5
	Chapitre BII : Caractérisation du coût minimal et des algorithmes optimaux..... B17
	Chapitre BIII : Etude de la notion de rang tensoriel.. B28
	Chapitre BIV : Bases tensorielles. Applications..... B53
	Chapitre BV : Minoration du rang tensoriel. Résultats d'optimalité..... B95
 <u>PARTIE C</u> : ETUDE DES PRINCIPAUX CALCULS MATRICIELS POUR DIFFERENTS CRITERES DE COÛTS.	
	Plan détaillé de la partie C..... C 1
	Chapitre CI : Rappels sur les calculs matriciels.... C 5
	Chapitre CII : Applications des résultats de la partie B C26
	Chapitre CIII : Optimalité de quelques algorithmes basés sur l'emploi des matrices élémentaires C44
	Chapitre CIV : Utilisation optimale du parallélisme.. C70
	Chapitre CV : Influence de la pagination sur la rapidité d'exécution des algorithmes de calculs matriciels..... C88

TABLE DES MATIERES

	pages
INTRODUCTION GENERALE -----	1-6
PARTIE A : GENERALITES -----	A1
INTRODUCTION -----	A3
I - Fondements théoriques de la complexité -----	A4
II - Discussion des principaux critères de coûts -----	A7
1/ Problèmes de calculs algébriques -----	A7
2/ Complexité de processus itératifs -----	A11
3/ Un exemple de fonction coût mémoire :	
calculs avec des matrices creuses -----	A12
4/ Un cas de fonction coût logarithmique :	
calculs algébriques exacts -----	A15
5/ Utilisation du parallélisme dans les calculs -----	A18
REFERENCES SUR LA PARTIE A -----	A21-A35
 PARTIE B : COMPLEXITE DU CALCUL DE PLUSIEURS FORMES BILINEAIRES -	 B1
INTRODUCTION -----	B3
<u>Chapitre BI - Classe des algorithmes utilisés. Critères d'optimalité</u>	B5
1. Notations -----	B6
2. Calcul de p formes bilinéaires -----	B9
3. Algorithmes de calculs algébriques -----	B10
4. Critère de coût utilisé -----	B12
5. Algorithmes optimaux -----	B14
REFERENCES SUR LE CHAPITRE BI -----	B16
 <u>Chapitre BII - Caractérisation du coût minimal et des algorithmes</u>	
<u>optimaux</u> -----	B17
1. Coût minimal dans le cas commutatif -----	B18
2. Coût minimal dans le cas non commutatif -----	B21
3. Interprétations avec la notion de rang tensoriel -----	B22
4. Algorithmes de coût minimal -----	B24
REFERENCES SUR LE CHAPITRE BII -----	B27

	Pages
<u>Chapitre BIII - Etude de la notion de rang tensoriel</u> -----	B28
1. Principales propriétés -----	B29
2. Rang d'un tenseur -----	B35
3. Condition pour que p matrices régulières aient un rang tensoriel minimal -----	B37
4. Rang tensoriel d'un espace de matrices contenant au moins une matrice régulière -----	B41
5. Cas du produit de deux matrices -----	B45
6. Calcul approché du rang tensoriel -----	B49
REFERENCES SUR LE CHAPITRE BIII -----	B52
 <u>Chapitre BIV - Bases tensorielles. Applications</u> -----	 B53
1. Un théorème d'optimalité -----	B54
2. Espace des matrices cycliques -----	B55
a/ Unicité de la base tensorielle -----	B55
b/ Application au produit de convolution -----	B57
c/ Inversion d'une matrice cyclique -----	B58
3. Espace des matrices de Hankel (ou de Toeplitz) -----	B60
a/ Bases tensorielles -----	B60
b/ Application au calcul du produit de deux polynomes --	B64
c/ Inversion en $O(n)$ multiplications d'une matrice de Toeplitz triangulaire inférieure -----	B69
d/ Division de deux polynomes -----	B74
4. Espaces des matrices ayant des symétries particulières	
a/ Matrices symétriques -----	B79
b/ Matrices centrosymétriques -----	B83
c/ Matrices horizontales (verticales) symétriques -----	B88
d/ Matrices roto-symétriques droites (gauches) -----	B89
e/ Matrices à éléments complexes -----	B91
REFERENCES SUR LE CHAPITRE BIV -----	B93-B94

	Pages
<u>Chapitre BV - Minoration du rang tensoriel</u>	
<u>Résultats d'optimalité</u> -----	B95
1. Principes généraux de minoration du rang tensoriel -----	B97
a/ Utilisation des propriétés du rang tensoriel -----	B97
b/ Utilisation des transformations invariantes -----	B101
2. Complexité du produit de deux matrices -----	B105
a/ Borne inférieure dans le cas général -----	B105
b/ Cas particulier du produit de deux matrices 2,2 -----	B107
3. Produit de deux quaternions -----	B113
4. Produit vectoriel de deux vecteurs, produit de LIE de deux matrices 2 x 2 -----	B127
5. Rang tensoriel des matrices anti-symétriques -----	B134
Conclusions -----	B138
REFERENCES SUR LE CHAPITRE BV -----	B143-145
 PARTIE C : ETUDE DES PRINCIPAUX CALCULS MATRICIELS POUR DIFFERENTS CRITERES DE COUTS -----	
INTRODUCTION -----	C1
 <u>CHAPITRE C1 : Rappels sur les calculs matriciels</u> -----	
1. Principales notations -----	C6
2. Matrices de formes particulières -----	C7
3. Quelques propriétés sur les matrices -----	C10
4. Inversion d'une matrice par des méthodes de partitionnement	C13
a/ Partitionnement d'une matrice carrée stable pour la multiplication -----	C15
b/ Construction de tels partitionnements -----	C16
c/ Calcul de l'inverse d'une matrice :	
1) Cas d'un groupe cyclique -----	C18
2) Cas d'un groupe abélien -----	C20
REFERENCES SUR LE CHAPITRE C1 -----	C25

	Pages
<u>CHAPITRE CII : APPLICATIONS DES RESULTATS DE LA PARTIE B ---</u>	C26
1. Inversion d'une matrice régulière -----	C27
2. Calcul du déterminant d'une matrice -----	C31
3. Décomposition LR et OR -----	C32
4. Cas des matrices de formes particulières -----	C36
5. Algorithmes utilisant les matrices de rotations élé- mentaires -----	C41
REFERENCES SUR LE CHAPITRE CII -----	C43
 <u>CHAPITRE CIII : OPTIMALITE DE QUELQUES ALGORITHMES BASES SUR L'EMPLOI DES MATRICES ELEMENTAIRES -----</u>	 C44
1. Résolution d'un système linéaire -----	C46
a/ Algorithmes optimaux -----	C47
b/ Optimalité de Gauss pour l'obtention d'un système triangulaire -----	C48
c/ Algorithmes optimaux de résolution d'un système linéaire -----	C53
2. Transmutation d'une matrice sous forme d'Hessenberg, tridiagonale, et de Frobenius -----	C56
a/ Algorithmes optimaux -----	C57
b/ Transmutations par des matrices élémentaires -----	C57
c/ Méthode optimale d'obtention de la forme d'Hessenberg	C59
d/ Méthode optimale de réduction sous forme tridiagonale	C63
e/ Méthode optimale de réduction sous forme Frobenius	C67
REFERENCES SUR LE CHAPITRE CIII -----	C69
 <u>CHAPITRE C IV : UTILISATION OPTIMALE DU PARALLELISME -----</u>	 C70
1. Produit de deux matrices -----	C72
2. Inversion d'une matrice. Résolution d'un système linéaire	C74
3. Décomposition LR et QR -----	C77
4. Obtention d'une forme canonique semblable à une matrice	C79
5. Matrices de formes particulières -----	C81
REFERENCES SUR LE CHAPITRE C IV -----	C87

<u>CHAPITRE C V : INFLUENCE DE LA PAGINATION SUR LA RAPIDITE</u>	
<u>D'EXECUTION DES ALGORITHMES DE CALCULS</u>	
<u>MATRICIELS -----</u>	C88
1. Principales définitions -----	C90
2. Modèles mathématiques -----	C93
3. Etudes de différents algorithmes -----	C97
1/ Stockage d'une matrice -----	C97
2/ Opérations élémentaires sur les matrices -----	C98
3/ Résolution d'un système linéaire -----	C100
4/ Obtention des formes canoniques -----	C100
5/ Tridiagonalisation d'une matrice symétrique -----	C105
a/ Méthode de Givens -----	C105
b/ Méthode d'Householder -----	C108
6/ Calcul des valeurs propres et des vecteurs propres par la méthode de Jacobi -----	C109
REFERENCES SUR LE CHAPITRE C V -----	C115

INTRODUCTION GENERALE

Ce travail est divisé en trois parties A, B et C relativement indépendantes.

Dans la partie A, on présente brièvement les différents aspects des recherches engagées depuis une dizaine d'années dans le domaine de la complexité des algorithmes. On y discute en particulier des principaux critères de coûts envisageables. Cette partie a été rédigée dans le but d'introduire certains concepts repris par la suite et de permettre de situer dans un contexte plus large l'objet des parties B et C. En la complétant d'une liste de références sur ce sujet, on espère faire oeuvre utile.

Les parties B et C constituent l'essentiel de ce travail.

La partie B est consacrée à l'étude de la complexité du calcul de plusieurs formes bilinéaires, cette complexité étant mesurée par le nombre minimal de multiplications, entre deux opérandes quelconques, que doit nécessairement utiliser tout algorithme de calcul de ces formes bilinéaires.

Cette étude est évidemment motivée par le fait que de nombreux calculs se présentent sous la forme de l'évaluation de plusieurs formes bilinéaires particulières. On peut citer, par exemple, les cas, du produit de deux polynomes, du produit de convolution de deux vecteurs, du produit de deux quaternions, du produit scalaire et du produit vectoriel de deux vecteurs, et évidemment le cas du produit de deux matrices. En particulier, la découverte par STRASSEN, d'une méthode de calcul du produit de deux matrices carrées $n \times n$ en $n^{\log_2 7}$ multiplications a subitement révélé ce nouveau domaine de recherches possibles, concernant pourtant des calculs à priori simples.

Dans la partie C, on se propose d'étudier la complexité des principaux calculs matriciels pour des critères de coûts variés.

Les calculs matriciels ainsi examinés sont les suivants : résolution d'un système linéaire, inversion d'une matrice, calcul d'un déterminant, décomposition LR ou QR d'une matrice, obtention des formes canoniques semblables à une matrice (forme d'Hessenberg, de Frobenius et forme tridiagonale), calcul du polynôme caractéristique, calcul des valeurs propres par les méthodes LR, QR et par la méthode de Jacobi. Les critères de coûts étudiés sont : le nombre de multiplications, le nombre d'étapes de calculs en parallèle, le nombre d'appels de pages (dans un système avec pagination).

On va maintenant présenter les principaux résultats obtenus dans ces deux parties.

Pour un calcul de plusieurs formes bilinéaires particulières il s'agit, d'une part d'obtenir une minoration (la meilleure possible) du nombre minimal de multiplications nécessaires pour évaluer ces formes bilinéaires, et d'autre part de construire un algorithme utilisant le moins de multiplications possibles. L'idéal étant d'arriver au cas où la minoration obtenue est égale au nombre de multiplications utilisées par un algorithme connu, qui sera donc qualifié d'optimal.

Pour développer cette étude, il est tout d'abord nécessaire de définir la classe des algorithmes, que l'on va utiliser ainsi que le critère de coût employé. Ceci sera fait dans le chapitre BI.

Dans le chapitre BII, on donne une caractérisation du coût minimal de calcul de p formes bilinéaires à la fois dans le cas où l'on utilise la commutativité et dans le cas où la commutativité n'est pas utilisée. On relie ce coût minimal à la notion de rang tensoriel de p matrices.

Les chapitres suivants contiennent l'essentiel de nos résultats personnels sur cette question.

Le chapitre BIII, est consacré à l'étude de la notion de rang tensoriel de p matrices. On y donne les principales propriétés du rang tensoriel d'un ensemble de matrices. On montre en particulier que l'on

peut interpréter ce rang tensoriel comme le rang d'un tenseur d'ordre trois associé à ces matrices. On donne aussi la condition pour que p matrices dont une au moins soit régulière ait un rang tensoriel égal à leur dimension. On examine ensuite dans cette optique le problème du produit de deux matrices. Enfin, on termine ce chapitre par l'exposé d'une méthode de calcul approché du rang tensoriel.

Dans le chapitre BIV on étudie les espaces de matrices de rang tensoriel égal à leur dimension. On montre que sur un corps K possédant une racine n ème de l'unité, l'espace des matrices cycliques $n \times n$ possède une seule base tensorielle. On en déduit la seule formule optimale de calcul du produit de convolution de deux vecteurs de K^n . On donne toutes les bases tensorielles de l'espace des matrices de Toeplitz et on en déduit toutes les formules optimales de calcul du produit de deux polynômes. On montre que l'on peut inverser une matrice de Toeplitz triangulaire en $O(n)$ multiplications, et on en déduit que la division de deux polynômes peut se faire en $O(n)$ multiplications. On étudie enfin les espaces de matrices, ayant des symétries particulières, qui possèdent des bases tensorielles.

Le chapitre BV regroupe plusieurs preuves originales de l'optimalités de certains algorithmes. On commence d'abord par donner quelques procédés généraux de minoration du rang tensoriel. On donne ensuite une nouvelle preuve de l'optimalité de la méthode de STRASSEN, une démonstration de l'optimalité du produit de deux quaternions en huit multiplications, de même qu'une démonstration de l'optimalité du calcul du produit vectoriel de deux vecteurs de R^3 en cinq multiplications et une détermination du rang tensoriel de l'espace des matrices anti-symétriques de dimension n .

Dans la partie C, les résultats originaux sont essentiellement contenus dans les chapitres CII, CIII, CIV et CV.

Le chapitre CI en effet regroupe surtout les notations utilisées dans cette partie, ainsi que les principales définitions relatives aux algorithmes étudiés. Le chapitre se termine par l'étude de l'inversion d'une matrice par des méthodes de partitionnement, l'étude du cas d'un partitionnement dû à un groupe abélien étant personnel.

Le chapitre CII fait le lien avec les résultats de la partie B. On montre comment l'utilisation de la méthode de STRASSEN permet d'obtenir des algorithmes nécessitant $O(n^{\log_2 7})$ opérations arithmétiques pour inverser une matrice régulière, pour calculer son déterminant et pour déterminer ses décompositions du type LR ou QR. Ces résultats sont dûs en partie à SCHONHAGE et à STRASSEN. Par contre, les résultats sur les matrices cycliques, et sur le produit par des matrices de rotation sont personnels. On montre en effet que le calcul de l'inverse et le calcul du déterminant d'une matrice cyclique de dimension n peuvent se faire en $O(n \log n)$ opérations arithmétiques si on peut appliquer la FFT, et en $O(n^2)$ sinon. On termine ce chapitre sur une remarque concernant le produit d'une matrice par une matrice de rotation élémentaire, remarque qui permet de diminuer du quart le nombre de multiplications utilisées par tous les algorithmes basés sur l'emploi de telles matrices.

Dans le chapitre CIII, on étudie la classe des algorithmes qui n'utilisent que des produits par des matrices élémentaires et des calculs rationnels. On montre tout d'abord que la méthode de Gauss est, dans cette classe, la méthode optimale pour l'obtention de la forme triangulaire d'un système linéaire. Sous l'hypothèse que les algorithmes considérés conservent les zéros créés on montre également l'optimalité de la méthode de Gauss pour la résolution du système. On décrit tous les algorithmes optimaux de résolution du système, et on montre en particulier qu'il en existe qui ne passe pas par l'obtention de la forme triangulaire. Ce dernier résultat montrant que l'étude de Klyuyev-Kokovkin Scherback sur ce sujet méritait d'être reconsidéré.

On montre ensuite, avec les mêmes hypothèses, l'optimalité de certains algorithmes qui permettent d'obtenir les formes d'Hessenberg, de Frobenius, ou tridiagonale d'une matrice par des transmutations à l'aide des matrices élémentaires. Les algorithmes optimaux dans le cas de l'obtention des formes de Frobenius et tridiagonale diffèrent des algorithmes classiques.

Le chapitre CIV est consacré à l'étude du cas où différentes opérations arithmétiques peuvent être effectuées en même temps (c'est-à-dire "en parallèle").

On étudie les performances des principaux algorithmes de calculs matriciels quand on admet un nombre arbitraire de calculateurs "en parallèle". On montre en particulier que le calcul de la décomposition LR d'une matrice peut se faire en parallèle en $O((\log_2 n)^3)$ unités de temps, la décomposition QR en $O(n)$, et l'obtention d'une forme canonique en $O(n \log n)$. On montre aussi que le calcul du déterminant et de l'inverse d'une matrice bande de largeur de bande $2k+1$ peut se faire en $O(\log_2 \left(\frac{2k+1!}{k! k+1!} \right) \log_2 n)$ unités de temps. Ceci généralise, par une technique différente, un résultat connu sur les matrices tridiagonales.

On termine ce chapitre par le cas des matrices symétriques ou hermitiques, en montrant que la méthode de Givens pour la tridiagonalisation d'une telle matrice se prête mal au calcul en parallèle puisqu'elle nécessite $O(n^2)$ unités de temps.

Le dernier chapitre de cette partie, et de cette thèse, est consacré à l'influence de la pagination sur la conception et sur la performance des principaux algorithmes de calculs matriciels. On suppose dans ce chapitre que les données relatives à un calcul sont groupées en blocs appelés pages dont certaines se trouvent dans la mémoire centrale et d'autres dans une mémoire secondaire. Pour qu'un algorithme puisse effectuer une opération il faut que toutes les données nécessaires soient dans la mémoire principale de l'ordinateur et donc que leurs pages y soient. Si toutes les pages contenant les données ne peuvent être en mémoire centrale simultanément, il faudra, à certains instants, aller en chercher une nouvelle dans la mémoire secondaire. Ceci étant une opération très coûteuse, il s'agit d'organiser les calculs de façon à minimiser ces cas.

Après un bref rappel sur les notions liées à la pagination, on étudie dans cette optique la conception des principaux calculs matriciels. Les résultats originaux concernent l'étude de l'obtention des formes d'Hessenberg, de Frobenius et tridiagonale d'une matrice par l'utilisation des matrices élémentaires, ainsi que l'étude de la tridiagonalisation d'une matrice symétrique par la méthode de Givens et la méthode d'Householder et l'étude de la méthode de Jacobi.

A propos de l'organisation du texte, chaque partie a sa propre pagination. La page B10 se rapporte à la dixième page de la partie B. La partie A est constituée de deux chapitres I et II suivie de la bibliographie.

L'organisation des parties B et C est la même. Le plan détaillé de la partie figure dans les premières pages, ainsi qu'une introduction. Les chapitres sont repérés par la lettre de la partie à laquelle ils appartiennent suivie d'un chiffre romain. Les paragraphes d'un chapitre sont numérotés par des chiffres arabes. Chaque chapitre est suivi des références qui y sont mentionnées. A l'intérieur d'une même partie toutes les références distinctes ont des numéros distincts. La numérotation des théorèmes, remarques, ..., est interne à chaque chapitre.

Un théorème sera référencé par son numéro s'il se trouve dans le même chapitre que la référence, son numéro précédé de la lettre de son chapitre (et de sa partie) s'il se trouve dans un autre chapitre (dans une autre partie).

On emploiera très souvent les notations suivantes :

$\lceil x \rceil$ pour désigner le plus petit entier supérieur ou égal à x .

$\lfloor x \rfloor$ pour désigner le plus grand entier inférieur ou égal à x

et on écrira : $f(x) = O(g(x))$ s'il existe une constante c positive telle que l'on ait (sauf peut-être pour un ensemble fini de valeurs) :

$$f(x) < c g(x).$$

A1

PARTIE A

GENERALITES

PLAN DE LA PARTIE A

- I Fondements théoriques de la complexité.

- II Discussion des principaux critères de coûts
 - 1/ Problèmes de calculs algébriques
 - 2/ Complexité de processus itératifs
 - 3/ Un exemple de fonction coût mémoire :
calculs avec des matrices creuses
 - 4/ Un cas de fonction coût logarithmique :
calculs algébriques exacts
 - 5/ Utilisation du parallélisme dans les calculs.

INTRODUCTION

Cette partie a pour but, comme on l'a indiqué précédemment, de présenter les différentes voies de recherches possibles dans le domaine de la complexité des algorithmes. Elle ne constitue donc pas un exposé de résultats personnels et n'a pas non plus la prétention d'être un "survol" des multiples recherches entreprises ces dix dernières années dans ce domaine.

On va plutôt chercher à illustrer par quelques exemples les différents aspects de ces recherches, et par la même occasion d'introduire les principaux concepts qui serviront par la suite. Ceci devrait permettre également de situer dans un contexte plus large l'objet de ce travail.

Dans le premier chapitre de cette partie, on évoque les recherches sur les fondements de la complexité. Ces recherches se présentent naturellement comme un prolongement des études sur la calculabilité, et utilisent souvent comme modèle de base une machine de Turing. On essaie de montrer rapidement à quel type de résultats elles peuvent conduire et leur importance théorique.

Dans le second chapitre, on présente sur des exemples les principaux critères de coûts envisageables quand on veut étudier la complexité d'une classe précise de problèmes.

On montre dans cet esprit, comment on peut se proposer d'étudier la complexité des processus itératifs, des calculs sur les matrices creuses, ou des calculs algébriques exacts... .

Les références situées à la fin de cette partie concernent les questions évoquées ici mais qui ne sont pas étudiées dans le cours de la thèse

I . FONDEMENTS THEORIQUES DE LA COMPLEXITE DES ALGORITHMES

Pour étudier la complexité des algorithmes il faut tout d'abord définir la classe \mathcal{A} d'algorithmes que l'on utilise, et ensuite se donner une fonction coût qui à tout algorithme A de \mathcal{A} associera son coût $C(A)$ ($C(A) \in \mathbb{R}^+$). La classe \mathcal{A} la plus générale que l'on puisse raisonnablement considérer est celle qui permet de calculer toutes les fonctions partielles récursives (cf. DAVIS (16)) ou qui reconnaît les langages récursivement énumérables. Cette question a été étudiée dans le domaine de la calculabilité, et on peut utiliser plusieurs définitions possibles de \mathcal{A} :

- en utilisant le concept de machine de Turing (TM),
- en utilisant les algorithmes normaux de Markov (24),
- en utilisant des modèles de machines moins primitives que les machines de Turing comme les machines à accès indexé (RAM) qui sont plus proches des ordinateurs habituels :

une machine à accès indexé consiste en un nombre fini d'instructions d'un des types suivantes :

$$a/ \quad X_i \leftarrow X_j \quad T \quad X_k$$

T désignant une des opérations permises par la machine par exemple $+$, $-$, \times , $/$.

Les opérations portent sur le contenu des mémoires X_i , X_j , X_k . Chaque mémoire peut contenir un entier (de taille arbitraire).

b/ Des instructions sur les mémoires :

$$X_i \leftarrow X_{X_j} \quad (\text{chargement indirect})$$

$$X_{X_i} \leftarrow X_j \quad (\text{rangement indirect})$$

c/ Des instructions de contrôle : branchement conditionnel ou non et instructions d'entrées sorties.

Si on définit le coût de chaque instruction d'une RAM, on pourra calculer le coût d'un programme. On peut prendre :

- un coût fixe pour chaque instruction
- un coût variable pour chaque instruction. Par exemple on peut supposer le coût de chaque instruction proportionnel à la taille des opérandes (représentés en binaire) : coût logarithmique.

En ce qui concerne les machines de Turing a une ou plusieurs bandes on peut définir le coût en temps de calcul (nombre d'instructions à faire), ou en mémoire (largeur maximale de la bande).

Les questions que l'on peut étudier dans ce cadre sont les suivantes :

- a/ Comparer les performances des différents modèles pour un même calcul.
- b/ Comparer au sein d'un même modèle, les coûts des différentes fonctions.

1/ Comparaison des différents modèles.

Soit $f : N \rightarrow N$ une fonction récursive, calculée par une machine de Turing TM et par une RAM (T = +, -).

On désigne par $TM(n)$ le coût du calcul de $f(n)$ par la machine de Turing (coût en temps par exemple) et par $RAM(n)$ le coût logarithmique du calcul de $f(n)$ par la machine RAM.

On cherche à obtenir les résultats du type suivant :

"Si on sait calculer f en $TM(n)$ par une machine de Turing, on saura la calculer par une RAM en $TM^2(n)$ " (cf. COOK-RECKHOW (14)).

Il y a donc une bonne correspondance entre les temps d'exécution par une machine de Turing et par une RAM à coût logarithmique. Par contre si le coût de chaque instruction de la RAM est fixe les résultats sont différents.

2/ Comparaison des coûts des différentes fonctions.

On montre par exemple que pour tout critère de coût il existe des problèmes de complexité aussi élevée que l'on veut.

Le résultat suivant illustre ce genre d'études :

"Si $T_1(n)$ et $T_2(n)$ sont deux fonctions coûts telles que l'on ait $\lim_{n \rightarrow \infty} \frac{T_1(n)}{T_2(n)} = \infty$

alors il existe un problème reconnaissable par une machine de Turing avec $T_2(n)$ mémoires, mais non reconnaissable avec $T_1(n)$ mémoires seulement".

On peut aussi comparer les coûts en mémoires et en temps...

Les références (5, 6, 7, 8, 14, 15, 18, 19, 20, 21, 26) concernent ce genre d'études.

3/ Machine non déterministe .

On peut concevoir les machines de Turing non déterministes ou les RAM non déterministes, Elles servent à définir les algorithmes non déterministes, c'est-à-dire ceux qui comportent des instructions de choix. Une machine de Turing non déterministe peut à partir d'un état donné se mettre dans plusieurs états.

On peut définir le coût d'un calcul de la même façon que pour les machines non déterministes : le temps du calcul d'une machine de Turing non déterministe est le plus petit temps possible pour qu'elle effectue le calcul.

On peut essayer de comparer les performances des machines déterministes et des machines non déterministes.

Les résultats suivants illustrent cette optique.

"Si un problème est reconnaissable en temps $T(n)$ par une TM non déterministe (avec deux changements d'états possibles), alors il est reconnaissable en temps $2^{T(n)}$ par une TM déterministe".

A l'heure actuelle, on ne sait pas si on pourrait remplacer dans l'énoncé de ce résultat $2^{T(n)}$ par $P(T(n))$ où $P(x)$ serait un polynôme en x .

Cette question est celle de savoir si la classe P des problèmes résolubles en temps polynomial par machine déterministe est égale à la classe NP des problèmes résolubles en temps polynomial par machine non déterministe. (On sait que parmi les problèmes NP se trouvent les problèmes de coloriage des graphes, de programmation en nombre entiers...réputés difficiles).

Les références suivantes traitent de cette question (4, 13, 17,22).

Ce qui précède donne un aperçu des études théoriques sur la complexité des algorithmes, encore que nous n'ayons pas parlé de la complexité au sens de KOLMOGOROV (cf. (11, 23)), ni des implications possibles sur la construction de certaines théories (cf. (1, 2, 9, 25)). Cette approche est cependant trop générale pour étudier de façon précise la complexité d'un problème particulier (du moins en ce qui concerne les problèmes de la classe P). C'est cette question qui va être abordée dans la suite.

II . DISCUSSION DES PRINCIPAUX CRITERES DE COUT.

Pour étudier la complexité d'un problème P donné, on doit tout d'abord définir une classe d'algorithmes \mathcal{A} dans laquelle on distingue l'ensemble \mathcal{A}_P des algorithmes qui résolvent P. On définit ensuite le coût d'un algorithme de \mathcal{A} (par la donnée d'une fonction coût $C : \mathcal{A} \rightarrow \mathbb{R}^+$). La complexité du problème P (vis-à-vis de la classe \mathcal{A} et de la fonction coût C) est alors caractérisée par la quantité : $C = \min_{A \in \mathcal{A}_P} C(A)$.

Une majoration M de cette quantité est évidemment fournie par l'étude du coût d'un algorithme connu (cette analyse peut être très ardue dans certains cas). L'obtention d'une minoration m de C constitue l'étude de la complexité du problème P. Cette étude peut être considérée comme terminée quand on a effectivement déterminé la valeur de C (on doit avoir $m = M$). Dans ce cas, on connaît au moins un algorithme de coût minimal (le meilleur, pour le critère considéré, dans l'ensemble \mathcal{A}_P). Si par contre $m < M$ il reste soit à trouver une meilleure borne inférieure, soit à chercher un algorithme de coût moindre.

En général, la définition de la classe \mathcal{A} et la définition de la fonction coût dépendront de la nature du (ou des) problème(s) dont on veut étudier la complexité. Pour étudier la complexité des méthodes de tri on ne prendra pas forcément la même classe \mathcal{A} et la même fonction coût que pour étudier la complexité du calcul d'une racine carrée ou du produit de deux polynômes.

De ceci découle évidemment le caractère très divers des recherches de ce type. A part la motivation, et, dans certains cas, le critère de coût choisi, l'unité de ces recherches provient avant tout du domaine mathématique étudié. On va dans la suite présenter les différentes notions de coût possibles.

1/ Problèmes de calculs algébriques

Soit K un anneau, non forcément commutatif, et soient x_1, \dots, x_n , n indéterminées. On considère l'anneau des polynômes à n indéterminées $K[x_1, \dots, x_n]$. (On note par +, -, × les opérations de K et de $K[x_1, \dots, x_n]$. Quand aucune confusion n'est possible, le signe × sera parfois omis).

Deux éléments P_1 et P_2 de $K[x_1, \dots, x_n]$ seront dits équivalents si l'on peut transformer l'un en l'autre par application des propriétés des opérations $+$, $-$ et \times . On écrira $P_1 \equiv P_2$. Si les indéterminées X et Y ne commutent pas pour la multiplication, on a sur $K[x, y]$:

$$(x+y)(x-y) \equiv x^2 - y^2 + yx - xy .$$

Dans le cas commutatif, on a aussi :

$$(x+y)(x-y) \equiv x^2 - y^2 .$$

On peut donc considérer le problème suivant (problème P) :

Etant donné p éléments p_1, \dots, p_p de $K[x_1, \dots, x_n]$, Calculer à partir de $D = \{x_1, \dots, x_n\}$ ($D \subset K$), p éléments p'_1, \dots, p'_p de $K[x_1, \dots, x_n]$ tels que $p'_i \equiv p_i \quad \forall i \in \{1, \dots, p\}$.

a/ Définition de la classe d'algorithmes (cf. Winograd (137))

On définit la classe \mathcal{A} des algorithmes que l'on utilise de la manière suivante :

Un algorithme A de \mathcal{A} est constitué d'une suite finie d'instructions notée I . L'ensemble I doit être totalement ordonné et on numérote les éléments de I avec les entiers de 1 à $\text{card}(I)$. La k ème instruction d'un tel algorithme doit avoir l'un des deux types suivants :

$$\begin{array}{ll} \text{type 1} & \langle k \rangle \leftarrow \langle i \rangle T \langle j \rangle \quad (i < k, j < k) \\ \text{type 2} & \langle k \rangle \leftarrow a \end{array}$$

T peut être l'une des opérations $+$, $-$, \times de $K[x_1, \dots, x_n]$. Dans une instruction du type 2 (instruction d'affectation) le symbole a doit être remplacé par un élément quelconque de $D \cup \{x_1, \dots, x_n\}$. Enfin, $\langle k \rangle$ désigne le résultat de la k ème instruction.

\mathcal{A}_p est l'ensemble des éléments de \mathcal{A} satisfaisant la propriété suivante : $A \in \mathcal{A}$ et il existe p instructions i_1, \dots, i_p telles que $\langle i_k \rangle \equiv p_k$, $k=1, \dots, p$.

b/ Fonctions coûts

Une instruction de type 2 aura toujours un coût nul. Le coût d'un algorithme est la somme des coûts de toutes ses instructions.

Le coût en multiplications actives correspond au cas où le coût d'une instruction de type 1 est égal à un si T est la multiplication et si aucun des deux arguments n'est un élément de D et vaut zéro dans les autres cas.

Le coût en multiplications (C_x) correspond au cas où le coût d'une instruction 1 vaut un si T est une multiplication et zéro autrement. Le coût en addition (C_+), est défini de façon analogue. Enfin, le coût C_{+x} (coût arithmétique) correspond au cas où toute instruction de type 1 a un coût unité.

Si l'analyse de la complexité d'un algorithme de \mathcal{A} ne pose, avec ces définitions, aucune difficulté, il n'en n'ira pas de même pour l'obtention d'une minoration m du coût d'un problème.

c/ Exemple : évaluation d'un polynome.

Le problème P considéré dans ce paragraphe est celui de l'évaluation d'un polynome $p(x)$ de $K[x]$ (K étant ici un corps). Il s'agit donc de calculer $p(x) \equiv a_0 + a_1x + \dots + a_nx^n$ à partir de $D \cup \{x\}$ avec ici $D = \{a_0, \dots, a_n\}$.

Le schéma d'Horner permet le calcul de $p(x)$ en n multiplications et n additions ; Ostrowski, pour les faibles valeurs de n, Pan et Belaga dans le cas général, ont démontré le théorème suivant :

THEOREME 1

/ Tout algorithme de la classe \mathcal{A} utilise au moins n opérations \pm et au moins n opérations x pour évaluer un polynome de degré n. /

Ainsi le schéma d'Horner est optimal à la fois pour les fonctions coûts C_+ , C_x et C_{+x} définies précédemment. De plus, Borodin (112) a montré qu'il n'existait pas d'autre algorithme optimal.

Dans le cas où un même polynôme doit être évalué en un grand nombre de points, il devient avantageux d'effectuer au préalable des calculs sur les coefficients de façon à diminuer le nombre d'opérations à effectuer ensuite (préconditionnement).

Pour les schémas avec preconditionnement on ne comptera que les multiplications dont un des arguments utilise la variable x .

Dans ce cas, on a le théorème suivant (Bélagá (110)).

THEOREME 2

/ Tout schéma de calcul avec preconditionnement nécessite au moins $\lceil \frac{n+1}{2} \rceil$ multiplications. /

Désignons par q_1, \dots, q_m les résultats des m multiplications que l'on compte. On peut écrire :

$$\begin{aligned}
 q_1 &= x \times (\beta_1' x + \gamma_1') \quad , \\
 q_j &= \left(\sum_{i=1}^{j-1} \alpha_{ji} q_i + \beta_j x + \gamma_j \right) \times \left(\sum_{i=1}^{j-1} \alpha_{ji}' q_i + \beta_j' x + \gamma_j' \right) \quad , \\
 &\quad (j=2, \dots, m) \quad (\alpha_{ji}, \beta_j, \alpha_{ji}', \beta_j' \in K) \quad . \\
 p(x) &= \sum_{i=1}^m \alpha_i q_i + \gamma_{m+1} \quad (1) \quad .
 \end{aligned}$$

Les γ_i ($i=2, \dots, m+1$) et les γ_i' ($i=1, \dots, m$) désignent des polynômes en les coefficients a_0, \dots, a_n .

On voit d'après (1), que l'on peut écrire :

$$a_i = g_i(\gamma_2, \dots, \gamma_{m+1}, \gamma_1', \dots, \gamma_m') \quad i=0, \dots, n \quad .$$

Il est aisé de montrer que l'on doit avoir $2m \geq n+1$ car sinon, les a_i seraient liés par une relation algébrique.

Par conséquent, on a bien $m \geq \lceil \frac{n+1}{2} \rceil$.

Dans Knuth (75), il est montré comment construire de tels schémas avec preconditionnement utilisant $\lceil \frac{n+1}{2} \rceil + 1$ multiplications.

Paterson et Stockmeyer (129) ont de plus construit un préconditionnement n'utilisant que des rationnels qui nécessite $\frac{n}{2} + O(\log n)$ multiplications et $\frac{5n}{4}$ additions (cf. aussi Rabin et Winograd (131)).

Remarque

Quand le polynôme $p(x)$ doit être évalué en k points, connus à l'avance, Borodin et Munro (113) ont donné un algorithme qui n'utilise qu'environ $k.n^{0.9}$ multiplications dès que $k > \sqrt{n+1}$.

2. Complexité de processus itératifs

On considère le problème suivant : trouver une solution x^* de l'équation $p(x) = 0$ où $p(x)$ est un polynôme de degré n à coefficients dans \mathbb{Q} .

Il s'agit en fait de construire une suite x_0, x_1, \dots , telle que $\lim_{p \rightarrow \infty} x_p = x^*$. Comment peut-on étudier la complexité d'un tel problème ?

On considère pour simplifier (mais la démarche reste la même dans les cas plus généraux), uniquement les suites x_i générées de la façon suivante : x_0 étant donné, on calcule par itération :

$$x_{i+1} = \frac{p_1(x_i)}{q_1(x_i)} \quad (p_1(x) \text{ et } q_1(x) \text{ sont deux polynômes à coefficients rationnels}).$$

On désigne par \bar{M} (Resp. M) le nombre minimal de multiplications (d'opérations mathématiques) nécessaires pour évaluer la fraction rationnelle $\frac{p_1(x)}{q_1(x)}$.

La rapidité de convergence de la suite x_i est caractérisée par son ordre de convergence p : c'est le nombre p tel que :

$$\lim_{i \rightarrow \infty} \frac{|x_{i+1} - x^*|}{|x_i - x^*|^{p-\epsilon}} = 0 \quad \text{et} \quad \lim_{i \rightarrow \infty} \frac{|x_{i+1} - x^*|}{|x_i - x^*|^{p+\epsilon}} \neq 0 \quad \forall \epsilon > 0.$$

La complexité C de la génération de la suite x_i doit être une fonction croissante de \bar{M} (ou de M) et décroissante avec p et telle que si l'on compose le processus deux fois elle reste inchangée :

$$x_{i+1} = \frac{P_1}{Q_1} \left(\frac{p_1(x_i)}{q_1(x_i)} \right) \text{ a une complexité } 2M \text{ et un ordre de convergence } p^2.$$

On peut donc prendre :

$$C = \frac{M}{\log_2(P)} \quad \left(\bar{C} = \frac{\bar{M}}{\log_2(P)} \right) .$$

Pour ce problème on a les résultats suivants :

THEOREME 3 (Paterson) $1 \leq \bar{C} \leq C$.

/ La méthode de Newton est optimale pour le calcul de la racine d'un polynôme de degré deux. /

THEOREME 4 (Kung).

/ Si x^* est un nombre algébrique de degré k (son polynôme minimal est de degré k), alors on a :

$$\bar{C} \geq \frac{\lceil \log_2(k(\lceil p \rceil - 1) + 1) \rceil - 1}{\log_2(p)} . /$$

Il est clair que l'on peut étudier ainsi la complexité d'un grand nombre de problèmes donnant lieu à des processus de résolution itératifs en restreignant ou en généralisant le type de processus. La bibliographie jointe en annexe donne une idée des recherches diverses engagées dans cette direction .

3. Un exemple de fonction coût mémoire :

Calculs avec des matrices creuses.

Soit A une matrice de $M_{n,n}(R)$. On dira que A est une matrice creuse si le nombre de ses éléments non nuls est de l'ordre de kn avec $k \ll n$. Dans ces conditions, pour n grand, il devient intéressant de ne placer en mémoire que les éléments non nuls de la matrice et de concevoir les algorithmes de calcul matriciel en fonction de cette propriété. En ce qui concerne le problème du calcul des valeurs propres d'une matrice disons tout de suite qu'il s'agit pratiquement d'un problème ouvert. Par contre, en ce qui concerne le problème de la résolution du système linéaire $AX = B$ et du calcul de l'inverse plusieurs résultats intéressants ont été obtenus.

a/ Recherche des permutations optimales

Soit à résoudre le système $AX = B$ par une méthode directe. Si P et Q sont deux matrices de permutations, il revient au même de résoudre le système $(PAQ)X' = B'$ avec $X' = Q^t X$, $B' = PB$. Si $Q = P^t$ le conditionnement de la matrice A et celui de la matrice PAP^t sont égaux. On va se limiter à ce cas dans la suite.

PROBLEME

Déterminer la matrice de permutation P telle que si on applique la méthode de Gauss sur PAP^t on minimise le nombre de mémoires à utiliser.

$$A = \begin{pmatrix} x & x & x & \dots & x \\ x & x & & & \\ x & & x & & 0 \\ \vdots & & & \ddots & \\ x & & 0 & & x \end{pmatrix}$$

Considérons l'exemple de la matrice $n \times n$ symétrique définie positive A .

A a des éléments non nuls uniquement sur la diagonale, sur la première ligne et sur la première colonne.

Si on applique sur cette matrice la méthode de Gauss, il va falloir en fait $n(n+1)/2$ mémoires pour stocker la décomposition LDL^t et effectuer pour l'obtenir $\frac{n^3}{6} + \frac{n^2}{2} - \frac{2n}{3}$ multiplications.

$$\text{Si par contre on considère } A' = PAP^t = \begin{pmatrix} x & & & & x \\ & x & & & \\ & & \ddots & & \\ & & & x & \\ x & x & & & x \end{pmatrix}$$

alors on aura besoin que de $2n-1$ mémoires et $2(n-1)$ multiplications (et $n-1$ additions).

L'hypothèse que la matrice A soit symétrique et définie positive n'est pas fortuite : elle permet d'affirmer que l'on peut effectuer l'élimination de Gauss sur la matrice PAP^t ($\forall P$) sans rencontrer de pivot nul. On aura la même assurance si A est à diagonale dominante. Dans ces conditions la structure creuse de la matrice sera représentée par le graphe de n noeuds N_1, N_2, \dots, N_n ayant un axe de N_i à N_j si $A(i,j) \neq 0$. (Si la matrice est symétrique, le graphe sera non orienté). On peut toujours se ramener au cas d'un graphe fortement connexe (A irréductible) car sinon il existe P telle

que PAP^t soit une matrice bloc triangulaire supérieure avec les matrices blocs diagonales irréductibles. On peut facilement interpréter l'élimination de Gauss sur ce graphe : Si $A(1,1)$ est le pivot, l'on veut annuler tous les éléments non nuls se trouvant en-dessous de lui. La matrice des $n-1$ dernières lignes et colonnes obtenues correspond à un nouveau graphe de noeuds N_2, \dots, N_n déduit du graphe G associé à A de la manière suivante :

On enlève N_1 ainsi que les arcs incidents à N_1 et on rajoute un arc de N_i à N_j si $(N_i, N_j) \notin G$ et si il existe un chemin de longueur deux de N_i à N_j passant par N_1 .

Le problème sur le graphe G est de trouver une permutation de ses noeuds telle que l'application du processus précédent crée le moins possible d'arcs.

Conjecture : le problème précédent est N_p complet au sens de Karp.

Un certain nombre de méthodes pour obtenir une solution approchée à ce problème on cependant été trouvées (cf. Tewarson-Tinney-Walker). On peut essayer de trouver P telle que la matrice PAP^t soit une matrice bande de largeur minimum (Cuthill et Mckee). Dans le cas de matrices symétriques, on peut rechercher une triangulation optimale du graphe (Bunch).

b/ Bornes inférieures

Le problème est de minorer le nombre de mémoires nécessaires pour effectuer l'élimination de Gauss sur les matrices PAP^t . Dans le cas de la discrétisation de l'équation de Laplace (en n^2 points) avec la formule à cinq points on obtient un graphe particulier G . Hoffman, Martin et Rose ont montré, par ce graphe qu'il serait toujours nécessaire d'avoir $n^2 \log(n)$ mémoires (Georges a trouvé une permutation conduisant à $8 n^2 \text{Log}(n)$ mémoires nécessaires).

Remarque 1

Si la matrice A ne vérifie pas une condition assurant la non nullité de tous les pivots possibles des matrices PAP^t on peut alors chercher à modifier, à chaque étape de la méthode de Gauss, le système à résoudre, de façon à minimiser le nombre de zéros créés ensuite (cf. Tewarson-Markowitz).

Remarque 2

Il ne faut évidemment pas oublier cependant que les méthodes itératives sont particulièrement bien adaptées à la résolution de très gros systèmes linéaires. Au point de vue complexité des calculs et encombrement mémoire, la méthode de sur-relaxation est aussi bonne que les meilleures méthodes directes vues précédemment.

4/ Un cas de fonction coût logarithmique :
Calculs algébriques exacts.

Mesurer la complexité d'un algorithme par le nombre de multiplications (ou d'additions) qu'il utilise est justifié si le coût d'une multiplication (ou d'une addition) est indépendant des opérands.

Ceci est bien le cas pour les calculs usuels portant sur des nombres flottants en simple précision. Cette hypothèse ne serait plus justifiée pour des calculs en précisions multiples, car le coût du produit de deux nombres en double ou triple précision est supérieur à celui du produit de deux nombres en simple précision.

Depuis une dizaine d'années, on a assisté à un développement important de systèmes de calculs formels dont la caractéristique essentielle est de permettre des calculs sur des entiers de tailles arbitraires (avec une précision illimitée). La remarque précédente s'applique donc tout particulièrement dans le cas des algorithmes qui ont été développés pour ces systèmes.

a/ Coût des opérations élémentaires

Soit a et b deux entiers. On désigne par $\ell(a)$ et $\ell(b)$ le nombre de bits nécessaires pour les représenter: $\ell(a) = \log_2(a)+1$, $\ell(b) = \log_2(b)+1$

On peut définir le coût d'une addition et d'une multiplication de deux entiers par :

$$C(a+b) = \max (\ell(a), \ell(b)) ,$$

$$C(a \times b) = \ell(a) \times \ell(b) .$$

(On suppose que l'on utilise la méthode classique pour faire le produit de deux entiers et non pas par exemple la méthode de SCHONHAGE-STRASSEN qui conduirait à un coût proportionnel à $n \log n \log \log n$ pour le produit de deux entiers de n bits).

Le coût de la division de deux entiers (obtention du quotient et du reste) peut être pris égal à :

$$C(a/b) = \ell(b) \cdot (\ell(a) - \ell(b) + 1) .$$

Soit $A(x_1, \dots, x_r)$ un polynôme de $I[x_1, \dots, x_r]$. On peut l'écrire comme un polynôme en la variable x_r :

$$A(x_1, \dots, x_r) = \sum_{i=0}^{m_r} A_i(x_1, \dots, x_{r-1}) x_r^i .$$

(m_i désigne le degré de $A(x_1, \dots, x_r)$ en x_i).

On peut définir par induction la norme $|A|$ du polynôme $A(x_1, \dots, x_r)$ par :

$$|A| = \max_{0 \leq i \leq m_r} |A_i|$$

et $|a|$ égale à la valeur absolue de a si $a \in I$.

Le coût du produit de deux polynômes $A(x_1, \dots, x_r)$ et $B(x_1, \dots, x_r)$ sera majoré par :

$$(\ell(|A|) + \ell(|B|)) \times \prod_{i=1}^r (m_i + n_i + 1) .$$

(n_i désigne le degré de $B(x_1, \dots, x_r)$ en x_i).

b/ Analyse du coût d'un algorithme

Frenons le cas d'un algorithme qui doit calculer un certain polynôme $C(x)$ à partir des deux polynômes $A(x)$ et $B(x)$. Le coût de cet algorithme dépendra à la fois des degrés m_1 et n_1 des polynômes $A(x)$ et $B(x)$, mais aussi de la taille de leurs coefficients : $\ell(a_0), \dots, \ell(a_{m_1})$, et $\ell(b_0), \dots, \ell(b_{n_1})$.

Il est évidemment en général hors de question d'obtenir l'expression de ce coût en fonction de tous ces paramètres.

Pour avoir une bonne idée de la performance de l'algorithme, il suffit d'estimer son coût maximal, son coût minimal, ainsi que son coût moyen.

Ces trois notions sont définies de la manière suivante :

On désigne par $C(A,B)$ le coût de l'algorithme pour les deux polynomes $A(x)$ et $B(x)$.

$D_{a,b,m,n}$ est l'ensemble des couples de polynomes $A(X)$, $B(X)$ tels que :
 $|A| = a$, $|B| = b$, $\deg A \leq m$, $\deg B \leq n$.

- Le coût maximal de l'algorithme est alors égal à :

$$\max_{A,B \in D_{a,b,m,n}} C(A,B)$$

On peut poser :

$$C^M(a,b,m,n) = \max_{A,B \in D_{a,b,m,n}} C(A,B) .$$

Les polynomes $A(x)$, $B(x)$ qui donnent ce coût maximal constituent les cas les plus défavorables pour l'algorithme.

-Le coût minimal de l'algorithme est égal à :

$$\min_{A,B \in D_{a,b,m,n}} C(A,B) = C^m(a,b,m,n) .$$

Les polynomes $A(x)$, $B(x)$ qui donnent ce coût minimal constituent les cas les plus favorables pour l'algorithme.

- Le coût moyen de l'algorithme est égal à :

$$\left[\sum_{A,B \in D_{a,b,m,n}} C(A,B) \right] \times \frac{1}{\text{card}(D_{a,b,m,n})}$$

Les problèmes posés par l'analyse du coût d'un algorithme sont dans ces conditions très considérables (cf. le cas de la division de deux polynomes ou du calcul du PGCD). A notre connaissance, il n'existe aucun résultat de complexité proprement dit dans ce domaine (résultat de minoration du coût moyen pour un problème donné). L'emploi des techniques modulaires pour l'obtention d'algorithmes performants illustre d'ailleurs la

difficulté de ce problème, mais aussi l'intérêt mathématique à la fois des méthodes de construction des algorithmes et des problèmes posés par l'analyse de leurs performances.

La bibliographie jointe en annexe donne une idée des principaux travaux engagés ces dernières années dans ce domaine.

5/ Utilisation du parallélisme dans les calculs.

On termine cette étude des principaux critères de coûts envisageables par le cas où l'on suppose que plusieurs opérations peuvent être effectuées en même temps, c'est-à-dire en parallèle.

Plusieurs types d'ordinateurs permettent en effet d'effectuer en parallèle plusieurs opérations ("Array processors", ordinateurs pipeline,...).

Pour un calcul donné, il s'agit alors, non plus de minimiser le nombre total d'opérations effectuées, mais de minimiser le nombre d'étapes de calculs en parallèle nécessaire. Pour un algorithme donné, il est en général indispensable de réorganiser ces calculs de façon à profiter au maximum des possibilités de parallélisme.

On peut d'autre part se proposer de chercher de nouveaux algorithmes conçus pour ce type d'exploitation. Enfin on peut essayer d'estimer le gain maximal que peut apporter l'introduction du parallélisme pour un problème spécifique, (en admettant par exemple un nombre arbitraire d'unités de calculs en parallèle). Pour cela, on admet en général que toutes les opérations arithmétiques peuvent se faire en parallèle en une unité de temps. On néglige donc le coût d'accès aux données et de transmissions des résultats. Les très nombreuses études entreprises ces dernières années dans ce domaine ont montré que certains algorithmes peu performant dans le cas des calculs séquentiels pouvaient être les meilleures dans le cas des calculs en "parallèle" et que l'on pouvait être amené à concevoir de nouveaux algorithmes. Pour illustrer ceci, on va étudier le cas de l'évaluation d'un polynôme.

Exemple

Evaluation d'un polynôme.

Soit $p(x) = a_0 + \dots + a_n x^n$ un polynôme de degré n .

On peut évaluer $p(x)$ en x_0 en utilisant le schéma d'Horner qui est optimal

au point de vue du nombre d'additions et de multiplications utilisées :

$$p(x_0) = (a_n x_0 + a_{n-1})x_0 + \dots + a_0$$

Si on dispose de k unités de calculs fonctionnant en parallèle, le temps de l'exécution de ce schéma est toujours de $2n+1$ (On ne peut pas profiter de cette possibilité).

Estrin (90) et Dorn (89) ont été les premiers à montrer que l'on pouvait faire mieux que le schéma d'Horner dans le cas du calcul en parallèle.

Dorn donne une généralisation de la règle d'Horner.

On obtient la règle d'Horner d'ordre k de la façon suivante :

$$p(x) = (x^k - x_0^k)(b_k + b_{k+1}x + \dots + b_n x^{n-k}) \\ + b_{k-1}x^{k-1} + \dots + b_1x + b_0 .$$

Donc :

$$p(x_0) = b_{k-1}x_0^{k-1} + b_{k-2}x_0^{k-2} + \dots + b_1x_0 + b_0$$

les b_j étant calculés de la manière suivante :

$$b_j = a_j \quad j=n, \dots, n-k+1$$

$$b_j = a_j + x_0^k b_{j+k} \quad j=n-k, \dots, 0 .$$

Ce schéma correspond à un calcul en parallèle d'ordre k .

Pour n grand, le calcul de $p(x_0)$ avec k unités arithmétiques opérant en parallèle, prend de l'ordre de $\frac{n}{k}$ fois le temps d'une multiplications (t_m), et $\frac{n}{k}$ fois le temps d'une addition (t_a).

$$\text{Si} \quad t_a = t_m = 1$$

on a :

$$\text{pour} \quad K = 2 \quad T_2 = n+2 .$$

Dorn analyse en détail, les règles d'Horner d'ordre 2, 3 et 4.

Estrin (90), expose une autre règle de calcul utilisant $(n+1)/2$ unités arithmétiques et effectuant le calcul en $T = 2(\log_2 n + 1)$.

Dans (100), on donnait un nouvel algorithme permettant d'évaluer le polynôme $p(x)$ de degré n , en $\frac{3}{2} \log_2 n$ unités de temps avec $\lceil \frac{n}{2} \rceil + 1$ unités de calculs.

Les meilleurs résultats connus sur ce problème sont ceux de Munro-Paterson (105) et de Maruyama (101) qui ont donné une méthode permettant d'évaluer un polynôme de degré n en $\log_2(n) + O((\log_2 n)^{1/2})$ unités de temps avec n unités de calculs ; ce qui est presque optimal puisqu'il faut au moins $\lceil \log_2 n \rceil$ unités de temps pour évaluer un polynôme en parallèle.

L'analyse du coût d'un algorithme pour un nombre limite k d'unités de calculs est en général assez difficile.

L'obtention d'une minoration pour le temps d'exécution en parallèle d'un calcul passe en général par l'obtention d'une minoration du nombre total d'opérations arithmétiques nécessaires ; ce qui peut donc conduire à des minoration assez faibles.

Dans le chapitre CIV on donne quelques résultats sur les calculs matriciels effectués en parallèle.

REFERENCES SUR LA PARTIE AI . Aspects théoriques de la complexité.

- (1) ABERTH, O., "Analysis in the computable number field".
J. ACM 15 (1968), 275-299.
- (2) ABERTH, O., "The concept of effective method applied to computational problems of linear algebra".
J. Comput. Syst. Sci. 5(1971), 17-25.
- (3) AHO, AV., HOPCROFT, JE, ULLMANN, JD., "Time and tape complexity of pushdown automaton languages".
Information and control, 13:3, (1968), 186-206.
- (4) AHO, AV., HOPCROFT, JE., ULLMANN, JD. "The design and analysis of computer algorithm".
Addison Wesley publishing Company (1974).
- (5) ARBIB, MA., BLUM, M., "Machine dependence of degrees of difficulty".
Proceedings Amer. Math. Soc. XVI, n° 3, (June 1967), 442-447.
- (6) BLUM, M., "Recursive function theory and speed of computation.
Canadian Mathematical bulletin, Vol.9 n° 6, (1966), 745-750.
- (7) BLUM, M. "A machine independent theory of the complexity of recursive functions".
J. ACM, vol.14, n°2, (april 1967), 332-336.
- (8) BORODIN, A., "Computational complexity and the existence of complexity gaps".
J. ACM vol.19, n° 1, (January 1972), 158-174.
- (9) BOUHIER, M., "Analyse calculable".
Thèse, Grenoble (1973).

- (10) CHAITIN, GJ., "On the difficulty of computation".
IEEE trans on Information Theory.
Vol. IT 16, n° 1, (January 1970), 5-9.
- (11) CHAITIN, GJ., "On the simplicity and speed for computing infinite sets of natural numbers".
J. ACM vol.16, (July 1969), 407-422.
- (12) COOK, SA. and AANDERAA, SO., "On the minimum complexity of functions".
Trans. Amer. Math. Soc. 142, (1969), 291-314).
- (13) COOK, SA., "A hierarchy for non deterministic time complexity".
J. Computer and System Sciences 7:4, (1973), 343-353.
- (14) COOK, SA. and RECKHOW, RA. "Time bounded random access machines".
J. Computer and System Sciences, 7:4, (1973), 354-375.
- (15) CONSTABLE, "The operator gap".
J. ACM vol.19, n°1 (January 1972), 175-183.
- (16) DAVIS, "Computability and Unsolvability".
McGrawHill book co, Jnc, New-York (1958).
- (17) FLAJOLET, P., STEYAERT JM. "Hierarchies de complexité et réductions entre problèmes".
Journées sur la Conception et l'Analyse des Algorithmes.
IRIA, (22 octobre 1975).
- (18) HARTMANIS, J. and STEARNS, RE., "On the computational complexity of algorithms".
Trans. American, Math. Society, vol.117, Issue 5. (May 1965),
285-306.
- (19) HARTMANIS, J., STEARNS, RE., "Automata based computational complexity".
Information Sciences, 1 (1969), 173-184.

- (20) HARTMANIS, J., LEWIS, PM., STEARNS, RE.,
"Classification of computation by time and memory requirements".
Proc. IFIP, Congress (1965), Spartan, N.Y., 31-35.
- (21) HELLERMAN, L., "A measure of computational work".
IFIP, Trans. on Computers, vol. c21, n° 5, (May 1972)..
- (22) KARP, RM. "Reducibility among combinatorial problems". in Miller
and Thatcher (1972), 85-104.
- (23) LOVELAND, AW., "A variant of the kolmogorov concept of complexity."
Information and control 15, (1969), 510-526.
- (24) MARKOV, "Théorie des algorithmes".
- (25) MILLER, W., "Toward abstract numerical analysis".
J. ACM vol.20, n°3, (July 1973) 395-408.
- (26) PAGER, D. "On the efficiency of algoritms".
J. ACM vol.17, n°4, (octobre 1970), 708-714.
- (27) STRONG, H.R., "Depth Bounded Computation".
J. Computer and System Sciences, 4, (1970), 1-14.
- (28) VALIANT, LG. "Regularity and related problems for deterministic
pushdown automata".
J. ACM , vol.22, n°1 (January 1975), 1-10 .
- (29) YONG, P., "Speed up by changing the order on which sets are
enumerated".
First ACM Symposium on theory of Computing, (May 1969).

II. Complexité des processus itératifs.

- (30) BRENT, RP., "The computational complexity of iterative methods for systems of non linear equations". In Complexity of Computer Computation .
Miller and Thatcher ed. Plenum Press, New-York (1972), 61-71.
- (31) BRENT, RP., "A class of optimal order zero finding methods using derivative evaluations".
Carnegie Mellon University T.R. (June 1975).
- (32) BRENT, RP., " Multiple precision zero finding methods and the complexity of elementary function evaluation".
Carnegie Mellon University T.R. (July 1975).
- (33) BRENT, RP., WINOGRAD, S., WOLFE, P., " Optimal iterative processes for root finding".
Numer. Math. 20, (1973), 317-341.
- (34) HINDMARSH, "Optimality in a class of rootfinding algorithms".
SIAM, J. Numer. Analysis, vol.9, n°2, (June 1972).
- (35) KACEWICZ, B., "An integral-interpolatory iterative method for the solution of nonlinear scalar equations".
Carnegie Mellon University T.R. (January 1975).
- (36) KUNG, HT., "The computational complexity of algebraic numbers".
SIAM J. Numerical Analysis, vol.12, n°1 (1975), 89-96.
- (37) KUNG, HT., TRAUB, JF., "Optimal order of one point and multipoint iteration".
J. ACM , vol.21, n°4, (octobre 1974), 643-651.
- (38) PATERSON, MS., "Efficient iterations for algebraic numbers".
in Complexity of Computer Computations.
R. Miller and J.W. Thatcher ed., plenum press. N.Y., (1972), 44-52.

- (39) RISSANEN, J., "On optimum root-finding algorithms.
J. of Mathematical Analysis and Application 36 (1971), 220-225.
- (40) SCHULTZ, "The computational complexity of elliptic partial differential equations.
Complexity of Computer Computations, New-York (1972).
- (41) WOZNIAKOWSKI, "Maximal stationary iterative methods for the solution of operator equations".
Carnegie Mellon University TR. (décembre 1973).
- (42) WOZNIAKOWSKI, "Generalized information and maximal order of iteration for operator equations".
Carnegie Mellon University TR. (April 1974).
- (43) WOZNIAKOWSKI, "Maximal order of multipoint iterations using n evaluations".
Carnegie Mellon University TR. (July 1975).

III. Matrices creuses

- (44) BIRKHOFF, G. and GEORGE, A., "Elimination by nested dissection".
in Complexity of sequential and parallel algorithm, Traub ed., Acad. Press, (1973).
- (45) BUNCH, JR., "Complexity of sparse elimination".
in Complexity of sequential and parallel algorithms,
Traub ed., Acad. Press (1973).
- (46) BUNCH, JR., "Analysis of sparse elimination".
Cornell Techn. Report 158, (January 1973).
- (47) BUNCH, JR., "Partial pivoting strategies for symmetric matrices".
SIAM J. Num. Analysis vol.11, n°3 (june 1973) 521-525.

- (48) BUNCH, JR., "On block elimination for sparse linear system".
SIAM J. Num. Analysis, vol.11, n°3 (june 1975), 585-603.
- (49) CUTHILL, E., "Several strategies for reducing the band width of matrices". Sparse matrices and their applications. ed. by Rose and Willoughby. Plenum Press, New-York, (1972).
- (50) DUFF, IS., "On the number of zeros added when gaussian elimination is performed on sparse random matrices".
Mathematics of Computation, vol.28, n°125, (january 1974).
- (51) GUSTAVSON, "Some basic techniques for solving sparse systems of equations". in Sparse matrices and their applications.
Plenum Press, New-York, (1972).
- (52) HOFFMAN, MARTIN, ROSE, "Complexity bounds for regular finite difference and finite elements grids".
SIAM J. Num. Analysis, vol.10, n°2, (april 1973).
- (53) ROSE, D., "Triangulated graph and the elimination process".
Journal of Math. Analysis and Applications, vol.32, (1970),
597-609.
- (54) ROSE, D. and TARJAN, E., "Algorithmic Aspects of vertex elimination".
Preprint, (june 1975).
- (55) TARJAN, RE., "Depth first search and linear graphs algorithms".
SIAM J. Computing 1 (1972), 146-160.
- (56) TEWARSON, R.P., "Computations with sparse matrices".
SIAM Review, vol.12, n°5, (oct. 1970) 527-543.
- (57) TINNEY-WALKER, "Direct solutions of sparse network equations by optimally ordered triangular factorizations".
Proc. IEEE 55 (1967) 1801-1809.
- (58) WILLOUGHBY, RA., "A characterization of matrix irreducibility via triangular factorization". SIAM J. Num. Analysis (1975).

IV. Calculs algebriques exacts.

- (59) BERLEKAMP, ER., "Factoring polynomials over finite fields.
Bell system tech Journal vol.46 (1967) 1853-1859.
- (60) BERLEKANP, ER., "Algebraic coding theory".
McGraw Hill New-York (1968).
- (61) BERLEKAMP, ER., "Factoring polynomials over large finite fields
Math. Comp. vol.24, n° 111 (july 1970).
- (62) BROWN, WS., "On euclid's algorithm and the computation of polynomial
greatest common divisors".
J. ACM vol.18, n° 4 (octobre 1971) 478-504.
- (63) BROWN, WS. and TRAUB, JF., "On euclid's algorithm and the theory
of subresultants".
J. ACM vol.18, n°4 (oct. 1971), 515-532.
- (64) COLLINS, GE., "Polynomials remainder sequences and determinants".
AM Math. Monthly, vol.73, n° 7 (august-september 1966) 708-712.
- (65) COLLINS, GE., "Subresultants and reduced polynomial remainder
sequences". J.ACM vol.14, n°1 (jan. 1976) 128-142.
- (66) COLLINS, GE., "Computing time analysés for some arithmetic and
algebraic algorithms".
Proc. (1968) summer Institute on Symbolic Mathematical computations,
(195-231).
- (67) COLLINS, GE., "Computer algebra of polynomials and rational functions".
Am Math. Monthly, vol.80, n°7 (aug.sept. 1973), 725-7254.
- (68) CABAY, S., "Exact solution of linear equations".
Proc. 2th Symposium on symbolic and algebraic manipulation.
(March 23-25, 1971), Los Angelès.

- (69) GENTLEMAN, WM., JOHNSON, SC., "The evaluation of determinants by expansion by minors and the general problem of substitution".
Mathematics of computation, vol.28, n° 126, (april 1974), 543-548.
- (70) HEINDEL, LE., "Integer arithmetic algorithms for polynomial real zero determination".
J.ACM vol.18, n° 4, (oct.1971) 533-548.
- (71) HENRICI, P., "A subroutine for computations with rational numbers".
J. ACM vol.3, n° 1 (jan.1956), 6-9.
- (72) HOROWITZ, E., and SAHNI, S. "On computing the exact determinant of matrices with polynomial entries".
J. ACM, vol.22, n°1 (january 1975) 38-50.
- (73) HOROWITZ, E., "Modular arithmetic and finite field theory".
Proc. 2th symposium on Symbolic and Algebraic Manipulation.
188-194, Los Angelès, (1971).
- (74) HOWELL, JA., "An algorithm for the exact reduction of a matrix to Frobenius form using modular arithmetic I-II.
Mathematics of Computations, vol.27, n° 124, (oct. 1973), 887-920.
- (75) KNUTH, DE., "The art of Computer programming". vol.II.
Seminumerical algorithms. Addison-Wesley Reading, Mass. (1969).
- (76) LIPSON, JD., "Chineser Remainder and interpolation algorithms".
Proc. 2th symposium on Symbolic and Algebraic Manipulation
(1971), 372-391, ACM, New-York.
- (77) LOOS, R., "New exact algebraic algorithms".
Third International colloquium on advanced computing methods in theoretical physics. Marseille, (june 1973) ed. by Visconti .
- (78) McClellan, MT., "The exact solution of systems of linear equations with polynomial coefficients.
Proc. second symposium on symbolic and algebraic manipulation,
ACM New-York, (1971), 399-414.

- (79) MIGNOITE, M., "An inequality about factors of polynomial"
Math. Computations, to appear.
- (80) MOSES, J., YUN, D.Y.Y., "The EZ G CD Algorithms "
Proc. ACM 73, Assoc. Comp. Mech. New-York (1973), 159-166.
- (81) MUSSER, DR., "Multivariate polynomial factorization".
J. ACM vol.22, n° 2 (april 1975), 291-308.
- (82) STALLINGS, WT., and BOULLION, TL., "Computation of pseudoinverse
matrices using residue arithmetic".
SIAM Review vol.14, n°1, (jan. 1972).
- (83) YUN, "Factorization of multivariate polynomials".
Ph.D. Thèse 1973.

V. Calculs numériques en "parallèle"

Les références que l'on donne ici complète celle qui se trouvent à la fin du chapitre CIV.

- (84) BRENT, RP., "The parallel evaluation of arithmetic expressions in
logarithmic time".
in Complexity of Sequential and parallel numerical algorithms,
ed. by Traub, Academic Press, (1973), 83-102.
- (85) BRENT, R.P., KUCK, DJ., and MARUYAMA, KM.,
"The parallel evaluation of arithmetic expressions without divisions".
IEEE trans. comp. c-22, (may 1973).
- (86) CARROLL, AB., WETHERALD, RT., "Application of parallel processing to
numerical weather prediction".
J. ACM vol.14, n° 3, (july 1967) 591-614.

- (87) CHAZAN, D., MIRANKER, WL., "Chaotic relaxation".
Linear algebra and its applications, 2 (1969) 199-222.
- (88) CHAZAN, D., MIRANKER, WL., "A non gradient and parallel algorithm
for unconstrained minimization".
SIAM J. Control, vol.8, n°2 (may 1970) 207-217.
- (89) DORN, W.S., "Generalization of Horner's rule for polynomial evaluation".
I.B.M. J. Res. Develop. (1962), 239-245.
- (90) ESTRIN, G., "Organization of computer systems. The fixed plus variable
structure computer."
Proc. Western Joint Comput. Conf. (1960) 33-40.
- (91) GILMORE, "Structuring of parallel algorithms.
J. ACM, vol.15, n° 2 (april 1968) 176-192.
- (92) GROGINSKY, H.L. WORKS, GA., "A pipeline fast fourier transform".
IEEE Trans on Computers vol.c19, n°11,(November 1970), 1015-1019.
- (93) HYAFIL, L., and KUNG. HT., "Bounds on the speed up of parallel
evaluation of recurrences.
Computer Science Depart. Report. Carnegie Mellon University (1975).
- (94) HYAFIL, L., and KUNG, HT., "The complexity of parallel evaluation
of linear recurrence".
7th ACM symposium on theory of computing (may 1975).
- (95) HELLERMAN, "Parallel processing of algebraic expressions".
IEEE Trans. on E.C., vol. EC-15, n° 1, (feb. 1966) 82-91.
- (96) KARP, RM., MILLER, RE., "Parallel program schemata".
J. Comput. Systems. Sci., 3 (1969), 147-195.
- (97) KARP, RM., MIRANKER WL., "Parallel minimax search for a maximum".
Journal of Combinatorial theory 4, 19-35 (1968).

- (98) KUCK, NJ. and MARUYAMA, KM., "The parallel evaluation of arithmetic expressions of special forms".
Report RC 4276, I.B.M. Research Center, Yorktown Heights, (1973).
- (99) KUNG, H.T., "New algorithms and lower bounds for the parallel evaluation of certain rational expressions".
Proc. 6th Annual ACM Symposium on theory of computing; (1974), 323-333.
- (100) LAFON, JC., "Calcul algébrique en parallèle".
Colloque d'Analyse Numérique, Super-Besse, (mai-juin 1970).
- (101) MARUYAMA, KM., "On the parallel evaluation of polynomials".
IEEE Trans c-22 (1973), 2-5.
- (102) MIRANKER, WL., "A survey of parallélism in numerical analysis".
SIAM Review 13 (1971), 524-547.
- (103) MIRANKER, WL, and LINIGER, W., "Parallel methods for the numerical integratinon of ordinary differential equations". (nov. 1966).
- (104) MIRANKER, WL., "Parallel methods for approximating the root of a function". I.B.M. J. Res. Developp. (may 1969).
- (105) MUNRO and PATERSON, M. "Optimal algorithms for parallel polynomial evaluation".
Report RC 3497, IBM Res. Center, Yorktown Heights (1971).
- (106) PEASE, MC., "An adaptation of the fast fourier transform for parallel processing".
J ACM vol.15, n° 2 (april 1968), 252-264.
- (107) PEASE, MC., "Organisation of large scale processors".
J. ACM vol.16, n°3, (july 1969), 474-482.

- (108) PEASE, MC., "Matrix inversion using parallel processing".
J. ACM vol.14, n°4 (oct. 1967) 757-764.
- (109) WILDE, DJ. and AVRUEL, M., "Optimal search for a maximum with sequences of simultaneous function evaluations".
Management Science vol.12, n° 9, (may 1966) 722-731.

VI Problèmes de calculs algébriques

Les références citées ci-dessous complètent celles qui sont données à la fin de chaque chapitre des parties B et C.

- (110) BELAGA, EC., "Some problems in the computation of polynomials",
Dokl, Akad Nauk, SSSR, 123, (1958), 775-777.
- (112) BORODIN, A. "Horner's rule is uniquely optimal".
Theory of machines and computations, ed. by Z. Kohavi and A. Praz,
Academic press, (1971).
- (113) BORODIN and MUNRO, I. "Evaluation of polynomials at many points"
Information processing letters, vol.1, N° 2, (july 1971).
- (114) BRENT, R.P., "Algorithms for matrix multiplication".
Stan. CS 70-157.
- (115) BRENT, RP. ; KUNG, HT. " $O((n \log n)^{3/2})$ algorithmes for composition
and reversion of power series".
Carnegie Mellon University, T.R. (may 1975).
- (116) DE POLIGNAC, C. "Methodes optimales du calcul des produits de
matrices." Thèse doct-ing. Grenoble (juin 1970).
- (117) FISCHER and MEYER, A. "Boolean matrix multiplication and transi-
tive closure".
Conf. Rec. 12th Annual IEEE Symp. on switching and Automata theory.
(oct. 1971) 129-131.

- (118) FURMAN, M. "Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph".
Dokl. Akad. Nauk. SSSR 194 (1970), 524- transl. in English in Soviet Math. Dokl 11 (1970) 1252.
- (119) KEDEM, Z. , "Studies in algebraic computational complexity"
ph.D thesis, Israel Institute of Technology, Haifa, (april 1973).
- (120) KIRKPATRICK, D. "On the number of additions required to compute certain functions". Proc. ACM.SIGACT conf. (may 1972).
- (121) LADERMAN , "A noncommutative algorithm for multiplying 3 x 3 matrices using 23 multiplications". preprint (1975).
- (122) MORGENSTERN, J. "Linear algorithms and tangent algorithms".
IFIP proc. (august 1974).
- (123) MORGENSTERN, J. "Note on a lower bound of the linear complexity of the fast fourier transforms". J. ACM, (april 1973).
- (124) MUNRO, I. "Efficient determination of the transitive closure of a directed graph".
Information processing letters, 1 (1971), 56-58.
- (125) MUNRO and BORODIN , "Efficient evaluation of polynomials forms".
J. of Computer and System Sciences, (dec. 1972).
- (126) OSTROWSKI, AM. , "On two problems in abstract algebra connected with Horner's rule".
Studies presented to R von Mises, Academic Press, New-York, (1954)
40-48.
- (127) PAN, V, Ya , "Methods of computing values of polynomials".
Russian Math. Surveys 21 (1966).

- (128) PATERSON, M., "Complexity of product and closure algorithms for matrices".
Proc. Int. Congress of Math - Vancouver, (1974).
- (129) PATERSON, M. ; STOCKMEYER, L. " Bounds on the evaluation time of rational functions".
Proc. 12th annual IEEE Symposium on Switching and Automata theory (oct. 1971) 140-143.
- (130) PATERSON, M. "Complexity of monotone networks for boolean matrix product". Theoretical Computer Science 1, 1(1975) 13-20.
- (131) RABIN, M., WINOGRAD, S., "Fast evaluation of polynomials by rational preparation".
I.B.M. Technical Report RC3645, (dec. 1971).
- (132) SHAW, M., TRAUB, JF., "On the number of multiplications for the evaluation of a polynomial and some of its derivatives".
Carnegie Mellon University T.R. (1972), also in JACM, vol.11, n° 1 (january 1974), 161-167.
- (133) SCHONHAGE, A., STRASSEN, V. " Fast multiplication of large numbers".
Computing 7 (1971), 281-292.
- (134) SPIEB, J., "Untersuchung zur implementierung der algorithmen von s. Winograd und v. Strassen zur Matrizen-multiplikation.
GWDG-bericht nR 10 (august 1974) Göttingen.
- (135) STRASSEN, V., "Polynomials with rational coefficients which are hard to compute".
unpublished manuscript (Summer 1970).
- (136) VALIANT, L., "general context free recognition in less than cubic time". J. CSS, 10, 2 (1975), 308-315.

- (137) VAN LEEUWEN, J. " On the non vanishing terms in a product of multivariate polynomials".
Stichting Mathematisch centrum. Amsterdam, T.R. (may 1975).
- (138) WINOGRAD, V., "On the number of multiplications necessary to compute certain functions".
Comm. Pures Appl. Math. 23 (1970) 165-179.

PARTIE B



COMPLEXITE DU CALCUL

DE PLUSIEURS FORMES BILINEAIRES

PLAN DE LA PARTIE B

Introduction

Chapitre BI - Classe des algorithmes utilisés. Critère d'optimalité.

1. Notations.
2. Calcul de p formes bilinéaires.
3. Algorithmes de calculs algébriques.
4. Critère de coût utilisé.
5. Algorithmes optimaux.

Chapitre BII - Caractérisation du coût minimal et des algorithmes optimaux.

1. Coût minimal dans le cas commutatif.
2. Coût minimal dans le cas non commutatif.
3. Interprétations avec la notion de rang tensoriel.
4. Algorithmes de coût minimal.

Chapitre BIII - Etude de la notion de rang tensoriel.

1. Principales propriétés.
2. Rang d'un tenseur.
3. Condition pour que p matrices régulières aient un rang tensoriel minimal.
4. Rang tensoriel d'un espace de matrices contenant au moins une matrice régulière.
5. Cas du produit de deux matrices.
6. Calcul approché du rang tensoriel.

Chapitre BIV - Bases tensorielles. Applications.

1. Un théorème d'optimalité.
2. Espace des matrices cycliques.
 - a/ Unicité de la base tensorielle.
 - b/ Application au produit de convolution.
 - c/ Inversion d'une matrice cyclique.

3. Espace des matrices de Hankel (ou de Toeplitz)
 - a/ Bases tensorielles.
 - b/ Application au calcul du produit de deux polynomes.
 - c/ Inversion en $O(n)$ multiplications d'une matrice de Toeplitz triangulaire inférieure.
 - d/ Division de deux polynomes.
4. Espaces des matrices ayant des symétries particulières.
 - a/ Matrices symétriques.
 - b/ Matrices centrosymétriques.
 - c/ Matrices horizontales (verticales) symétriques.
 - d/ Matrices roto-symétriques droites (gauches).
 - e/ Matrices à éléments complexes.

Chapitre BV - Minoration du rang tensoriel. Résultats d'optimalité .

1. Principes généraux de minoration du rang tensoriel
 - a/ Résultats basés sur l'étude d'un espace de matrices.
 - b/ Utilisation des transformations invariantes d'une forme trilinéaire.
2. Produit de deux matrices.
 - a/ Borne inférieure dans le cas général.
 - b/ Optimalité du résultat de STRASSEN.
3. Produit optimal de deux quaternions.
4. Produit vectoriel de deux vecteurs. Produit de LIE de deux matrices.
5. Rang tensoriel de l'ensemble des matrices anti-symétriques.

Conclusions.

INTRODUCTION

Dans cette partie, on se propose d'étudier la complexité "multiplicative" du calcul de plusieurs formes bilinéaires. Cette complexité "multiplicative" est caractérisée par le nombre minimal, m par exemple, de multiplications que doit nécessairement employer tout algorithme de calcul de ces formes bilinéaires.

Cette étude est évidemment motivée par le fait que de nombreux calculs se présentent sous la forme de l'évaluation de plusieurs formes bilinéaires particulières. On peut citer, par exemple, les cas du produit de deux polynômes, du produit de convolution de deux vecteurs, du produit de deux quaternions, et évidemment le cas du produit de deux matrices. En particulier, la découverte par STRASSEN (1), d'une méthode de calcul du produit de deux matrices carrées n, n en $n^{\log_2 7}$ multiplications a subitement révélé ce nouveau domaine de recherches possibles, concernant pourtant des calculs à priori simples.

Pour un calcul de plusieurs formes bilinéaires particulières il s'agit, d'une part d'obtenir une minoration (la meilleure possible) du nombre minimal de multiplications nécessaires pour évaluer ces formes bilinéaires, et d'autre part de construire un algorithme utilisant le moins de multiplications possible. L'idéal est d'arriver au cas où la minoration obtenue est égale au nombre de multiplications utilisées par un algorithme connu, qui sera donc qualifié d'optimal.

Pour développer cette étude, il est tout d'abord nécessaire de définir la classe des algorithmes que l'on va utiliser ainsi que le critère de coût utilisé. Ceci sera fait dans le premier chapitre.

Dans le second chapitre on donne une caractérisation du coût minimal de calcul de p formes bilinéaires à la fois dans le cas où l'on utilise la commutativité et dans le cas où la commutativité n'est pas utilisée. On relie ce coût minimal à la notion de rang tensoriel de p matrices.

Le chapitre suivant est consacré à l'étude de la notion de rang tensoriel de p matrices. On y donne les principales propriétés du rang tensoriel d'un ensemble de matrices. On montre en particulier que l'on peut interpréter ce rang tensoriel comme le rang d'un tenseur d'ordre trois associé à ces matrices. On donne aussi la condition pour que p matrices dont une au moins soit régulière aient un rang tensoriel égal à leur dimension. On examine ensuite dans cette optique le problème du produit de deux matrices. Enfin, on termine ce chapitre par l'exposé d'une méthode de calcul approché du rang tensoriel.

Le quatrième chapitre est consacré à l'étude des espaces de matrices dont la dimension est égale à leur rang tensoriel. On montre que dans ce cas la connaissance de toutes les bases tensorielles de l'espace permet de construire tous les algorithmes optimaux de calcul des formes bilinéaires correspondantes. On montre que l'on peut aussi déterminer tous les algorithmes optimaux pour le calcul du produit de deux polynômes ainsi que pour le produit de convolution de deux vecteurs. On donne ensuite d'autres exemples d'espaces dont le rang tensoriel est égal à la dimension.

Le cinquième chapitre regroupe plusieurs preuves d'optimalité d'algorithmes. On commence d'abord par donner quelques procédés généraux de minoration du rang tensoriel. On donne ensuite une nouvelle preuve de l'optimalité de la méthode de STRASSEN, une démonstration de l'optimalité du produit de deux quaternions en huit multiplications, de même qu'une démonstration de l'optimalité du calcul du produit vectoriel de deux vecteurs de R^3 en cinq multiplications ainsi qu'une généralisation de ce résultat.

REFERENCE

- (1) STRASSEN : "Gaussian elimination is not optimal".
Numerisch. Math. 13, (1969), 354-356.

CHAPITRE BI

CLASSE DES ALGORITHMES UTILISES

CRITERE D'OPTIMALITE

PLAN

1. Notations
2. Calcul de p formes bilinéaires
3. Algorithmes de calculs algébriques
4. Critère de coût utilisé
5. Algorithmes optimaux.

Dans le premier paragraphe de ce chapitre, on donne les principales notations et définitions utilisées ultérieurement. On définit ensuite, d'abord, la notion de calcul de plusieurs formes bilinéaires, puis la classe des algorithmes de calculs algébriques à laquelle tous les algorithmes considérés dans cette partie devront appartenir. A tout algorithme de cette classe, on associe un coût, égal au nombre de multiplications (n'utilisant pas des scalaires) qu'il emploie. Enfin, on termine ce chapitre par la définition de l'algorithme optimal et du coût minimal associé à un calcul de p formes bilinéaires particulières.

1 . NOTATIONS

On désigne par K un corps. On note par $+$ et \cdot les deux opérations (addition et multiplication) qui confèrent à l'ensemble K la structure de corps. Si a désigne un élément de K , alors on note par $-a$ son inverse pour l'addition, et par a^{-1} son inverse pour la multiplication. L'élément neutre pour l'addition sera noté 0 , et l'élément neutre pour la multiplication 1 . Sauf indication expresse du contraire, l'opération \cdot de K sera toujours supposée commutative. Si cela n'est pas le cas, K sera spécifié être un corps non commutatif. (De la même façon, on parlera de l'anneau A , ou de l'anneau non commutatif A). On fera toujours implicitement l'hypothèse que le corps K n'est pas de caractéristique égale à deux.

Exemples

- Les principaux corps utilisés dans la suite sont les suivants :
- corps de Galois noté $GF(p)$, (p nombre premier),
 - corps des rationnels noté Q ,
 - corps des réels noté R ,
 - corps des complexes noté C .

$M_{m,n}(K)$ désigne l'espace vectoriel des matrices à m lignes et n colonnes à éléments appartenant au corps K . $M_{m,m}(K)$ possède une structure d'anneau non commutatif quand il est muni des deux opérations addition et multiplication de deux matrices (opérations notées $+$ et \cdot ,

en général, si une ambiguïté pouvait survenir on utiliserait les deux symboles $+_m$ et \cdot_m . On note également par \cdot l'opération externe de $\mathcal{M}_{m,n}(K)$ (multiplication d'une matrice par un élément de K). $\mathcal{M}_{m,m}(K)$ possède une structure d'algèbre sur le corps K (Il suffirait d'ailleurs que K soit un anneau avec un élément unité, dans ce cas on parlerait non pas de l'espace vectoriel $\mathcal{M}_{m,m}(K)$ mais d'un module sur K).
Toute matrice A de $\mathcal{M}_{m,m}(K)$ peut s'écrire sous la forme :

$$A = \sum_{i,j}^m A_{i,j} E_{i,j} ,$$

les matrices $E_{i,j}$ ayant comme unique élément non nul, l'élément en position (i,j) égal à un \cdot . Ces m^2 matrices forment la base canonique de $\mathcal{M}_{m,m}(K)$.

Dans la suite, on notera par A^t la transposée de la matrice A , par $\det(A)$ son déterminant, par $\text{Rang}(A)$ son rang. Enfin, on note par \oplus la somme directe de deux matrices, et par \otimes le produit de Kronecker (ou produit direct) de deux matrices. Si A et B sont deux matrices on a :

$$A \oplus B = \begin{pmatrix} A & O \\ O & B \end{pmatrix} ,$$

$$C = A \otimes B \quad \text{avec} \quad C_{ij,kh} = A_{i,k} \cdot B_{j,h} .$$

Soient x_1, \dots, x_n n indéterminées formant un monoïde libre à élément unité et soit R un anneau à élément unité tel que : $x_i \notin R$ ($i=1, \dots, n$). On note alors par $R[x_1, \dots, x_n]$ l'anneau obtenu à partir de R par adjonction des n indéterminées x_1, \dots, x_n . L'anneau R est l'anneau des coefficients (si à la place de l'anneau R on a un corps K on parlera du corps des scalaires). Un élément de $R[x_1, \dots, x_n]$ sera appelé polynôme. Soit $P(X)$ un élément de $R[x_1, \dots, x_n]$ ($X^t = (x_1, \dots, x_n)$). Si M est un module sur R , alors si on substitue dans $P(X)$, aux indéterminées x_1, \dots, x_n , n éléments a_1, \dots, a_n de M , on obtiendra un élément de M que l'on notera : $P(a_1, \dots, a_n)$. On note par $+$ et \cdot les deux opérations de $R[X]$. Si K est un corps, alors on note par $K(X)$ ou $K(x_1, \dots, x_n)$, le corps des fractions rationnelles en les indéterminées x_1, \dots, x_n (corps quotient de $K[X]$).

Si les indéterminées x_1, \dots, x_n forment un monoïde commutatif, alors $R[x_1, \dots, x_n]$ sera un anneau commutatif.

Un élément du monoïde engendré par x_1, \dots, x_n est appelé monome. Tout élément de $R[X]$ peut s'écrire comme une combinaison de monomes les coefficients étant des éléments de R .

Définition 1

/ On dira que deux éléments $P_1(X)$ et $P_2(X)$ de $R[X]$ sont égaux si, dans leur développement sous forme de monomes, apparaissent les mêmes monomes affectés des mêmes coefficients. On écrira alors :

$$P_1(X) \equiv P_2(X) . \quad /$$

Remarque 1

/ Il est clair que si $P_1(X)$ et $P_2(X)$ sont égaux, alors $\forall X_0 \in M$ (M module sur A) on aura :

$$P_1(X_0) = P_2(X_0) \quad (\text{égalité de deux éléments de } M) . \quad /$$

Par contre, la réciproque est fautive, comme le montre l'exemple suivant :

On prend $K = GF(p)$ (p le nombre premier). On a alors :

$$\forall a \in K \quad a^p = a \quad (\text{si } a \neq 0 \quad a^{p-1} = a^{-1}, \text{ si } a = 0 \quad a^p = 0) .$$

Considérons les deux éléments suivants de $K[x]$:

$$P_1(x) = x^p \quad , \quad P_2(x) = x .$$

En tant qu'éléments de $K[x]$, $P_1(x)$ et $P_2(x)$ sont distincts.

Par contre, si on substitue à x un élément quelconque de K , x_0 , alors :

$$P_1(x_0) = P_2(x_0) .$$

Remarque 2

Dans le cas de plusieurs indéterminées, cette notion d'égalité dépendra de la commutativité ou de la non-commutativité de $R[X]$:

Si $R[x_1, x_2]$ est non-commutatif ($x_1x_2 \neq x_2x_1$)

alors on aura :

$$(x_1+x_2)(x_1-x_2) \equiv x_1^2+x_2^2x_1-x_1x_2-x_2^2 \quad ,$$

par contre :

$$(x_1+x_2)(x_1-x_2) \neq x_1^2-x_2^2 \quad .$$

Si $R[x_1, x_2]$ est commutatif, alors on a :

$$(x_1+x_2)(x_1-x_2) \equiv x_1^2-x_2^2 \quad .$$

2 . CALCUL DE p FORMES BILINEAIRES

K désigne un corps B_1, \dots, B_p sont p matrices de $\mathcal{M}_{m,n}(K)$,
 $x_1, \dots, x_m, y_1, \dots, y_n$ sont $m+n$ indéterminées n'appartenant pas à K .
 Dans toute la suite, on désignera par $P(B_1, \dots, B_p)$ le problème suivant :

Calculer, à partir de $K \cup \{x_1, \dots, x_m, y_1, \dots, y_n\}$ p éléments
 P'_1, \dots, P'_p de $K[x_1, \dots, x_m, y_1, \dots, y_n]$ tels que l'on ait :

$$P'_i \equiv X^t B_i Y \quad i=1, \dots, p \quad .$$

Remarque 3

On note $b_{i,j,k}$ l'élément de la matrice B_k en position i, j ($i \geq m, j \geq n$).
 L'expression $X^t B_k Y$ représente un polynôme homogène de degré deux en les
 $m+n$ indéterminées $x_1, \dots, x_m, y_1, \dots, y_n$.

On peut écrire de façon explicite :

$$X^t B_k Y \equiv \sum_{i=1, j=1}^{m, n} b_{i,j,k} X_i Y_j \quad .$$

Le problème $\mathcal{P}(B_1, \dots, B_p)$ est celui du calcul de p éléments de
 $K[x_1, \dots, x_m, y_1, \dots, y_n]$ égaux à ces p polynômes, mais c'est aussi celui
 de l'évaluation simultanée, en un point quelconque de $K^m \times K^n$, des p
 formes bilinéaires f_1, \dots, f_p définies par les matrices B_1, \dots, B_p de la
 manière suivante :

$$f_i : \quad K^m \times K^n \rightarrow K$$

$$\forall (X_0, Y_0) \in K^m \times K^n \quad (X_0, Y_0) \mapsto X_0^t B_i Y_0 \quad i=1, \dots, p \quad .$$

Si le corps K est infini, dire que l'on veut évaluer les p formes bilinéaires f_1, \dots, f_p en un point arbitraire (générique) de $K^m \times K^n$ revient à dire que l'on cherche une formule de calcul des p polynomes $X^t B_i Y$ $i=1, \dots, p$ les x_i et les y_i étant des indéterminées qui donc ne vérifient sur K aucune relation algébrique particulière.

Remarque 4

L'écriture explicite des polynomes $X^t B_i Y$ sous la forme :

$$X^t B_k Y \equiv \sum_{i,j}^{m,n} b_{i,j,k} X_i Y_j \quad k=1, \dots, p$$

fournit une première méthode de calcul de p_1, \dots, p_p .

Dans la suite, on veut trouver parmi l'ensemble de toutes les méthodes de calcul de $P(B_1, \dots, B_p)$ celle qui est la meilleure en fonction du nombre de multiplications "générales" utilisées.

Le but du paragraphe suivant est de préciser la classe des algorithmes (méthodes) de calcul de $P(B_1, \dots, B_p)$ ainsi que la notion de multiplication "générale".

Tout ce qui vient d'être dit s'applique évidemment au cas où K est un anneau et au cas où on évalue les formes bilinéaires en un élément d'un module M sur K . Pour simplifier, on expose ces définitions pour le cas d'un corps.

3 . ALGORITHMES DE CALCULS ALGEBRIQUES

La classe des algorithmes qui va être utilisée dans la suite est identique à la classe des algorithmes de calculs algébriques définie par OSTROWSKI (5) et WINOGRAD (6) et développée par STRASSEN (3,4) dans un cadre plus général.

a/ Ensemble \mathcal{A}_p des algorithmes résolvant $P(B_1, \dots, B_p)$.

Un algorithme A sera considéré comme algorithme résolvant le problème $P(B_1, \dots, B_p)$ s'il définit, de la manière suivante, le calcul de p éléments p'_1, \dots, p'_p de $K[x_1, \dots, x_m, y_1, \dots, y_n]$ tels que :

$$X^{t_{B_i}} Y \equiv p'_i \quad (i=1, \dots, p) /$$

A doit consister en un ensemble fini I d'instructions. L'ensemble I doit être totalement ordonné, et mis en bijection avec l'ensemble des entiers de 1 à card I (numérotation des instructions). La kème instruction d'un tel algorithme doit avoir l'un des deux types suivant :

type 1 $\langle k \rangle \leftarrow \langle i \rangle T \langle j \rangle$, $(i \leftarrow k, j \leftarrow k)$

type 2 $\langle k \rangle \leftarrow \langle a \rangle$.

Le symbole T désigne l'une des opérations + et . de K ou de $K[x_1, \dots, x_m, y_1, \dots, y_n]$. Dans une instruction de type 2, le symbole a peut être remplacé par tout élément de $K \cup \{x_1, \dots, x_m, y_1, \dots, y_n\}$. Une instruction de type 2 est en fait une instruction d'affectation. On désigne par : $\langle k \rangle$ le résultat de la kème instruction (le résultat de la kème instruction, $\langle i \rangle T \langle j \rangle$ par exemple, est mis dans la kème mémoire, dont le contenu est donc $\langle k \rangle$).

Enfin, un tel ensemble I d'instructions sera un algorithme de calculs des p polynomes $X^{t_{B_i}} Y$ ($i=1, \dots, p$) s'il existe p entiers i_1, \dots, i_p tels que :

$$\langle i_k \rangle \equiv X^{t_{B_k}} Y \quad , \quad (k=1, \dots, p)$$

On peut donc parler maintenant de l'ensemble des algorithmes de calcul des p polynomes $X^{t_{B_i}} Y$. On note par \mathcal{A}_p cet ensemble.

4 . CRITERE DE COUT UTILISE.

A tout algorithme A de l'ensemble \mathcal{A}_p précédemment défini, on va associer un coût, c'est-à-dire un entier positif ou nul que l'on note C_A et qui est défini de la manière suivante :

Le coût de A sera la somme du coût de chacune de ses instructions.

$$C_A = \sum_{i=1}^N C_i \quad \begin{array}{l} (C_i \text{ coût de la } i\text{ème instruction}) \\ (N = \text{card } I). \end{array}$$

Le coût d'une instruction est définie ainsi :

- le coût de toute instruction de type 2 est pris égal à zéro,
- le coût de toute instruction de type 1 est nul si $T = +$; si $T = .$ ce coût vaut 1 si aucun des deux arguments n'est un élément de K, et vaut zéro dans le cas contraire.

Une instruction de type 1 avec $T = .$ et aucun des deux arguments n'appartenant à K définit une multiplication "générale".

Le coût d'un algorithme A de \mathcal{A} est donc en fait le nombre de multiplications "générales" qu'il utilise.

Exemple :

Soit $K = \mathbb{R}$. On prend $m = 2$ et $n = 2$ et on considère les deux matrices B_1 et B_2 de $\mathcal{M}_{2,2}(\mathbb{R})$ suivantes :

$$B_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} .$$

Le problème $P(B_1, B_2)$ est celui du calcul de deux éléments p_1' et p_2' de $K[x_1, x_2, y_1, y_2]$ tels que l'on ait :

$$p_1' \equiv X^t B_1 Y \quad \text{et} \quad p_2' \equiv X^t B_2 Y .$$

On doit donc avoir :

$$\begin{array}{l} p_1' \equiv x_1 y_1 - x_2 y_2 \quad , \\ p_2' \equiv x_1 y_2 + x_2 y_1 \quad . \end{array}$$

Ceci peut donc s'interpréter comme le calcul du produit des deux nombres complexes $(x_1 + i x_2)$, $(y_1 + i y_2)$.

Ce calcul nécessite avec les formules usuelles quatre multiplications "générales". L'algorithme suivant montre que l'on peut calculer ce produit avec trois multiplications générales seulement :

$$\begin{array}{ll}
 \langle 1 \rangle \leftarrow x_1 & , \\
 \langle 2 \rangle \leftarrow y_1 & , \\
 \langle 3 \rangle \leftarrow x_2 & , \\
 \langle 4 \rangle \leftarrow y_2 & , \\
 \langle 5 \rangle \leftarrow \langle 1 \rangle \cdot \langle 2 \rangle & , \quad \text{coût un } (x_1 \cdot y_1) \quad , \\
 \langle 6 \rangle \leftarrow \langle 3 \rangle \cdot \langle 4 \rangle & , \quad \text{coût un } (x_2 \cdot y_2) \quad . \\
 \langle 7 \rangle \leftarrow \langle 5 \rangle - \langle 6 \rangle & , \quad (x_1 y_1 - x_2 y_2) \\
 \langle 8 \rangle \leftarrow \langle 1 \rangle + \langle 3 \rangle & , \quad (x_1 + x_2) \\
 \langle 9 \rangle \leftarrow \langle 2 \rangle + \langle 4 \rangle & , \quad (y_1 + y_2) \\
 \langle 10 \rangle \leftarrow \langle 8 \rangle \cdot \langle 9 \rangle & , \quad \text{coût un } (x_1 + x_2)(y_1 + y_2) \\
 \langle 11 \rangle \leftarrow \langle 10 \rangle - \langle 5 \rangle - \langle 6 \rangle & , \quad (x_1 + x_2)(y_1 + y_2) - x_1 y_1 - x_2 y_2
 \end{array}$$

Il est clair que cet algorithme est bien un algorithme du type précédemment défini. Il consiste en une suite de 11 instructions, dont trois seulement ont un coût unité. Il permet donc le calcul du produit de deux nombres complexes en trois multiplications. L'instruction 7 donne la partie réelle et l'instruction 11 la partie imaginaire du nombre $(x_1 + i x_2)(y_1 + i y_2)$. On peut en fait écrire plus brièvement :

$$\begin{array}{l}
 (1) \quad x_1 y_1 - x_2 y_2 \equiv x_1 y_1 - x_2 y_2 \quad , \\
 \quad \quad x_1 y_2 + x_2 y_1 \equiv (x_1 + x_2)(y_1 + y_2) - x_1 y_1 - x_2 y_2 \quad .
 \end{array}$$

Remarque :

Cet algorithme ne suppose pas la commutativité de $K[x_1, x_2, y_1, y_2]$. Il est donc valable pour multiplier $(A_1 + i A_2)$ par $(B_1 + i B_2)$ quand A_1, A_2, B_1, B_2 sont des matrices de dimensions appropriées.

5 . ALGORITHMES OPTIMAUX

Soit \mathcal{A}_p la classe des algorithmes résolvant le problème $P(B_1, \dots, B_p)$. A tout algorithme A de \mathcal{A}_p on sait associer un coût C_A égal au nombre de multiplications "générales" qu'il utilise. On peut donc se poser les deux questions suivantes :

- quel est le coût minimum ?
- quel est l'algorithme qui utilise le nombre minimum de multiplications générales ?

On pose

$$C_* = \min_{A \in \mathcal{A}_p} C_A$$

(C_* existe forcément puisque l'on ne considère qu'un ensemble d'entiers positifs ou nuls).

Un algorithme A_* de \mathcal{A}_p sera dit optimal si :

$$C_{A_*} = C_* .$$

Il faut remarquer de suite, que l'unicité de l'algorithme optimal n'est pas assurée.

Dans la suite, pour un problème $P(B_1, \dots, B_p)$, on cherchera à encadrer C_* par deux entiers m et M tels que l'écart $M-m$ soit le plus petit possible.

Si $m \leq C_* \leq M$, on dira que m est une borne inférieure de C_* , et que M en est une borne supérieure. Une borne supérieure est obtenue en exhibant un algorithme de coût M de \mathcal{A}_p . Par contre une borne inférieure peut être obtenue par des méthodes non constructives.

Démontrer l'optimalité d'un algorithme de \mathcal{A}_p de coût C_A revient à démontrer que l'on a forcément :

$$C_* \geq C_A .$$

La raison pour laquelle on ne considère que le nombre de multiplications "générales" est surtout due au fait que l'on ne sait pratiquement rien dire si l'on tient compte de plusieurs types d'opérations arithmétiques. Il faut cependant remarquer les trois choses suivantes :

- a/ Les indéterminées $x_1, \dots, x_m, y_1, \dots, y_n$ peuvent représenter autre chose que des éléments de K . Par exemple dans le cas non commutatif, on peut prendre $x_i \in M_{n,n}(K)$, $y_i \in M_{n,n}(K)$. Dans ces conditions il est parfaitement licite de négliger le coût des opérations du type $+$ et celles du type $a.x$ avec $a \in K$ par rapport au coût du produit de deux matrices.
- b/ Si $X \in R^m$ et $Y \in R^n$, il est également licite de négliger le coût d'une opération du type $a.x$ avec $a \in N$ connu d'avance et x quelconque. Par contre, le coût d'une opération du type x_1+x_2 et le coût de celle du type $x_1.x_2$ est en général seulement dans un rapport fixe; si t_+ désigne le temps d'exécution moyen de l'addition de deux réels en simple précision, et t désigne le coût moyen d'une multiplication de deux tels nombres on a :

$$\frac{t}{t_+} = r .$$

De ce rapport r dépend alors la performance réelle d'un algorithme.

- c/ Un résultat de STRASSEN (4) montre que pour calculer plusieurs polynômes de degré deux de $K[X,Y]$ on n'a pas intérêt à effectuer une partie des calculs dans $K(X,Y)$, c'est-à-dire que l'on ne peut diminuer le coût du calcul en utilisant la division de deux polynômes. (Le coût de cette division étant égal à celui d'une multiplication). Il est donc licite de ne considérer que l'opération multiplication pour le calcul de plusieurs formes bilinéaires.

REFERENCES SUR LE CHAPITRE BI

a/ Articles

- (2) STRASSEN, V., "Berechnung und programm I"
Acta Informatica- 1 - (1972), 320-335.
- (3) STRASSEN, V., "Berechnung und programm II".
Acta Informatica- 2 - (1973), 64-79.
- (4) STRASSEN, V., "Vermeidung von divisionen".
Crelle J. Für die Reine und angew. Mathematik 1973.
- (5) OSTROWKI, "On two problems in abstract algebra connected with
linear's rule".
Studies presented to R von Mises. Acad. Press, New-York (1954).
- (6) WINOGRAD, S., "On the number of multiplications required to
compute certain function".
Communications on pure and Applied Mathematics.
Vol XXIII, (1970), 165-179.

b/ Livres

- (7) LANG, S., "Algebra".
Addison Wesley publishing Company.

CHAPITRE BII

CARACTERISATION DU COÛT MINIMAL ET

DES ALGORITHMES OPTIMAUX

PLAN

1. Coût minimal dans le cas commutatif
2. Coût minimal dans le cas non commutatif
3. Interprétations avec la notion de rang tensoriel
4. Algorithmes de coût minimal.

On garde les notations précédentes. Le but cherché ici est de donner une caractérisation des algorithmes de \mathcal{A}_p qui utilisent le nombre minimal de multiplications "générales" pour le calcul des p polynômes $X^t B_i Y$ ($i=1, \dots, p$). Le résultat dépendant de la commutativité ou de la non-commutativité de $K[x_1, \dots, x_m, y_1, \dots, y_n]$, on obtient en fait deux théorèmes distincts (K est un corps).

1. COÛT MINIMAL DANS LE CAS COMMUTATIF.

Théorème 1.

/ Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est commutatif, le nombre minimal de multiplications "générales" que doit employer tout algorithme de \mathcal{A}_p pour calculer $X^t B_i Y$ ($i=1, \dots, p$) est égal au plus petit entier q tel que l'on puisse satisfaire les conditions suivantes :

$$(1) \quad B_i = \sum_{j=1}^q \alpha_j^i (A_1^j B_2^{j^t} + A_2^j B_1^{j^t}) \quad i=1, \dots, p$$

$$(2) \quad \sum_{j=1}^q \alpha_j^i (A_1^j A_2^{j^t} + A_2^j A_1^{j^t}) = 0 \quad \alpha_j^i \in K$$

$$(3) \quad \sum_{j=1}^q \alpha_j^i (B_1^j B_2^{j^t} + B_2^j B_1^{j^t}) = 0$$

avec

$$A_1^j, A_2^j \in K^m \quad (j=1, \dots, q), \quad B_1^j, B_2^j \in K^n \quad (j=1, \dots, q) . /$$

□ Soit A un algorithme de \mathcal{A}_p . Si la même instruction de cet algorithme a un coût unité, on doit pouvoir écrire :

$$\langle k \rangle \leftarrow \langle i \rangle . \langle j \rangle, \quad i, j \leq k .$$

Cette instruction devant correspondre à une multiplication générale, $\langle i \rangle$ et $\langle j \rangle$ doivent être égaux à des éléments de $K[x_1, \dots, x_m, y_1, \dots, y_n]$ de degré 1 au moins. On peut donc écrire :

$$\langle i \rangle \equiv a^i + P_1^i + P_r^i, \quad a^i \in K \quad P_1^i, P_r^i \in K[X, Y]$$

$$\langle j \rangle \equiv b^j + P_1^j + P_r^j, \quad b^j \in K \quad P_1^j, P_r^j \in K[X, Y] .$$

P_1^i et P_1^j désignent deux polynomes homogènes de degré 1 de $K[X,Y]$,

P_r^i et P_r^j désignent deux polynomes dont tous les termes sont de degré supérieur à un si $P_r^i \neq 0$ et $P_r^j \neq 0$

On peut écrire :

$$\langle k \rangle \equiv (a^i + p_1^i + p_r^i) \cdot (b^j + p_1^j + p_r^j).$$

Les seuls termes de degré deux dans $\langle k \rangle$, proviennent de l'expression :

$$p_1^i p_1^j + a^i p_r^j + b^j p_r^i.$$

Si on désigne par p_1^k le polynome $p_1^i p_1^j$, on voit que le polynome homogène de degré deux que permet d'obtenir une instruction de coût unité est la somme du polynome p_2^k et d'une combinaison linéaire de polynomes du même type précédemment calculés, par des instructions de coût un.

Désignons donc par p_2^k ($k = 1, \dots, q$), les q polynomes homogènes de degré deux, produits de deux formes linéaires, qui apparaissent dans les q instructions de coût un.

Toute instruction de coût nul de l'algorithme A, ne peut donner qu'une combinaison linéaire de ces q polynomes, et on vient de voir, d'autre part, que la $k^{\text{ème}}$ instruction de coût un ne peut donner qu'un polynome homogène de degré deux du type :

$$p_2^k + \sum_{i=1}^{k-1} \alpha_i p_2^i \quad (\alpha_i \in K \quad i = 1, \dots, k-1).$$

Par conséquent, tous les polynomes homogènes de degré deux que peut calculer l'algorithme A seront tous de la forme :

$$\sum_{i=1}^q \alpha_i p_2^i \quad \text{avec} \quad \alpha_i \in K, \quad i = 1, \dots, q.$$

En particulier, l'algorithme A calculant les p polynomes $X^t B_i Y$, $i=1, \dots, p$, on doit avoir :

$$(4) \quad X^t B_i Y \equiv \sum_{j=1}^q \alpha_j^i p_2^j \quad i = 1, \dots, p, \quad \alpha_j^i \in K.$$

Ceci montre qu'à tout algorithme A de coût égal à q on peut associer un algorithme A' plus simple, de même coût, dont les instructions de coût unité donnent comme résultat les polynomes p_2^j .

On peut écrire :

$$p_2^j \equiv (A_1^j X + B_1^j Y) \cdot (A_2^j X + B_2^j Y) \quad A_1^j, A_2^j \in K^m, B_1^j, B_2^j \in K^n$$

Ceci peut se mettre sous la forme suivante :

$$(5) \quad p_2^j \equiv X^t A_1^j A_2^j X + Y^t B_1^j B_2^j Y + X^t A_1^j B_2^j Y + Y^t B_1^j A_2^j X.$$

En remplaçant dans (4), les polynomes p_2^j par cette expression, on obtient

$$(6) \quad X^t B_i Y \equiv \sum_{j=1}^q \alpha_j^i (X^t A_1^j A_2^j X + Y^t B_1^j B_2^j Y + X^t A_1^j B_2^j Y + Y^t B_1^j A_2^j X) \quad i=1, \dots, p.$$

Pour que (6) soit vérifiée, il est nécessaire que l'on ait :

$$X^t \left(\sum_{j=1}^q \alpha_j^i A_1^j A_2^j \right) X \equiv 0, \quad X^t \left(\sum_{j=1}^q \alpha_j^i B_1^j B_2^j \right) Y \equiv 0,$$

$$X^t B_i Y \equiv X^t \left(\sum_{j=1}^q \alpha_j^i (A_1^j B_2^j) \right) Y \quad (i=1, \dots, p).$$

De la dernière identité, on en déduit de suite la relation (1) du théorème à savoir :

$$B_i = \sum_{j=1}^q \alpha_j^i (A_1^j B_2^j + A_2^j B_1^j) \quad (i=1, \dots, p).$$

Les conditions 2 et 3 du théorème sont les conditions qui expriment que le

matrices $\sum_{j=1}^q \alpha_j^i A_1^j A_2^j$ et $\sum_{j=1}^q \alpha_j^i B_1^j B_2^j$ sont anti-symétriques, ce qui

est bien la condition pour que les formes quadratiques construites avec

ces matrices soient en faite équivalentes au polynome nul.

Le coût minimal est donc bien égal au plus petit entier q tel que les conditions (1), (2) et (3) soient satisfaites.

2 . COUT MINIMAL DANS LE CAS NON COMMUTATIF

Théorème 2

/ Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est non-commutatif, le nombre minimal de multiplications "générales" que doit employer tout algorithme de \mathcal{A}_p pour calculer $X^t B_i Y$ ($i=1, \dots, p$) est égal au plus petit entier q tel que l'on puisse écrire :

$$B_i = \sum_{j=1}^q \alpha_j^i A_1^j B_2^{j^t} \quad , \quad \alpha_j^i \in K \quad A_1^j \in K^m \quad B_2^j \in K^n$$

($j=1, \dots, q$) . /

□ On peut en effet dans ce cas reprendre exactement le même raisonnement que celui utilisé dans le théorème 1 et ceci jusqu'à l'obtention de l'identité (5), à savoir :

$$X^t B_i Y \equiv \sum_{j=1}^q \alpha_j^i (X^t A_1^j A_2^{j^t} X + Y^t B_1^j B_2^{j^t} Y + X^t A_1^j B_2^{j^t} Y + Y^t B_1^j A_2^{j^t} X)$$

($i=1, \dots, p$) .

En effet, jusqu'à ce stade du raisonnement, la commutativité de $K[x_1, \dots, x_m, y_1, \dots, y_n]$ n'avait pas été encore invoquée. Par contre, les conditions à vérifier pour que ces identités aient lieu dépendent de la commutativité ou de la non-commutativité de $K[x_1, \dots, x_m, y_1, \dots, y_n]$.

Dans le cas non-commutatif, pour que l'on ait :

$$X^t A X \equiv 0 \quad (A \in \mathcal{M}_{m,m}(K))$$

il est nécessaire que la matrice A soit égale à la matrice nulle (et non plus seulement que A soit anti-symétrique) . Par conséquent, les identités écrites en (5) seront satisfaites dans le cas non-commutatif si et seulement si on a les égalités suivantes :

$$(1') \quad \sum_{j=1}^q \alpha_j^i A_1^j A_2^{j^t} = 0 \quad ,$$

$$(2') \quad \sum_{j=1}^q \alpha_j^i B_1^j B_2^{j^t} = 0 \quad ,$$

$$(3') \quad \sum_{j=1}^q \alpha_j^i B_1^j A_2^{j^t} = 0 \quad ,$$

$$(4') \quad B_i = \sum_{j=1}^q \alpha_j^i A_1^j B_2^{j^t} \quad , \quad i=1, \dots, p .$$

Les égalités (1'), (2') et (3') peuvent être toutes satisfaites si l'on prend $B_1^j = 0$, $j=1, \dots, q$ et $A_2^j = 0$, $j=1, \dots, q$.

Par conséquent le plus petit entier q tel que l'on ait (4') est bien égal au nombre minimal de multiplications générales nécessaires pour calculer $X^t B_i Y$ ($i=1, \dots, p$) dans le cas où $K[X, Y]$ est supposé être non-commutatif. \square

3 . INTERPRETATIONS AVEC LA NOTION DE RANG TENSORIEL.

Définition 1 Rang tensoriel d'un ensemble de matrices.

/ Le rang tensoriel des p matrices B_1, \dots, B_p de $\mathcal{M}_{m,n}(K)$ est, par définition, le plus petit entier q tel que l'on puisse écrire :

$$B_i = \sum_{j=1}^q \alpha_j^i A_1^j A_2^{j^t} \quad A_1^j \in K^m, A_2^j \in K^n, \quad i=1, \dots, p, \\ j=1, \dots, q. \quad /$$

Une matrice de rang un (appelée encore matrice antiscalaire) s'écrit toujours sous la forme $A B^t$ avec A et B deux vecteurs de dimensions adéquates. On peut donc dire que le rang tensoriel des p matrices B_i est égal au plus petit nombre de matrices de rang un qui permettent d'exprimer toutes les matrices B_i . On peut interpréter les deux théorèmes précédents uniquement en fonction de cette notion de rang tensoriel, cf. Lafon (12) .

Théorème 2'

/ Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est non-commutatif le nombre minimal de multiplications "générales" nécessaires pour évaluer $X^t B_i Y$ ($i=1, \dots, p$) par un algorithme de \mathcal{A}_p est égal au rang tensoriel des p matrices B_i .

\square Cela provient directement de la définition du rang tensoriel des p matrices B_i . \square

Théorème 1'

/ Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est commutatif le nombre minimal de multiplications "générales" nécessaires pour évaluer $X^t B_i Y$ ($i=1, \dots, p$) pour un algorithme de \mathcal{A}_p est égal au minimum du rang tensoriel des matrices N_1, \dots, N_p de $\mathcal{M}_{m+n, m+n}(K)$ telles que :

$$N_i + N_i^t = \begin{pmatrix} 0 & B_i \\ B_i^t & 0 \\ & \vdots \end{pmatrix} \quad (i=1, \dots, p) \quad . \quad /$$

□ Soit q le rang tensoriel des matrices N_1, \dots, N_p .

On peut donc écrire :

$$(6) \quad N_i = \sum_{j=1}^q \lambda_j^i U_j V_j^t \quad \lambda_j^i \in K \quad U_j, V_j \in K^{m+n} \quad \begin{matrix} i=1, \dots, p \\ j=1, \dots, q \end{matrix}$$

On pose :

$$U_j^t = (A_1^j \ B_1^j)^t \quad A_1^j \in K^m \quad B_1^j \in K^n \quad j=1, \dots, q$$

$$V_j^t = (A_2^j \ B_2^j)^t \quad A_2^j \in K^m \quad B_2^j \in K^n \quad j=1, \dots, q$$

$$N_i = \begin{pmatrix} N_{11}^i & N_{12}^i \\ N_{21}^i & N_{22}^i \end{pmatrix} \quad \begin{matrix} N_{11}^i \in \mathcal{M}_{m,m}(K) & N_{12}^i \in \mathcal{M}_{m,n}(K) \\ N_{21}^i \in \mathcal{M}_{n,m}(K) & N_{22}^i \in \mathcal{M}_{n,n}(K) \end{matrix}$$

La relation (6) peut se décomposer en les relations suivantes

$$N_{11}^i = \sum_{j=1}^q \lambda_j^i A_1^j A_2^{j^t}, \quad i=1, \dots, q$$

$$N_{12}^i = \sum_{j=1}^q \lambda_j^i A_1^j B_2^{j^t},$$

$$N_{21}^i = \sum_{j=1}^q \lambda_j^i B_1^j A_2^{j^t},$$

$$N_{22}^i = \sum_{j=1}^q \lambda_j^i B_1^j B_2^{j^t}.$$

Les conditions $N_i + N_i^t = \begin{pmatrix} 0 & B_i \\ B_i^t & 0 \end{pmatrix}$ se traduisent donc sur les sous

matrices par les conditions :

$$N_{11}^i + N_{11}^{i^t} = 0, \quad N_{22}^i + N_{22}^{i^t} = 0, \quad N_{12}^i + N_{21}^{i^t} = B_i^t.$$

On a donc en fait exactement les conditions du théorème 1. □

4. ALGORITHMES DE COÛT MINIMAL

a/ Algorithmes optimaux dans le cas $K[X, Y]$ commutatif.

Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est commutatif les théorèmes 1 et 1' donnent une caractérisation du coût minimal des algorithmes de \mathcal{A}_P . Dans la démonstration de ces théorèmes, on a considéré des algorithmes tout à fait généraux, en particulier on a admis que des multiplications "générales" pouvaient donner comme résultats des polynômes de degré supérieur à deux. Le résultat du théorème 1 montre que le nombre minimal de multiplications "générales" utilisées par les algorithmes de \mathcal{A}_P pour résoudre le problème P (B_1, \dots, B_p) est aussi égal au nombre minimal de multiplications "générales" utilisées, pour résoudre ce même problème, par les algorithmes de \mathcal{A}_P dont les instructions de coût 1 sont seulement de la forme :

$$\langle k \rangle \leftarrow \langle i \rangle \cdot \langle j \rangle ,$$

avec $\langle i \rangle \equiv A_1^{k,t} X + B_1^{k,t} Y ,$

$$\langle j \rangle \equiv A_2^{k,t} X + B_1^{k,t} , B_2^{k,t} \in K^m .$$

De la connaissance d'une décomposition des matrices B_i ($i=1, \dots, p$) satisfaisant aux conditions (1), (2) et (3) du théorème 1, on peut déduire aussitôt un algorithme optimal de ce type particulier.

b/ Algorithmes optimaux dans le cas $K[X, Y]$ non commutatif

Si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ n'est pas commutatif, les théorèmes 2 et 2' donnent une caractérisation du coût minimal des algorithmes de \mathcal{A}_P . En fait, le résultat obtenu montre qu'il existe au moins un algorithme optimal tel que les seules opérations de coût unité soient du type :

$$\langle k \rangle \leftarrow \langle i \rangle \cdot \langle j \rangle ,$$

avec $\langle i \rangle \equiv A_1^{k,t} X$

$$\langle j \rangle \equiv B_2^{k,t} Y , \quad A_1^{k,t} \in K^m , \quad B_2^{k,t} \in K^n .$$

En particulier, la connaissance de la décomposition des p matrices B_i en q matrices de rang un permet de construire aussitôt un algorithme optimal de ce type particulier (q égal le rang tensoriel des p matrices B_i).

$$\text{Si } B_i = \sum_{j=1}^q \alpha_j^i U_j V_j^t \quad U_i \in K^m, V_j \in K^n, i=1, \dots, p.$$

Les q multiplications "générales" de l'algorithme optimal en question seront :

$$P_i \equiv (U_j^t X) \cdot (V_j^t Y) \quad i=1, \dots, q,$$

et on aura :

$$X^t B_i Y \equiv \sum_{j=1}^q \alpha_j^i (U_j^t X) \cdot (V_j^t Y) \quad i=1, \dots, p.$$

c/ Gain apporté par l'emploi de la commutativité

L'utilisation de la commutativité permet elle de diminuer le nombre de multiplications "générales" nécessaires pour calculer p éléments $X^t B_i Y$ de $K[X, Y]$?

La réponse à cette question est évidemment affirmative. Il suffit pour s'en convaincre de considérer le calcul du polynôme $p_1(x, y)$ suivant :

$$p_1(x, y) = x^2 - y^2.$$

Si $K[x, y]$ n'est pas commutatif, le calcul de $p_1(x, y)$ à partir de $K \cup \{x, y\}$ nécessite deux multiplications générales, à savoir les deux produits $x \cdot x$ et $y \cdot y$. Par contre si $K[x, y]$ est commutatif, alors ce même calcul peut être fait en une seule multiplication, car on peut écrire dans ce cas :

$$p_1(x, y) = (x+y)(x-y).$$

Les théorèmes 1 et 2 permettent d'énoncer le théorème suivant :

Théorème 3

/ Soit q , (respectivement q'), le nombre minimal des multiplications "générales" nécessaires pour calculer les p éléments $X^t B_i Y$ ($i=1, \dots, p$) de $K[X, Y]$ quand $K[X, Y]$ est commutatif (respectivement n'est pas commutatif). Alors on a :

$$\lceil \frac{q'}{2} \rceil \leq q \leq q' . /$$

□ D'après le théorème 1, q est tel que l'on peut écrire :

$$B_i = \sum_{j=1}^q \alpha_j^i (A_1^j B_2^j + A_2^j B_1^j) \quad (i=1, \dots, p) .$$

Ceci montre que les p matrices s'expriment à l'aide de $2q$ matrices de rang un. Comme le nombre minimal de matrices de rang un avec lesquelles elles peuvent s'exprimer est égal à q' (théorème 1 et 1') on a forcément :

$$2q \geq q' ,$$

soit $q \geq \lceil \frac{q'}{2} \rceil .$

($\lceil a \rceil$ désigne l'entier le plus petit supérieur à a).

La majoration est évidente puisque tout algorithme qui calcule $X^t B_i Y$ ($i=1, \dots, p$), quand $K[X, Y]$ n'est pas commutatif est aussi valable quand $K[X, Y]$ est supposé commutatif. □

Le fait que l'algorithme optimal de calcul de plusieurs formes bilinéaires n'utilise que des multiplications entre des formes linéaires des indéterminées (théorème 1 et 2) a été montré pour la première fois par HOPCROFT et KERR (11) et par WINOGRAD (15). L'interprétation avec le rang tensoriel est essentiellement due à GASTINEL (10), et à STRASSEN (14), mais a été étudié aussi par BROCKETT-DOBKIN (8), FIDUCCIA (9) et LAFON (12).

REFERENCES SUR LE CHAPITRE BII

- (8) BROCKETT, R.W., DOBKIN, D.,
"On the optimal evaluation of a set of bilinear forms".
5th Annual ACM symposium on theory of Computing
Austin Texas, April 30 May 2, (1973), 88-95`.

- (9) FIDUCCIA, C.M., "On obtaining upper bounds on the complexity
of matrix multiplication".
Proc. of IBM symp on Complexity of Computer Computation (march
(march 1972), 31-40.

- (10) GASTINEL, N., "Le rang tensoriel d'un ensemble de matrices.
Applications". Séminaire d'Analyse Numérique, Grenoble, N° 159,
(octobre 1972).

- (11) HOPCROFT, J., KERR, L., "On minimizing the number of multipli-
cations necessary for matrix multiplication".
SIAM J. Appli. Math., vol 20, (1971), 30-36.

- (12) LAFON, J.C., "Optimum computation of p bilinear forms".
J. Linear Algebra, 10, (1975), 225-240.

- (13) STRASSEN, V., "Evaluation of rational functions".
proc. IBM symp. on the complexity of Computer Computations.
(march 1972).

- (14) STRASSEN, V., "Vermeidung von Divisionen".
Crelle J. für die Reine und Angew. Mathematik (1973).

- (15) WINOGRAD, S. " On multiplication of 2×2 matrices".
J. Linear Algebra, 4, (1971), 381-388.

CHAPITRE BIII

ETUDE DU RANG TENSORIEL DE p MATRICES

PLAN

1. Principales propriétés.
2. Rang d'un tenseur
3. Condition pour que p matrices régulières aient un rang tensoriel minimal.
4. Rang tensoriel d'un espace de matrices possédant au moins une matrice régulière.
5. Exemple du produit de deux matrices.
6. Calcul approché du rang tensoriel.

Les résultats précédents montrent l'importance du calcul du rang tensoriel d'un ensemble de p matrices, ainsi que celle de la détermination d'une décomposition en matrices de rang un de ces matrices. Cette notion de rang tensoriel généralise en fait la notion de rang d'une matrice. On peut l'interpréter comme le rang d'un tenseur d'ordre trois. L'intérêt mathématique de cette notion justifie aussi l'étude qui en est faite dans ce chapitre.

1 . PRINCIPALES PROPRIETES.

Définition 1

/ Le rang tensoriel des p matrices B_1, \dots, B_p de $\mathcal{M}_{m,n}(K)$ est égal au plus petit entier q tel que l'on puisse écrire :

$$B_i = \sum_{j=1}^q \alpha_j^i U_j V_j^t \quad i=1, \dots, p$$

$$U_j \in K^m, V_j \in K^n, j=1, \dots, q, \alpha_j^i \in K. /$$

Remarque 1

On pourrait développer toute l'étude qui va suivre en supposant seulement que K est un anneau à élément unité. Il faudrait alors parler du module des matrices à m lignes et n colonnes défini sur K . Pour éviter certaines lourdeurs d'expositions, on gardera l'hypothèse que K est un corps.

Les propriétés suivantes du rang tensoriel des p matrices B_i ($i=1, \dots, p$) sont assez évidentes, pour être données sans démonstration.

Propriété 1

Le rang tensoriel d'une seule matrice est égal à son rang.

Propriété 2

Le rang tensoriel des p matrices B_i reste le même si on remplace une matrice par une combinaison linéaire des autres.

Il suffit donc pour étudier le rang tensoriel des p matrices B_i ($i=1, \dots, p$) de ne considérer que celles qui sont linéairement indépendantes (c'est-à-dire celles qui forment une base de l'espace $\{B_i\}$ engendré par ces matrices). Toutes les bases de l'espace $\{B_i\}$ ont le même rang tensoriel : on peut donc parler du rang tensoriel de l'espace $\{B_i\}$ que l'on notera $Rt\{B_i\}$. Dans la suite $\dim\{B_i\}$ désignera la dimension du sous-espace $\{B_i\}$ de $\mathcal{M}_{m,n}(K)$.

Propriété 3

Si q est le rang tensoriel de l'espace $\{B_i\}$ alors toutes les matrices de $\{B_i\}$ s'expriment comme une combinaison linéaire de q matrices antiscales indépendantes.

- Le fait que toutes les matrices de $\{B_i\}$ puissent s'exprimer comme combinaison linéaire de q matrices antiscales découle directement de la proposition précédente. Si ces matrices n'étaient pas indépendantes alors le rang tensoriel de l'espace $\{B_i\}$ serait plus petit que q ce qui est impossible.

Propriété 4

On a l'encadrement suivant du rang tensoriel :

$$(2) \quad \max_{B \in \{B_i\}} (\dim\{B_i\}, \max \text{Rang}(B)) \leq Rt\{B_i\} \leq mn . \quad /$$

- Il faut au moins $\dim\{B_i\}$ matrices pour générer l'espace $\{B_i\}$, d'autre part une matrice B particulière ne peut être une combinaison linéaire de moins de $\text{Rang}(B)$ matrices de rang un. La minoration de (2) est donc évidente. La majoration provient du fait que l'on peut écrire toute matrice B de $\mathcal{M}_{m,n}(K)$ sous la forme suivante :

$$B = \sum_{i,j} B_{ij} E_{ij} \quad \text{avec} \quad E_{ij} = e_i e_j^t$$

Les vecteurs $\{e_i\}$ et $\{e_j^t\}$ formant les bases canoniques de K^m et K^n respectivement. □

Soit $B(Z)$ la matrice suivante :

$$B(Z) = \sum_{i=1}^p z_i B_i$$

On pose $Z^t = (z_1, \dots, z_p)$. Si les z_i ($i=1, \dots, p$) sont considérées comme des indéterminées, $B(Z)$ représente une matrice quelconque de l'espace $\{B_i\}$. On suppose que les indéterminées z_1, \dots, z_p commutent avec les indéterminées x_1, \dots, x_m et y_1, \dots, y_n .

Définition 2

Le rang tensoriel d'une matrice $B(Z)$ à m lignes et n colonnes, dépendant de p paramètres z_1, \dots, z_p est le plus petit entier q tel que l'on puisse écrire :

$$(3) \quad B(Z) = \sum_{j=1}^q (\lambda_j^t Z) U_j V_j^t \quad U_j \in K^m, V_j \in K^n, \lambda_j \in K^p, j=1, \dots, q.$$

On notera par $Rt B(Z)$ ce rang tensoriel. Cette notion n'est pas nouvelle comme le montre la proposition suivante.

Proposition 1

/ On a :

$$Rt B(Z) = Rt \{B_i\} . /$$

□ Si on pose $\lambda_j^t = (\lambda_1^j, \lambda_2^j, \dots, \lambda_q^j)$ $j=1, \dots, p$

alors la relation (3) entraîne :

$$B_i = \sum_{j=1}^q \lambda_j^i U_j V_j^t \quad i=1, \dots, p .$$

Réciproquement, si B_i ($i=1, \dots, p$) s'écrivent comme ci-dessus $B(Z)$ s'écrira bien comme en (3). □

En général, on étudiera le rang tensoriel de l'espace $\{B_i\}$ directement sur la matrice $B(Z)$ qui paramétrise cet espace.

Propriété 5

/ Pour toute matrice P de $\mathcal{M}_{m,m}(K)$ régulière et toute matrice Q régulière de $\mathcal{M}_{n,n}(K)$, on a :

$$\text{Rt} (PB(Z)Q) = \text{Rt} B(Z) . /$$

□ En effet, si :

$$B(Z) = \sum_{j=1}^q (\lambda_j^t Z) U_j V_j^t$$

on a aussi

$$PB(Z)Q = \sum_{j=1}^q (\lambda_j^t Z) (PU_j) (Q^t V_j)^t$$

et réciproquement. □

Cette propriété signifie en fait que le rang tensoriel des matrices B_1, \dots, B_p est attaché en réalité aux formes linéaires f_1, \dots, f_p qu'elles définissent par :

$$f_i : K^n \rightarrow K^m \quad \forall Y \in K^n \rightarrow B_i Y \quad i=1, \dots, p$$

(Les matrices P et Q sont les matrices de changement de base dans K^m et K^n respectivement).

On peut donc parler du rang tensoriel de ces p formes linéaires, qui est aussi le rang tensoriel de l'espace $\{f_i\}$ des formes linéaires qu'elles engendrent. La propriété 5 traduit ce fait sur la matrice B(Z).

Propriété 5'

/ Pour toutes ^{les} matrices P, Q, R régulières ($P \in \mathcal{M}_{m,m}(K)$, $Q \in \mathcal{M}_{n,n}(K)$, $R \in \mathcal{M}_{p,p}(K)$), on a :

$$\text{Rt}(PB(RZ)Q) = \text{Rt} B(Z) . /$$

□ En effet si $B(Z) = \sum_{j=1}^q (\lambda_j^t Z) U_j V_j^t$

on a également :

$$P B(RZ) Q = \sum_{j=1}^q (R^t \lambda_j^t) (P U_j) (Q^t V_j)^t$$

et réciproquement puisque les matrices sont supposées inversibles (sinon il faudrait remplacer le signe = par \leq). □

Remarque

Cette dernière propriété montre en particulier que le rang tensoriel de $B(Z)$ reste inchangé si l'on permute les lignes, les colonnes et les paramètres Z_1, \dots, Z_p , ou si on ajoute à une ligne (colonne) de $B(Z)$ une combinaison linéaire des autres lignes (colonnes).

Propriété 6

/ Si $\{B_i\} \subset \{B'_i\}$, on a la relation :

$$Rt \{B_i\} \leq Rt \{B'_i\} \quad . \quad /$$

Propriété 7

Si une matrice de rang k s'exprime comme une combinaison linéaire de q ($q \geq k$) matrices antiscales $U_j V_j^t$ ($j=1, \dots, q$) alors il doit exister k vecteurs linéairement indépendants parmi les q vecteurs U_j ($j=1, \dots, q$) ou V_j ($j=1, \dots, q$).

□ Soit donc B une matrice régulière de $\mathcal{M}_{m,n}(K)$.

Supposons B de rang k ($k < \text{Min}(m,n)$) et écrite sous la forme :

$$(4) \quad B = \sum_{j=1}^q \alpha_j U_j V_j^t \quad U_j \in K^m, V_j \in K^n, j=1, \dots, q.$$

On peut toujours supposer que U_1, \dots, U_ℓ sont les ℓ vecteurs linéairement indépendants parmi les q vecteurs U_1, \dots, U_q .

On a alors :

$$U_j = \sum_{i=1}^{\ell} a_i^j U_i \quad (j=\ell+1, \dots, q).$$

L'expression (4) devient alors :

$$B = \sum_{j=1}^{\ell} \alpha_j U_j V_j^t + \sum_{j=\ell+1}^q \alpha_j \left(\sum_{i=1}^{\ell} a_{ij} U_i \right)$$

Soit :

$$B = \sum_{j=1}^{\ell} U_j \left[\alpha_j V_j^i + \sum_{i=\ell+1}^q \alpha_i a_{ij} V_i \right]^t$$

La matrice B s'exprime donc avec ℓ matrices anti-scalaires.

D'après la propriété 4 on doit donc avoir $\ell \geq q$.

(Même raisonnement pour les vecteurs V_1, \dots, V_q). \square

Propriété 8

/ Soit $B_i^!$ ($i=1, \dots, p$) les p matrices obtenues à partir des matrices B_i ($i=1, \dots, p$) en gardant les éléments des lignes i_1, \dots, i_{m_1} et j_1, \dots, j_{n_1} ($B_i^! \in \mathcal{M}_{m_1, n_1}(K')$), alors on a :

$$Rt \{B_i^!\} \leq Rt \{B_i\} . /$$

Propriété 9

On a les inégalités suivantes :

$$1/ Rt(B(Z)+C(Z')) \leq Rt(B(Z)) + Rt(C(Z'))$$

$$2/ \text{Max}(Rt B(Z), Rt C(Z)) \leq Rt(B(Z) \oplus C(Z')) \leq Rt(B(Z)) + Rt(C(Z'))$$

$$3/ \text{Max}(Rt B(Z), Rt C(Z)) \leq Rt(B(Z) \otimes C(Z')) \leq Rt(B(Z)) \times Rt(C(Z'))$$

(Pour l'inégalité 1/, on suppose évidemment $B(Z)$ et $C(Z)$ de même dimension).

\square Les minorations données sont évidentes. Pour les majorations, il suffit de constater que si $B(Z)$ s'exprime à l'aide des q_1 matrices de rang un D_j ($j=1, \dots, q_1$) et $C(Z)$ à l'aide des q_2 matrices de rang un E_j ($j=1, \dots, q_2$) alors : $B(Z) + C(Z')$ s'exprime à l'aide des $q_1 + q_2$ matrices D_j ($j=1, \dots, q_1$) et E_j ($j=1, \dots, q_2$).
 $B(Z) \oplus C(Z')$ s'exprime à l'aide des $q_1 + q_2$ matrices de rang un $A_j \oplus 0$ et $E_j \oplus 0$ (au 0 désigne une matrice nulle de dimension appropriée). Enfin,

$B(Z) \otimes C(Z')$ s'exprime à l'aide des $q_1 q_2$ matrices de rang un $D_i \otimes E_j$ ($i=1, \dots, q_1, j=1, \dots, q_2$) .

2 . RANG D'UN TENSEUR

Suivant la définition de STRASSEN (4') (cf. aussi GASTINEL (10'), LAFON (12'), BROCKETT-DOBKIN (8')), on définit le rang d'un tenseur d'ordre trois $(b_{i,j,k})$ de $K^m \otimes K^n \otimes K^p$ comme étant le plus petit entier q tel que l'on puisse écrire :

$$(b_{i,j,k}) = \sum_{j=1}^q U_j \otimes V_j \otimes W_j$$

avec $U_j \in K^m, V_j \in K^n, W_j \in K^p$.

On peut relier cette notion à celle du rang tensoriel de p matrices.

Théorème 1

/ Le rang tensoriel des p matrices B_k ($B_k(i,j) = B_{i,j,k}$) est égal au rang du tenseur $B_{i,j,k}$ de $K^m \otimes K^n \otimes K^p$. /

□ Soit $F : K^m \times K^n \times K^p \rightarrow K$ la forme trilinéaire définie par :

$$F(X,Y,Z) = X^t B(Z) Y \quad \forall (X,Y,Z) \in K^m \times K^n \times K^p$$

On peut écrire :

$$F(X,Y,Z) = \sum_{i,j,k}^{m,n,p} B_{i,j,k} x_i y_j z_k$$

Soit q le rang tensoriel des p matrices B_i .

La matrice $B(Z)$ peut donc s'écrire sous la forme :

$$B(Z) = \sum_{j=1}^q \lambda_j^t Z U_j V_j^t .$$

On a donc :

$$X^t B(Z) Y = \sum_{j=1}^q (U_j^t X) (V_j^t Y) (\lambda_j^t Z) .$$

Par conséquent q est bien égal au rang du tenseur $B_{i,j,k}$ associé à la forme trilinéaire F (la réciproque est évidente). \square

Corollaire 1

/ Le rang tensoriel des p matrices B_k ($B_k(i,j) = B_{i,j,k}$), des m matrices B'_i ($B'_i(j,k) = B_{i,j,k}$) et celui des n matrices B''_j ($B''_j(i,k) = B_{i,j,k}$) est le même. /

\square En effet, ces trois familles de matrices (qui appartiennent respectivement aux espaces $\mathcal{M}_{m,n}(K)$, $\mathcal{M}_{n,p}(K)$ et $\mathcal{M}_{m,p}(K)$) définissent la même forme trilinéaire et par conséquent, d'après le théorème 1, leur rang tensoriel est le même. \square

Corollaire 2

/ Toutes les propriétés précédentes peuvent s'appliquer indifféremment aux matrices $B(Z)$, $B'(X)$ et $B''(Y)$, en particulier la propriété 3 donne :

$$\text{Max}[\text{Min}_i \text{Rang}(B_i), \text{Max}_i \text{Rang}(B'_i), \text{Max}_i \text{Rang}(B''_i)] \leq \text{Rt } F \leq \text{Min}(mn, mp, pn) . /$$

Propriété 10

/ Si k lignes (respectivement k colonnes) de la matrice $B(Z)$ sont K linéairement indépendantes alors on a :

$$k \leq \text{Rt } (B(Z)) . /$$

\square Les éléments de la matrice $B(Z)$ sont des formes linéaires en les indéterminées Z_1, \dots, Z_p . A toute ligne (resp. colonne) de $B(Z)$ correspond un vecteur de $K^n[z_1, \dots, z_p]$ (resp. de $K^m[z_1, \dots, z_p]$). k éléments ℓ_1, \dots, ℓ_k , de $K^n[z_1, \dots, z_p]$ seront dits K linéairement indépendants si :

$$\sum_{i=1}^k \lambda_i \ell_i = 0 \Rightarrow \lambda_i = 0 \quad i=1, \dots, k \quad , \quad \lambda_i \in K \quad i=1, \dots, k .$$

On peut toujours supposer que les k lignes indépendantes (au sens précédent) de la matrice $B(Z)$ sont les k premières. Soit q le rang tensoriel de $B(Z)$. On peut donc écrire :

$$B(Z) = \sum_{i=1}^q (\lambda_i^t Z) U_i V_i^t .$$

Supposons $q < k$. Alors on peut déterminer un vecteur W de K^m de la forme :

$$W^t = (W_1, \dots, W_k, 0, 0, \dots, 0)$$

tel que l'on ait :

$$W^t U_i = 0 \quad i=1, \dots, q \quad \prod_{i=1}^k W_i \neq 0 .$$

En effet, ce système à résoudre est un système de q équations à k ($k > q$) inconnues.

Avec ce choix de W on a :

$$W^t B(Z) = 0 \quad ,$$

ce qui entraîne la K dépendance des k premières lignes de la matrice $B(Z)$, d'où la contradiction. \square

3 . CONDITION POUR QUE p MATRICES REGULIERES AIENT UN RANG TENSORIEL MINIMAL

Théorème 2

/ Si les p matrices B_i ($i=1, \dots, p$) de $\mathcal{M}_{m,m}(K)$ sont toutes régulières, le rang tensoriel de $\{B_i\}$ est compris entre m et m^2 , il est égal à m si et seulement si les $p-1$ matrices $B_j B_1^{-1}$ ($j=2, \dots, p$) sont simultanément diagonalisables. (cf. Lafon (12')) .

\square Le rang tensoriel d'un ensemble de matrices est minoré par le rang de l'une quelconque des matrices de l'ensemble. Dans le cas présent, toutes les matrices B_i ($i=1, \dots, p$) étant de rang m on a bien :

$$m \leq \text{Rt} \{B_i\} \leq m^2 \quad .$$

On va chercher maintenant une condition nécessaire et suffisante pour que l'on ait $\text{Rt} \{B_i\} = m$.

- Condition nécessaire :

On suppose $\text{Rt } \{B_i\} = m$. On peut donc écrire :

$$B_j = \sum_{i=1}^m \lambda_i^j U_i V_i^t \quad U_i, V_i \in K^m \quad i=1, \dots, m$$

$$\lambda_i^j \in K \quad \lambda_i^j \neq 0 \quad \forall i=1, \dots, m.$$

Soit x un élément quelconque de K^m . On aura :

$$(1) \quad B_j x = \sum_{i=1}^m \lambda_i^j (V_i^t x) U_i, \quad j=1, \dots, p.$$

Soit U la matrice de $\mathcal{M}_{m,m}(K)$ dont les colonnes sont égales aux vecteurs U_1, \dots, U_m . Cette matrice U possède une inverse U^{-1} , car la régularité des matrices B_j entraîne l'indépendance des vecteurs U_i . (Il suffirait seulement que l'une des matrices B_j soit régulière pour qu'il en soit ainsi).

En résolvant (1) par rapport aux $\lambda_i^j V_i^t x$ ($i=1, \dots, m$) on obtient :

$$(2) \quad \begin{pmatrix} \lambda_1^j & V_1^t & x \\ \lambda_2^j & V_2^t & x \\ \vdots & \vdots & \vdots \\ \lambda_m^j & V_m^t & x \end{pmatrix} = U^{-1} B_j x \quad j=1, \dots, p \quad \forall x \in K^m$$

Comme tous les λ_i^j ($i=1, \dots, m$, $j=1, \dots, p$) sont distincts de zéro et donc inversibles, on peut écrire (2) sous la forme suivante :

$$\begin{pmatrix} V_1^t \\ V_2^t \\ \vdots \\ V_m^t \end{pmatrix} x = \begin{pmatrix} (\lambda_1^j)^{-1} & & & \\ & (\lambda_2^j)^{-1} & & 0 \\ & & \ddots & \\ 0 & & & (\lambda_m^j)^{-1} \end{pmatrix} U^{-1} B_j x \quad j=1, \dots, p.$$

Ceci devant être vrai quel que soit x de K^m on en déduit l'égalité des matrices :

$$(3) \begin{pmatrix} V_1^t \\ V_2^t \\ \vdots \\ V_m^t \end{pmatrix} = \begin{pmatrix} (\lambda_1^j)^{-1} & & & \\ & (\lambda_2^j)^{-1} & & \\ & & \ddots & \\ & & & 0 \\ 0 & & & (\lambda_m^j)^{-1} \end{pmatrix} U^{-1} B_j \quad j=1, \dots, p$$

On désigne par V^t la matrice dont les lignes sont égales à $V_1^t, V_2^t, \dots, V_m^t$.
De (3), on déduit que l'on a toujours pour $j=2, \dots, p$:

$$\begin{pmatrix} (\lambda_1^1)^{-1} & & & \\ & (\lambda_2^1)^{-1} & & \\ & & \ddots & \\ & & & 0 \\ 0 & & & (\lambda_n^1)^{-1} \end{pmatrix} U^{-1} B_1 = \begin{pmatrix} (\lambda_1^j)^{-1} & & & \\ & (\lambda_2^j)^{-1} & & \\ & & \ddots & \\ & & & (\lambda_n^j)^{-1} \end{pmatrix} U^{-1} B_j$$

Ceci s'écrit aussi sous la forme suivante :

$$U^{-1} B_j B_1^{-1} U = \begin{pmatrix} \lambda_1^j \cdot (\lambda_1^1)^{-1} & & & \\ & \lambda_2^j \cdot (\lambda_2^1)^{-1} & & \\ & & \ddots & \\ & & & 0 \\ 0 & & & \lambda_n^j \cdot (\lambda_n^1)^{-1} \end{pmatrix} \quad j=2, \dots, p .$$

Ceci signifie que la matrice U est la matrice des vecteurs propres des $p-1$ matrices $B_j B_1^{-1}$, $j=2, \dots, p$, et que par conséquent, ces matrices sont simultanément diagonalisables. Cette dernière condition est donc bien nécessaire pour que les matrices B_1, \dots, B_p aient un rang tensoriel égal à m .

- Condition suffisante :

Il faut montrer que la condition nécessaire est aussi bien suffisante.
Soit donc U la matrice des vecteurs propres des $p-1$ matrices $B_j B_1^{-1}$, $j=2, \dots, p$. On a donc :

$$(4) \quad U^{-1} B_j B_1^{-1} U = \begin{pmatrix} \lambda_1^j & & & \\ & \lambda_2^j & & \\ & & \ddots & \\ 0 & & & \lambda_m^j \end{pmatrix} \quad j=2, \dots, p .$$

On note par U_1, \dots, U_m les m vecteurs propres (vecteurs colonnes de U) des matrices $B_j B_1^{-1}$ ($j \neq 1$). Il s'agit de démontrer que le rang tensoriel des p matrices B_i ($i=1, \dots, p$) est exactement égal à m . Soit V^t la matrice $U^{-1} B_1$. On note par V_1^t, \dots, V_m^t les m lignes de cette matrice. On a évidemment :

$$B_1 = U V^t \quad \text{donc} \quad B_1 = \sum_{i=1}^m U_i V_i^t$$

D'autre part, les relations (4) s'écrivent maintenant sous la forme suivante :

$$U^{-1} B_j = \begin{pmatrix} \lambda_1^j & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \lambda_m^j \end{pmatrix} V^t \quad j=2, \dots, p .$$

C'est-à-dire que l'on a aussi :

$$B_j = \sum_{i=1}^m \lambda_i^j U_i V_i^t .$$

Ce qui montre bien que les p matrices B_1, \dots, B_p ont un rang tensoriel égal à m . \square

On va maintenant étudier le cas où une seule des p matrices B_i ($i=1, \dots, p$) est supposée régulière. Dans ce cas, on a le théorème suivant :

4. RANG TENSORIEL D'UN ESPACE CONTENANT AU MOINS UNE MATRICE REGULIERE

Théorème 3

/ Si l'une au moins des p matrices B_i ($i=1, \dots, p$) de $\mathcal{M}_{m,m}(K)$ est régulière, alors le rang tensoriel de ces p matrices est compris entre m et m^2 . Il est égal à m si et seulement si il existe deux matrices inversibles P et Q de $\mathcal{M}_{m,m}(K)$ telles que toutes les matrices $P B_i Q$ ($i=1, \dots, p$) soient diagonales. /

□ On peut toujours supposer B_1 régulière. Si q est le rang tensoriel de $\{B_i\}$ on a $q \geq \max_i \text{Rang } B_i \Rightarrow q \geq m$ ($\text{rang } B_1 = m$).

La majoration est évidente.

On va chercher à quelle condition on peut avoir $q = m$.

- Condition nécessaire :

On a :

$$(a) \quad B_i = \sum_{j=1}^m \lambda_j^i U_j V_j^t \quad i=1, \dots, p \quad U_j, V_j \in K^m \quad j=1, \dots, m$$

$$\lambda_j^i \in K.$$

Soit U la matrice dont les colonnes sont les vecteurs U_1, \dots, U_m et V^t la matrice dont les lignes sont les vecteurs lignes V_1^t, \dots, V_m^t . Les matrices U et V^t sont inversibles. En effet, B_1 étant régulière les vecteurs U_1, \dots, U_m doivent être linéairement indépendants car dans le cas contraire B_1 pourrait s'exprimer avec moins de m matrices de rang un et donc ne serait plus régulière (de même pour les vecteurs V_1, \dots, V_m).

On peut écrire (a) sous la forme :

$$B_i = \begin{pmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_m^i \end{pmatrix} U V^t \quad i=1, \dots, p$$

On a donc bien, puisque U et V^t sont inversibles :

$$U^{-1} B_i (V^t)^{-1} = \begin{pmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_m^i \end{pmatrix} \quad i=1, \dots, p.$$

La condition nécessaire du théorème est donc bien démontrée.

- Condition suffisante :

Si on a :

$$P B_i Q = \begin{pmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_m^i \end{pmatrix},$$

On en déduit aussitôt si P et Q sont inversibles :

$$B_i = \begin{pmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_m^i \end{pmatrix} P^{-1} Q^{-1} \quad i=1, \dots, p.$$

Par conséquent, le rang tensoriel des matrices B_1, \dots, B_p est bien égal à m .

Le théorème 2 précédemment donné est équivalent au présent théorème si toutes les matrices sont inversibles. \square

Théorème 4

/ Soient p matrices B_i ($i=1, \dots, p$) de $\mathcal{M}_{m,m}(K)$ telles que l'une d'entre elles au moins soit régulière.

Le rang tensoriel de ces p matrices B_i est égal à $m+k$ si k est le plus petit nombre de lignes et de colonnes à ajouter aux matrices B_i ($i=1, \dots, p$), pour obtenir p matrices B'_1, \dots, B'_p de $\mathcal{M}_{m+k, m+k}(K)$ possédant la propriété suivante :

Il existe deux matrices P et Q de $\mathcal{M}_{m+k, m+k}(K)$ régulières telles que toutes les matrices $P B'_i Q$ soient diagonales. /

La relation (b) s'écrit également (U et V étant inversibles par construction) :

$$(c) \quad U^{-1} B_i^! V^{t-1} = \begin{pmatrix} \lambda_1^i & & 0 \\ & \ddots & \\ 0 & & \lambda_{m+k}^i \end{pmatrix} \quad i=1, \dots, p,$$

Donc si $m+k$ est égal au rang tensoriel des p matrices B_i , on peut effectivement leur ajouter k lignes et k colonnes de façon à obtenir p nouvelles matrices $B_i^!$ de $\mathcal{M}_{m+k, m+k}(K)$ vérifiant (c). Réciproquement, s'il existe p matrices $B_i^!$ de $\mathcal{M}_{m+k, m+k}$ vérifiant (c), leur rang tensoriel est au plus égal à $m+k$ et donc aussi le rang tensoriel des matrices B_i ($i=1, \dots, p$). Le rang tensoriel des matrices B_i est donc bien égal au plus petit k tel que l'on ait cette propriété. \square

Les théorèmes précédents illustrent la difficulté du calcul du rang tensoriel de p matrices B_i , $i=1, \dots, p$.

Il n'existe pour le moment aucun procédé général de calcul du rang tensoriel de p matrices B_i , $i=1, \dots, p$. Cependant, dans le cas de deux matrices B_1 et B_2 de $\mathcal{M}_{n, n}(C)$, une caractérisation complète de leur rang tensoriel a été récemment donnée par GASTINEL (15) dans le cas où il existe au moins une matrice de l'espace engendré par B_1 et B_2 qui soit régulière. On a le théorème suivant cf. GASTINEL (15) :

Théorème 5

/ Le rang tensoriel du couple A, B , ($A, B \in \mathcal{M}_{n, n}(C)$), A inversible est égal à $dd(C_\alpha) + n$ pour presque toutes les valeurs de α , $dd(C_\alpha)$ étant le défaut diagonal de la matrice $C_\alpha = (A^{-1} B - \alpha I) A^{-1} B$, c'est-à-dire le nombre minimum de lignes et de colonnes qu'il faut ajouter à C_α pour la rendre diagonalisable.

Si $e_k(u), e_{k+1}(u), \dots, e_n(u)$ sont les diviseurs élémentaires de la matrice $C_\alpha - uI$ et si k est le plus grand indice i tel que $e_i(u)$ n'ait que des racines simples on a :

$$dd(C_\alpha) = n - k \quad .$$

Par contre, le cas d'un faisceau singulier de matrices reste à élucider. Avant de passer à l'étude, dans le chapitre suivant du rang tensoriel de certains espaces de matrices, on termine cette étude des propriétés générales du rang tensoriel par un exemple et ensuite par la description d'un programme de calcul "approché" du rang tensoriel d'un ensemble de matrices.

Exemple 1

Produit de deux matrices 2×2 .

Soit A, B et C trois matrices de $\mathcal{M}_{2,2}(K)$.

$$C = AB .$$

On pose :

$$A = \begin{pmatrix} x_1 & x_3 \\ x_2 & x_4 \end{pmatrix} , \quad B = \begin{pmatrix} y_1 & y_2 \\ y_3 & y_4 \end{pmatrix}$$

Alors :

$$C = \begin{pmatrix} x_1 y_1 + x_3 y_3 & x_1 y_2 + x_3 y_4 \\ x_2 y_1 + x_4 y_3 & x_2 y_2 + x_4 y_4 \end{pmatrix}$$

Si l'on veut calculer le produit de deux matrices arbitraires A et B, il est nécessaire de calculer les quatre formes bilinéaires suivantes :

$$x_1 y_1 + x_3 y_3 , \quad x_1 y_2 + x_3 y_4 , \quad x_2 y_1 + x_4 y_3 , \quad x_2 y_2 + x_4 y_4 .$$

Les quatre matrices B_i ($i=1, \dots, 4$) de $\mathcal{M}_{4,4}(K)$ associées à ces formes bilinéaires sont les suivantes :

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} , \quad B_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} ,$$
$$B_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} , \quad B_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

La matrice $B_2(Z) = \sum_{i=1}^4 Z_i B_i$ est la matrice suivante :

$$B_2(Z) = \begin{pmatrix} Z_1 & Z_2 & 0 & 0 \\ Z_3 & Z_4 & 0 & 0 \\ 0 & 0 & Z_1 & Z_2 \\ 0 & 0 & Z_3 & Z_4 \end{pmatrix} .$$

On pose $X^t = (x_1, x_2, x_3, x_4)$ et $Y^t = (y_1, y_2, y_3, y_4)$.

La forme trilinéaire associée à $B(Z)$ est $F : K^4 \times K^4 \times K^4 \rightarrow K$ définie par :

$$F(X, Y, Z) \rightarrow X^t B_2(Z) Y \quad \forall (X, Y, Z) \in K^4 \times K^4 \times K^4 .$$

On a donc :

$$F(X, Y, Z) \rightarrow (x_1 z_1 + x_2 z_3) y_1 + (x_1 z_2 + x_2 z_4) y_2 \\ + (x_3 z_1 + x_4 z_3) y_3 + (x_3 z_2 + x_4 z_4) y_4 .$$

D'après le théorème 1 les trois matrices $B(Z)$, $B'(X)$, $B''(Y)$, ont le même rang tensoriel (les x_i , y_i , z_i sont ici considérées comme des indéterminées).

On a :

$$B_2'(X) = \begin{pmatrix} x_1 & 0 & x_3 & 0 \\ 0 & x_1 & 0 & x_3 \\ x_2 & 0 & x_4 & 0 \\ 0 & x_2 & 0 & x_4 \end{pmatrix} ,$$

$$B_2''(Y) = \begin{pmatrix} y_1 & y_2 & 0 & 0 \\ 0 & 0 & y_1 & y_2 \\ y_3 & y_4 & 0 & 0 \\ 0 & 0 & y_3 & y_4 \end{pmatrix} .$$

Le rang tensoriel de ces matrices est manifestement au plus égal à huit. STRASSEN (4') a donné une formule permettant de calculer le produit AB en sept multiplications, formule qui ne suppose pas la commutativité des x_i et des y_j entre-eux et qui donc correspond à une décomposition de $B_2(Z)$ (et de $B_2'(X)$ et $B_2''(Y)$) en sept matrices anti-scalaires.

Cette décomposition est la suivante :

$$\begin{aligned}
 B_2(Z) = & Z_1 \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}^{(001-1)} + Z_4 \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}^{(1-100)} \\
 & + (Z_1 - Z_4) \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}^{(1001)} + (Z_1 + Z_2) \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix}^{(0001)} \\
 & + (Z_3 + Z_4) \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}^{(1000)} + (Z_2 + Z_3) \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}^{(0101)} \\
 & + (Z_1 + Z_3) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^{(1010)} .
 \end{aligned}$$

Comme la commutativité des x_i et des y_j n'est pas supposée dans ces formules, elles peuvent s'appliquer au cas où les éléments de A et de B appartiennent seulement à un anneau non-commutatif, en particulier au cas où ce sont des matrices : l'application récursive des formules de STRASSEN montre que le produit de deux matrices $n \times n$ peut se faire en $n^{\log_2 7}$ multiplications cf. STRASSEN (1).

Dans le paragraphe 5, de ce chapitre, on démontrera que le rang tensoriel de $B_2(Z)$ est exactement égal à sept, donc que les formules de STRASSEN sont optimales pour effectuer le produit de deux matrices 2×2 .

Exemple 2 Produit de deux matrices A et B de $\mathcal{M}_{n,n}(K)$.

Les éléments de la matrice A (resp. B) sont désignés par X_i (resp. Y_i) pour $i=1, \dots, n^2$. On pose :

$$A = \begin{pmatrix} X_1 & X_{n+1} & \dots & X_{n^2-n+1} \\ X_2 & X_{n+2} & & \vdots \\ \vdots & & & \vdots \\ X_n & X_{2n} & & X_{n^2} \end{pmatrix}, \quad B = \begin{pmatrix} Y_1 & Y_2 & \dots & Y_n \\ Y_{n+1} & \dots & Y_{2n} & \\ \vdots & & \vdots & \\ Y_{n^2-n+1} & \dots & Y_{n^2} & \end{pmatrix}$$

$$X^t = (X_1, \dots, X_{n^2}), \quad Y^t = (Y_1, \dots, Y_{n^2}), \quad Z^t = (Z_1, \dots, Z_{n^2})$$

Soit C la matrice produit A B .

On pose :

$$C = \begin{pmatrix} C_1 & C_2 & \dots & C_n \\ C_{n+1} & \dots & C_{2n} & \\ \vdots & & \vdots & \\ C_{n^2-n+1} & \dots & C_{n^2} & \end{pmatrix} .$$

On a donc n formes bilinéaires $X^t B_i Y$ et la matrice $B_n(Z) = \sum_{i=1}^n Z_i B_i$ a la forme suivante :

$$B_n(Z) = \begin{pmatrix} \begin{array}{c} Z_1 \dots Z_n \\ \vdots \\ Z_{n^2-n+1} \dots Z_{n^2} \end{array} & & & \\ & \begin{array}{c} Z_1 \dots Z_n \\ \vdots \\ Z_{n^2-n+1} \dots Z_{n^2} \end{array} & & \\ & & \dots & \\ & & & \begin{array}{c} Z_1 \dots Z_n \\ \vdots \\ Z_{n^2-n+1} \dots Z_{n^2} \end{array} \end{pmatrix} \quad B_n(Z) \in \mathcal{M}_{n^2, n^2}(K).$$

La matrice $B_n(Z)$ est constituée de n blocs diagonaux identiques. Etudier la complexité du produit de deux matrices revient à étudier le rang tensoriel de cette matrice $B_n(Z)$.

La détermination du rang tensoriel de $B_n(Z)$ est encore un problème ouvert pour $n > 2$.

On va appliquer sur cette matrice $B(Z)$ de $\mathcal{M}_{n^2, n^2}(K)$ la propriété 8 de réduction du rang tensoriel que l'on a déjà énoncée.

En effet, considérons la matrice de $\mathcal{M}_{n^2, n^2}(K)$ ayant des zéros partout sauf en les huit positions suivantes :

$$\begin{aligned} & (i_1+k_1n, j_1+k_1n) , (i_1+k_1n, j_2+k_1n) , (i_2+k_1n, j_1+k_1n) , \\ & (i_2+k_1n, j_2+k_1n) \text{ et } (i_1+k_2n, j_1+k_2n) , (i_1+k_2n, j_2+k_2n) , \\ & (i_2+k_2n, j_1+k_2n) , (i_2+k_2n, j_2+k_2n) . \end{aligned}$$

Le rang tensoriel de cette matrice est égal à sept car on peut lui appliquer la décomposition de Strassen.

Comme on peut décomposer $B(Z)$ en une somme de $\lfloor \frac{n}{2} \rfloor^3$ telles matrices de rang tensoriel égal à sept (au lieu de huit) on obtient :

$$\text{Rt } B(Z) \leq n^3 - \lfloor \frac{n}{2} \rfloor^3 .$$

(Ce résultat a été donné pour la première fois par GASTINEL (10').

6 . CALCUL APPROCHE DU RANG TENSORIEL.

On termine ce chapitre par la description d'une méthode de calcul approché du rang tensoriel si $K = \mathbb{R}$.

Soit F le tenseur de $\mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$. On peut poser :

$$F = (f^{i,j,k}) \quad i=1, \dots, n , j=1, \dots, n ; k=1, \dots, n.$$

La norme du tenseur F est notée $\|F\|$ (norme euclidienne).

$F(X,Y,Z) = \sum_{i,j,k} f^{i,j,k} X_i Y_j Z_k$ est la forme trilinéaire correspondante à F.

On a :

$$\|F\|^2 = \sum_{i,j,k} (f^{i,j,k})^2 .$$

Un tenseur Z de rang tensoriel égal à q peut s'écrire sous la forme suivante :

$$Z = \sum_{j=1}^q x^Y \otimes y^Y \otimes z^Y \quad x^Y, y^Y, z^Y \in \mathbb{R}^n$$

Considérons le produit scalaire $\langle Z, F \rangle$ on a :

$$\langle Z, F \rangle = \left(\sum_{\gamma=1}^q x^\gamma \otimes y^\gamma \otimes z^\gamma \right) \cdot \left(\sum_{i,j,k} f^{i,j,k} l_i \otimes l_j \otimes l_k \right)$$

$$\langle Z, F \rangle = \sum_{\gamma=1}^q F(x^\gamma, y^\gamma, z^\gamma) .$$

On a toujours la relation : $\frac{\langle Z, F \rangle}{\|Z\| \|F\|} \leq 1$.

Théorème

/ Le rang tensoriel de F est le plus petit entier q tel que l'on ait :

$$\max_{x^Y, y^Y, z^Y} \left(\frac{\sum_{\gamma=1}^q F(x^\gamma, y^\gamma, z^\gamma)}{\|Z\| \|F\|} \right) = 1 . /$$

□ En effet la quantité $\frac{\langle Z, F \rangle}{\|Z\| \|F\|}$ qui représente le cosinus de l'angle entre

Z et F atteint son maximum si effectivement F s'exprime sous la forme :

$$F = \sum_{\gamma=1}^q x^\gamma \otimes y^\gamma \otimes z^\gamma ,$$

c'est-à-dire si F a un rang tensoriel égal à q. □

On peut aisément programmer une méthode de recherche du maximum dans R^{3nq} de la fonction :

$$\varphi(X^1, \dots, X^q, Y^1, \dots, Y^q, \dots, Z^1, \dots, Z^q) = \frac{\sum_{\gamma=1}^q F(X^\gamma, Y^\gamma, Z^\gamma)}{\|Z\| \|F\|}$$

Si $Z = \sum_{\gamma=1}^q X^\gamma \otimes Y^\gamma \otimes Z^\gamma$

on a :

$$\|Z\|^2 = \sum_{\gamma=1}^q \|X^\gamma\|^2 \|Y^\gamma\|^2 \|Z^\gamma\|^2 + 2 \sum_{\gamma > \mu} \langle X^\gamma, X^\mu \rangle \langle Y^\gamma, Y^\mu \rangle \langle Z^\gamma, Z^\mu \rangle.$$

On applique cette méthode pour différentes valeurs de q en espérant en trouver une, q_0 , telle que le maximum que l'on obtienne soit égal à un.

Exemple

On a appliqué cette méthode au problème du calcul du rang de la forme trilineaire provenant du produit de deux matrices (exemple 1 précédent)

$$F(X, Y, Z) = (x_1 z_1 + x_2 z_3) y_1 + (x_1 z_2 + x_2 z_3) y_2 + (x_3 z_1 + x_4 z_3) y_3 + (x_3 z_2 + x_4 z_4) y_4$$

On a dans ce cas :

$$\|F\| = 2 \sqrt{2}$$

Le maximum de la fonction $\frac{\sum_{\gamma=1}^q F(X^\gamma, Y^\gamma, Z^\gamma)}{\|Z\|}$ est donc $2 \sqrt{2}$.

Pour q = 8, on obtient cette valeur au bout de quelques itérations.

Pour q = 7, on obtient une suite stationnaire, au bout d'une centaine d'itérations d'une méthode de gradient. La valeur obtenue est $2 \sqrt{2}$ à 10^{-3} près

Pour q = 6, la suite de valeurs obtenues converge vers une valeur plus petite.

REFERENCES DU CHAPITRE BIII

- (8') BROCKETT R.W., DOBKIN D., "On the optimal evaluation of a set of bilinear forms".
5th Annual ACM symposium on theory of computing.
Austin Texas, April 30, May 2, (1973) 88-95.
- (10') GASTINEL, N., "Le rang tensoriel d'un ensemble de matrices. Applications". Séminaire d'Analyse Numérique, Grenoble N° 159. (octobre 1972).
- (15) GASTINEL N., "Le problème de l'extension minimale diagonale d'un opérateur linéaire". Séminaire d'Analyse Numérique, Grenoble n° 235, (novembre 1975).
- (16) HOPCROFT, J., MUSINSKI, J., "Duality applied to the complexity of matrix multiplications and other bilinear forms".
5th Annual ACM symposium on theory of computing.
Austin, Texas, April 30, May 2, (1973), 73-87.
- (12') LAFON, J.C., "Optimum computation of p bilinear forms".
J. Linear Algebra, 10, (1975, 225-240.
- (4') STRASSEN, V., "Vermeidung von divisionen".
Crelle J. Für die Reine und Angew. Mathematik (1973).

CHAPITRE B IV

BASES TENSORIELLES - APPLICATIONS

PLAN

1. Un théorème d'optimalité
2. Espace des matrices cycliques
 - a/ Unicité de la base tensorielle
 - b/ Application au produit de convolution
 - c/ Inversion d'une matrice cyclique
3. Espace des matrices de Hankel (ou de Toeplitz)
 - a/ Bases tensorielles
 - b/ Applications au calcul du produit de deux polynomes
 - c/ Inversion en $O(n)$ multiplications d'une matrice de Toeplitz triangulaire inférieure
 - d/ Division de deux polynomes
4. Espace des matrices ayant des symétries particulières
 - a/ Matrices symétriques
 - b/ Matrices centro-symétriques
 - c/ Matrices horizontales (verticales) symétriques
 - d/ Matrices roto-symétriques droites (gauches)
 - e/ Matrices à éléments complexes

1 . UN THEOREME D'OPTIMALITE.

Dans ce chapitre, on se propose de chercher les espaces V de $M_{n,n}(K)$ tels que leur rang tensoriel soit égal à leur dimension. Un tel espace V doit donc posséder une base tensorielle, c'est-à-dire une base constituée de matrices de rang un.

Cette étude systématique est surtout motivée par le résultat suivant

Théorème 1

/ Si le rang tensoriel de l'espace $\{B_i\}$ engendré par les p matrices B_i est égal à sa dimension, alors le nombre minimal de multiplications "générales" nécessaires pour calculer les p polynômes $X^{t_{B_i}} Y$ $i=1, \dots, p$ est égal à $\dim \{B_i\}$ même si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est supposé commutatif. /

□ Soit q le nombre minimal de multiplications "générales" nécessaires pour résoudre $P(B_1, \dots, B_p)$ si $K[x_1, \dots, x_m, y_1, \dots, y_n]$ est commutatif. D'après le théorème B2,1 on a :

$$B_i = \sum_{j=1}^q \alpha_j^i (A_1^j B_2^{j^t} + A_2^j B_1^{j^t}) \quad i=1, \dots, p .$$

On a donc toujours : (1)

$$(1) \quad q \geq \dim \{B_i\}$$

car les p matrices B_i s'expriment sous forme de combinaison linéaire de q matrices (de rang deux).

D'autre part, le nombre minimal de multiplications "générales" nécessaires dans le cas commutatif est évidemment inférieur ou égal au nombre minimal du cas non-commutatif, c'est-à-dire de $Rt \{B_i\}$. On a donc :

$$(2) \quad q \leq Rt \{B_i\}$$

Si $Rt \{B_i\} = \dim \{B_i\}$ on a bien d'après (1) et (2) :

$$q = \dim \{B_i\} \quad . \quad \square$$

Tous les résultats de ce chapitre sont donc valables que $K[x_1, \dots, x_m, y_1, \dots, y_n]$ soit commutatif ou non. Les principaux espaces de matrices que l'on va étudier sont les espaces des matrices cycliques, des matrices de Hankel (ou de Toeplitz), des matrices symétriques. On explicite à chaque fois les formes bilinéaires à calculer et les formules optimales.

2 . ESPACE DES MATRICES CYCLIQUES.

a/ Unicité de la base tensorielle ; cf. LAFON (23).

Théorème 2

/ Le rang tensoriel de l'espace C_n des matrices cycliques de $M_{n,n}(K)$ est égal à sa dimension si le corps K possède une racine nème principale de l'unité. Les matrices formant une base tensorielle de C_n sont uniques (à un facteur près). /

□ Soit $\lambda \in K$ la racine nème principale de l'unité.

λ doit vérifier les propriétés suivantes :

$$\lambda \neq 1 \quad \lambda^n = 1 \quad \text{et} \quad \sum_{j=0}^{n-1} \lambda^{jq} = 0 \quad \text{pour } 1 \leq q < n .$$

Soit C une matrice cyclique quelconque de l'espace C_n .

Elle peut s'écrire sous la forme suivante :

$$(3) \quad C = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_n & x_1 & \dots & x_{n-1} \\ \dots & \dots & \dots & \dots \\ x_2 & x_3 & \dots & x_1 \end{pmatrix} \quad x_i \in K \quad i=1, \dots, n .$$

On a évidemment : $\dim C_n = n$.

Il s'agit de montrer que l'on a aussi : $\text{Rt } C_n = n$.

Pour cela, il faut construire une base tensorielle de C_n , c'est-à-dire qu'il faut trouver n matrices cycliques, de rang un et linéairement indépendantes.

Soit C_0, C_1, \dots, C_{n-1} les matrices suivantes :

$$C_q = \begin{pmatrix} 1 & \lambda^q & \lambda^{2q} & \dots & \lambda^{(n-1)q} \\ \lambda^{(n-1)q} & 1 & \lambda^q & \dots & \lambda^{(n-2)q} \\ \dots & \dots & \dots & \dots & \dots \\ \lambda^q & \lambda^{2q} & \lambda^{3q} & \dots & 1 \end{pmatrix} \quad (q=0, \dots, n-1) .$$

Ces n matrices C_0, \dots, C_{n-1} sont cycliques, de rang un, et linéairement indépendantes.

Elles forment donc bien une base tensorielle de l'espace C_n . On a donc bien : $\text{Rt } C_n = n$.

D'autre part, il est facile de constater que les matrices cycliques de rang un doivent être égales, à un coefficient près, à l'une des matrices C_i ($i=0, \dots, n-1$) précédentes. En effet, une matrice cyclique C , écrite comme en (3), sera de rang un si tous ses termes sont non nuls et vérifient les relations :

$$\frac{x_n}{x_1} = \frac{x_1}{x_2} = \frac{x_2}{x_3} = \dots = \frac{x_{n-1}}{x_n} .$$

Si on pose : $\frac{x_1}{x_2} = \frac{1}{t}$, on obtient forcément la solution suivante :

$$x_2 = tx_1, \quad x_3 = t^2x_1, \dots, x_k = t^{k-1}x_1, \dots, x_n = t^{n-2}x_1 .$$

Comme d'autre part, on doit avoir :

$$x_1^2 - x_2x_n = 0 ,$$

on obtient la condition :

$$x_1^2(1 - t^n) = 0 .$$

Mais x_1 doit être non nul, par conséquent t doit être une racine nème de l'unité. Pour avoir n matrices cycliques de rang un, linéairement indépendantes, il faut donc avoir n telles racines distinctes.

Donc, si λ désigne une racine nème principale de l'unité, on aura les matrices C_0, C_1, \dots, C_{n-1} à un coefficient près dans toute base tensorielle de C_n . \square

Remarque 1

D'après ce qui précède, si l'on prend $K = \mathbb{R}$, l'espace C_n des matrices cycliques de $\mathcal{M}_{n,n}(\mathbb{R})$ ne possède pas de bases tensorielles dès que n est supérieur à 2. Si $n = 2$, comme -1 et 1 sont deux racines distinctes de l'unité, l'espace C_2 sur \mathbb{R} possède une base tensorielle constituée des deux matrices :

$$C_0 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} , \quad C_1 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} .$$

b/ Application au produit de convolution.

Théorème 2'

/ Le nombre minimal de multiplications "générales" nécessaires pour calculer les n composantes du produit de convolution de deux vecteurs de K^n est égal à n . Il existe un seul type de formule optimale. /

□ On pose :

$$X^t = (x_0, x_1, \dots, x_{n-1}) \quad ,$$

$$Y^t = (y_0, y_1, \dots, y_{n-1}) \quad .$$

Soit $Z = X * Y$ le produit de convolution (circulaire) des deux vecteurs X et Y

Si $Z^t = (z_0, z_1, \dots, z_{n-1})$, on peut écrire :

$$z_k = x_0 y_{(k)} + x_1 y_{(k+1)} + \dots + x_{n-1} y_{(n+k-1)} \quad k=0, \dots, n-1$$

$$((k) \equiv k \pmod{n}).$$

Les matrices B_0, \dots, B_{n-1} qui définissent des n formes bilinéaires Z_0, Z_1, \dots, Z_{n-1} engendrent justement l'espace C_n des matrices cycliques. Le nombre minimum de multiplications "générales" nécessaires pour calculer Z est donc n d'après le théorème 1 puisque $\text{Rt } C_n = \dim C_n = n$.

Soit d'autre part, C_0, C_1, \dots, C_{n-1} une base tensorielle de C_n (on vient de voir que toutes les autres sont du type $k_0 C_0, \dots, k_1 C_1, \dots, k_{n-1} C_{n-1}$ avec $k_i \in K$). Toute matrice C de C_n peut donc s'écrire sous la forme :

$$C = \sum_{q=0}^{n-1} \alpha_q C_q \quad \alpha_q \in K \quad q=0, \dots, n-1 .$$

- Les coefficients $\alpha_0, \dots, \alpha_{n-1}$ de cette décomposition sont solutions du système suivant :

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \lambda & \dots & \lambda^{n-1} \\ \cdot & \cdot & \dots & \cdot \\ 1 & \lambda^{n-1} & \dots & \lambda^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} .$$

L'inverse de la matrice de ce système est la matrice :

$$\frac{1}{n} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \lambda^{-1} & \dots & \lambda^{-(n-1)} \\ \dots & \dots & \dots & \dots \\ 1 & \lambda^{-(n-1)} & \dots & \lambda^{-(n-1)^2} \end{pmatrix}$$

On a donc :

$$\alpha_q = \frac{1}{n} \sum_{j=0}^{n-1} \lambda^{-jq} x_{j+1} .$$

Par conséquent, l'algorithme optimum consiste en l'emploi des formules suivantes pour calculer les composantes Z_0, Z_1, \dots, Z_{n-1} de Z à partir de celles de X et de Y :

$$Z_k = \frac{1}{n} \sum_{p=0}^{n-1} \lambda^{kp} \left(\sum_{q=0}^{n-1} \lambda^{(n-q)p} x_q \right) \left(\sum_{q=0}^{n-1} \lambda^{qp} y_q \right) \quad k=0, \dots, n-1 .$$

(On peut appliquer alors la FFT et donc utiliser en tout $O(n \log n)$ opérations arithmétiques).

Remarque 2

Tout ce qui vient d'être dit à propos du rang tensoriel de l'espace C_n des matrices cycliques de $M_{n,n}(K)$ reste vrai si K est simplement un anneau (commutatif ou non) possédant une racine nème principale de l'unité et tel que n ait un inverse dans K .

c/ Inversion d'une matrice cyclique

On sait que l'inverse d'une matrice cyclique est aussi une matrice cyclique. Soit donc C une matrice cyclique de $M_{n,n}(K)$ régulière définie par x_1, x_2, \dots, x_n , soit C^{-1} son inverse et y_1, \dots, y_n les coefficients de sa première ligne. Pour calculer cette inverse il suffit de déterminer les valeurs de y_1, \dots, y_n . Pour cela il suffit donc de résoudre le système suivant :

$$\begin{pmatrix} x_1 & \dots & x_n \\ x_n & x_1 & \dots & x_{n-1} \\ \vdots & & \ddots & \\ x_2 & \dots & x_n & x_1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_n \\ \vdots \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Posons

$$X^t = (x_1, \dots, x_n)$$

$$Y^t = (y_1, y_n, y_{n-1}, \dots, y_2) .$$

Le $e_1^t = (1 \ 0 \ \dots \ 0)$.

Le système suivant peut s'écrire :

$$X * Y = e_1 .$$

Supposons que le corps K possède une racine n^{ème} de l'unité (que l'on puisse y appliquer la transformée de Fourier). On a alors :

$$\mathcal{F}(X) \times \mathcal{F}(Y) = \mathcal{F}(e_1)$$

Posons $\mathcal{F}(X)^t = (x'_1, \dots, x'_n)$

$$\mathcal{F}(Y)^t = (y'_1, y'_n, \dots, y'_2)$$

$$\mathcal{F}(e_1) = (a_1, \dots, a_n)$$

On a donc alors :

$$y'_i = a_i / x'_i \quad (n \text{ divisions à effectuer})$$

$$\mathcal{F}(Y)^t = \left(\frac{a_1}{x'_1}, \dots, \frac{a_n}{x'_n} \right)$$

On aura donc :

$$Y = \mathcal{F}^{-1}(\mathcal{F}(Y))$$

On peut donc énoncer le théorème suivant :

Théorème 3

/ Si le corps K possède une racine nème de l'unité, alors on peut inverser une matrice cyclique avec n divisions "générales" seulement (et en $O(n \log n)$ opérations arithmétiques totales avec la FFT). /

Remarque 1

Notons au passage que le produit de deux matrices cycliques nécessite aussi n multiplications générales et $O(n \log n)$ opérations arithmétiques totales si on utilise la FFT (il suffit de calculer $X * Y$).

Remarque 2

L'intérêt de notre approche réside surtout dans le fait que l'on a démontré qu'il fallait obligatoirement utiliser la transformée de Fourier (et son inverse) pour calculer le produit de convolution de deux vecteurs en le nombre minimum de multiplications "générales". Mais cela n'entraîne aucun résultat en ce qui concerne la complexité de ce calcul au point de vue du total des opérations arithmétiques utilisées. Cet aspect a été étudié par MORGENSTERN (25).

Une intéressante interprétation de la FFT a d'autre part été présentée par FIDUCCIA (19).

3 . ESPACE DES MATRICES DE HANKEL (OU DE TOEPLITZ) cf LAFON (22).

a/ Bases tensorielles.

Théorème 4

/ Le rang tensoriel de l'espace H_n (resp. \mathcal{E}_n) des matrices de Hankel (resp. de Toeplitz) de $\mathcal{M}_{n,n}(K)$ est égal à sa dimension, c'est-à-dire à $2n-1$, dès lors que le corps K possède $2n-1$ éléments distincts. /

□ Définition 1 Matrice de Hankel

Une matrice H de $\mathcal{M}_{n,n}(K)$ sera dite de Hankel si elle peut s'écrire sous la forme suivante :

$$H = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \vdots & & & \\ x_n & x_{n+1} & \dots & x_{2n-1} \end{pmatrix} \quad (1)$$

Définition 2 Matrice de Toeplitz

/ $T \in \mathcal{M}_{n,n}(K)$ sera dite de Toeplitz si les éléments d'une parallèle à la première diagonale sont égaux. /

Désignons par H^n le sous-espace de $\mathcal{M}_{n,n}(K)$ formé par les matrices de Hankel, et par \mathcal{L}^n le sous-espace de $\mathcal{M}_{n,n}(K)$ formé par les matrices de Toeplitz.

H^n est un sous-espace de dimension $2n-1$. Par conséquent le rang tensoriel de H^n est au moins $2n-1$. Nous allons montrer dans la suite, qu'il existe une base tensorielle de H^n constituée d'exactly $2n-1$ matrices anti-scalaires.

D'après la forme générale d'une matrice de Hankel, il est immédiat de constater que toutes les matrices de Hankel de rang un ont l'une des formes suivantes (à un coefficient multiplicatif près) :

$$H_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & & \\ 0 & \dots & & 0 \end{pmatrix}, \quad H_{2n-1} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \\ \vdots & & \\ 0 & \dots & 1 \end{pmatrix},$$

$$H(\lambda) = \begin{pmatrix} 1 & \lambda & \lambda^2 & \dots & \lambda^{n-1} \\ \lambda & \lambda^2 & & \dots & \lambda^n \\ \vdots & & & & \\ \lambda^{n-1} & & & \dots & \lambda^{2n-2} \end{pmatrix}.$$

Il s'agit maintenant de voir si l'on peut en trouver $2n-1$ qui soient linéairement indépendantes.

Peut-on choisir $2n-1$ valeurs $\lambda_1, \lambda_2, \dots, \lambda_{2n-1}$ de telle manière que les $2n-1$ matrices $H(\lambda_1), H(\lambda_2), \dots, H(\lambda_{2n-1})$ soient linéairement indépendantes ?

On doit avoir dans ce cas :

$$\alpha_1 H(\lambda_1) + \alpha_2 H(\lambda_2) + \dots + \alpha_{2n-1} H(\lambda_{2n-1}) = 0 \Rightarrow \alpha_1 = \alpha_2 = \dots = \alpha_{2n-1} = 0$$

Ceci donne le système suivant de $2n-1$ équations en $\alpha_1, \alpha_2, \dots, \alpha_{2n-1}$:

$$\begin{aligned} \alpha_1 &+ \alpha_2 &+ &\dots &+ \alpha_{2n-1} &= 0 \\ \alpha_1 \lambda_1 &+ \alpha_2 \lambda_2 &+ &&+ \alpha_{2n-1} \lambda_{2n-1} &= 0 \\ \alpha_1 \lambda_1^2 &+ \alpha_2 \lambda_2^2 &+ &&+ \alpha_{2n-1} \lambda_{2n-1}^2 &= 0 \\ &\vdots &&&& \\ \alpha_1 \lambda_1^{2n-2} &+ \alpha_2 \lambda_2^{2n-2} &+ &&+ \alpha_{2n-1} \lambda_{2n-1}^{2n-2} &= 0 \end{aligned}$$

La matrice de ce système est une matrice de Vandermonde.

Elle est régulière si et seulement si ses coefficients sont tous distincts et non nuls :

$$\forall i \neq j \quad \lambda_i \neq \lambda_j \quad \text{et} \quad \lambda_i \neq 0 \quad i=1, \dots, 2n-1$$

En choisissant $2n-1$ valeurs $\lambda_1, \dots, \lambda_{2n-1}$ distinctes et non nulles, nous obtiendrons donc $2n-1$ matrices $H(\lambda_1), \dots, H(\lambda_{2n-1})$ linéairement indépendantes. Comme elles sont toutes de Hankel elles forment une base de H^n , et comme elles sont toutes de rang un, cette base est une base tensorielle de H^n .

Le rang tensoriel de H^n est donc bien $2n-1$.

Soit H une matrice de H^n , écrite comme en (1).

Alors, on peut exprimer H sous forme d'une combinaison linéaire des $2n-1$ matrices $H(\lambda_i)$, de la manière suivante :

$$H = \sum_{i=1}^{2n-1} (\alpha_i^t \cdot x) H(\lambda_i) \quad \cdot \quad x^t = (x_1, \dots, x_{2n-1}) \quad \alpha_i \in K^{2n-1}$$

Pour obtenir les α_i il suffit de résoudre le système de $2n-1$ équations de I avec en second membre le vecteur x .

Le résultat similaire pour l'espace \mathcal{L}_n des matrices de Toeplitz provient du fait que toute matrice de Toeplitz se déduit d'une matrice de Hankel par le produit d'une matrice de permutation. En appliquant cette permutation aux matrices $H(\lambda_1), \dots, H(\lambda_{2n-1})$ qui forment une base tensorielle de H^n , on obtient une base tensorielle de \mathcal{L}_n formée des matrices

$$T(\lambda_1), \dots, T(\lambda_{2n-1});$$

$$\text{avec } T(\lambda_i) = \begin{pmatrix} \lambda_i^{n-1} & \lambda_i^n & \dots & \lambda_i^{2n-2} \\ \vdots & \vdots & & \vdots \\ \lambda_i & & & n-1 \\ 1 & \lambda_i & \lambda_i^2 & \dots & \lambda_i \end{pmatrix}$$

Les paramètres λ_i ($i=1,2,\dots,2n-1$) doivent être tous distincts et non nuls. \square

Théorème 4'

/ Toutes les bases tensorielles de l'espace H^n (resp. \mathcal{L}^n) se répartissent en quatre familles dont deux dépendent de $2n-2$ paramètres, une autre de $2n-3$ paramètres et la dernière de $2n-1$ paramètres. /

\square Les matrices $H(\lambda_1), \dots, H(\lambda_{2n-1})$ forment une base tensorielle pour tous les choix des paramètres $\lambda_1, \dots, \lambda_{2n-1}$ tels que :

$$\lambda_i \neq \lambda_j \quad \text{si } i \neq j \quad \text{et} \quad \lambda_i \neq 0 \quad i=1, \dots, 2n-1$$

Nous avons donc ainsi une famille dépendant bien de $2n-1$ paramètres.

En choisissant seulement $2n-2$ paramètres $\lambda_1, \dots, \lambda_{2n-2}$, nous pouvons obtenir deux types de base, en rajoutant soit la matrice H_1 , soit la matrice H_{2n-1} aux matrices $H(\lambda_1), \dots, H(\lambda_{2n-2})$.

$$(\lambda_i \neq \lambda_j \quad \text{et} \quad \lambda_i \neq 0) .$$

La dernière famille sera constituée par les deux matrices H_1, H_{2n-1} et $2n-3$ matrices indépendantes $H(\lambda_1), \dots, H(\lambda_{2n-3})$

$$(\lambda_i \neq \lambda_j \quad \text{et} \quad \lambda_i \neq 0) .$$

Le résultat pour l'espace \mathcal{L}^n se déduit du résultat pour H^n . Il faut considérer les matrices $T(\lambda_i)$ au lieu des matrices $H(\lambda_i)$, et les matrices T_1, T_{2n-1} au lieu des matrices H_1, H_{2n-1} :

$$T_1 = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \\ 1 & 0 & \dots & 0 \end{pmatrix}, \quad T_{2n-1} = \begin{pmatrix} 0 & \dots & 1 \\ \vdots & & \\ 0 & \dots & 0 \end{pmatrix} . \square$$

Dans $\mathcal{M}_{n+1, n+1}(K)$, les $2n+1$ matrices M_i engendrent un sous-espace de dimension $2n+1$, qui est en fait l'espace des matrices de Hankel $(n+1) \times (n+1) : H^{n+1}$.

D'après le théorème 4, nous savons que cet espace possède une base tensorielle de $2n+1$ matrices anti-scalaires H_1, \dots, H_{2n+1} .

On pourra donc écrire :

$$M_i = \sum_{j=1}^{2n+1} \lambda_j^i H_j$$

Donc la forme bilinéaire $a {}^t M_i b$ s'écrira :

$$a {}^t M_i b = \sum_{j=1}^{2n+1} \lambda_j^i a {}^t H_j b$$

Si l'on ne compte que les multiplications entre coefficients des polynomes (multiplications générales), on aura $2n+1$ multiplications à effectuer au lieu de $(n+1)^2$ pour le calcul de c_0, \dots, c_{2n} .

Aux quatre familles de bases tensorielles de H^{n+1} , correspondront quatre types de formules.

Nous aurons donc à effectuer pour le calcul de c_0, \dots, c_{2n}

soit $2n+1$ multiplications du type :

$$(a_0 + a_1 \lambda_i + \dots + a_n \lambda_i^n) (b_0 + b_1 \lambda_i + \dots + b_n \lambda_i^n)$$

soit $2n$ multiplications du type :

$$(a_0 + a_1 \lambda_i + \dots + a_n \lambda_i^n) (b_0 + b_1 \lambda_i + \dots + b_n \lambda_i^n)$$

plus le produit $a_0 \times b_0$ ou $a_n \times b_n$.

Enfin, pour le quatrième type de formule, nous aurons à effectuer les deux produits $a_0 \times b_0$ et $a_n \times b_n$, ainsi que $2n-1$ produits du type :

$$(a_0 + a_1 \lambda_i + \dots + a_n \lambda_i^n) (b_0 + b_1 \lambda_i + \dots + b_n \lambda_i^n)$$

Pour ces quatre types de formules les paramètres λ_i doivent être distincts et non nuls.

Dans la pratique, ces formules n'auront d'intérêt que si les produits du type $\lambda_i^p \times a_p$ sont plus économiques que les produits du type $a_p \times b_q$. On prendra par exemple pour les λ_i des puissances successives de la base du système de numération utilisé. □

Exemple

$$n = 2 \quad p_1(t) = a_0 + a_1 t + a_2 t^2 \quad \text{et} \quad p_2(t) = b_0 + b_1 t + b_2 t^2$$

$$c_0 = a_0 b_0, \quad c_1 = a_0 b_1 + a_1 b_0, \quad c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0,$$

$$c_3 = a_1 b_2 + a_2 b_1, \quad c_4 = a_2 b_2.$$

Prenons le choix le plus simple pour λ_1, λ_2 et λ_3 .

$$\lambda_1 = 1, \quad \lambda_2 = -1, \quad \lambda_3 = 2.$$

La base tensorielle utilisée sera constituée des matrices :

$$H_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad H_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad H_3 = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \quad H_4 = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 4 & 8 \\ 4 & 8 & 16 \end{pmatrix}$$

$$H_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Soit :

$$H = \begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ x_3 & x_4 & x_5 \end{pmatrix}$$

une matrice de Hankel : Calculons les α_i tels que :

$$H = \alpha_1 H_1 + \alpha_2 H_2 + \alpha_3 H_3 + \alpha_4 H_4 + \alpha_5 H_5$$

il faut résoudre le système de Vandermonde :

$$\alpha_2 - \alpha_3 + 2\alpha_4 = x_2$$

$$\alpha_2 + \alpha_3 + 4\alpha_4 = x_3$$

$$\alpha_2 - \alpha_3 + 8\alpha_4 = x_4$$

On obtient :

$$\alpha_2 = x_2 + \frac{1}{2}(x_3 - x_4)$$

$$\alpha_3 = \frac{x_3}{2} - \frac{1}{6}(2x_2 + x_4)$$

$$\alpha_4 = \frac{1}{6}(x_4 - x_2)$$

$$\alpha_1 = x_1 - x_3 + \frac{1}{2}(x_4 - x_2) \quad \text{et} \quad \alpha_5 = x_5 + 2(x_2 - x_4) - x_3$$

En particulier nous aurons :

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = -\frac{1}{2}H_1 + H_2 - \frac{1}{3}H_3 - \frac{1}{6}H_4 + 2H_5$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = -H_1 + \frac{1}{2}H_2 + \frac{1}{2}H_3 - H_5 \quad (\text{III})$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \frac{1}{2}H_1 - \frac{1}{2}H_2 - \frac{1}{6}H_3 + \frac{1}{6}H_4 - 2H_5$$

Les cinq produits à calculer seront :

$$p_1 = a_0 b_0, \quad p_5 = a_2 b_2$$

et

$$p_2 = (a_0 + a_1 + a_2)(b_0 + b_1 + b_2)$$

$$p_3 = (a_0 - a_1 + a_2)(b_0 - b_1 + b_2)$$

$$p_4 = (a_0 + 2a_1 + 4a_2)(b_0 + 2b_1 + 4b_2)$$

et on aura les formules suivantes (dédites de III)

$$a_0 b_1 + a_1 b_0 = -\frac{1}{2} p_1 + p_2 - \frac{1}{3} p_3 - \frac{1}{6} p_4 + 2p_5$$

$$a_0 b_2 + a_1 b_1 + a_2 b_0 = -p_1 + \frac{1}{2} p_2 + \frac{1}{2} p_3 - p_5$$

$$a_1 b_2 + a_2 b_1 = \frac{1}{2} p_1 - \frac{1}{2} p_2 - \frac{1}{6} p_3 + \frac{1}{6} p_4 - 2p_5$$

Remarque

Si K est un corps possédant une racine $2n+1$ ème principale de l'unité λ , on choisira $\lambda_i = \lambda^i$. La matrice du système à résoudre est maintenant la matrice suivante :

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \lambda & & \lambda^{2n} \\ \vdots & & & \\ 1 & \lambda^{2n} & \dots & \lambda^{(2n)(2n)} \end{pmatrix},$$

dont l'inverse est la matrice :

$$\frac{1}{2n+1} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \lambda^{-1} & \dots & \lambda^{-(2n)} \\ \cdot & \cdot & \dots & \cdot \\ 1 & \lambda^{-2n} & \dots & \lambda^{(-2n)2n} \end{pmatrix}.$$

On obtient dans ces conditions, les expressions suivantes pour le calcul des coefficients C_0, \dots, C_{2n} :

$$C_k = \frac{1}{2n+1} \left(\sum_{p=0}^{2n} \lambda^{-kp} \left(\sum_{q=0}^n \lambda^{qp} a_q \right) \left(\sum_{q=0}^n \lambda^{qp} b_q \right) \right)$$

(On peut encore ici employer la FFT et donc effectuer le calcul en $O(n \log n)$ opérations arithmétiques totales).

c/ Inversion en $O(n)$ multiplications d'une matrice de Toeplitz triangulaire inférieure

Théorème 6

Le produit d'une matrice de Toeplitz T de \mathcal{L}^n et d'un vecteur X de K^n peut se faire en $2n-1$ multiplications générales au plus.

□ En effet, d'après le théorème 4' toute matrice T de \mathcal{L}^n peut s'écrire sous la forme :

$$T = \sum_{i=1}^{2n-1} (\alpha_i^t \cdot t) u_i \cdot v_i^t$$

$u_i \cdot v_i^t$ étant une matrice de Toeplitz
 $t = (t_{-n+1}, \dots, t_0, \dots, t_{n-1})$
 $\alpha_i \in K^{2n-1}$

On a donc :

$$T \cdot X = \sum_{i=1}^{2n-1} (\alpha_i^t \cdot t) (v_i^t \cdot X) \cdot U_i$$

On a donc à effectuer les $2n-1$ multiplications générales :

$$(\alpha_i^t \cdot t) \times (v_i^t \cdot X)$$

Notations

On désigne par \mathcal{L}_I^n l'ensemble des matrices de Toeplitz $n \times n$ triangulaires inférieures .

L'inversion d'une matrice de Toeplitz générale peut se faire en $O(n^2)$ opérations multiplications. On va montrer dans la suite, que l'inversion d'une matrice de Toeplitz triangulaire inférieure (ou supérieure) peut se faire en $O(n)$ multiplications seulement.

Théorème 7

/ L'inverse d'une matrice de Toeplitz triangulaire inférieure est aussi de Toeplitz triangulaire inférieure. /

□ Soit $T \in \mathcal{L}_I^n$.

Le i ème vecteur colonne X_i de T^{-1} est solution de l'équation :

$$T X_i = \ell_i ,$$

(ℓ_1, \dots, ℓ_n) base canonique de K^n .

Soit S la matrice suivante :

$$S = \begin{pmatrix} 0 & & & & 0 \\ 1 & & & & \\ 0 & 1 & & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} ,$$

$S \in \mathcal{L}_I^n$.

On a :

$$T X_1 = \ell_1 ,$$

Soit $S T X_1 = S \ell_1 \Rightarrow S T X_1 = \ell_2$.

Or $S T X_1 = T.(S X_1)$ (propriété de base).

Donc $T.(S X_1) = \ell_2$.

par conséquent :

$$X_2 = S X_1 .$$

De manière analogue, on démontre que :

$$X_p = S^{p-1} X_1 \quad p=2, \dots, n .$$

Soit $X_1^t = (x_0, x_1, \dots, x_{n-1})$

Alors l'inverse de T est la matrice :

$$T^{-1} = \begin{pmatrix} x_0 & 0 & 0 \\ x_1 & x_0 & \\ \vdots & & \\ x_{n-1} & & x_0 \end{pmatrix}$$

Donc $T^{-1} \in \mathcal{L}_{\mathbb{I}}^n$.

Remarque

Ceci montre déjà que le calcul de T^{-1} ne nécessite que $\frac{n(n+1)}{2}$ multiplications au plus.

Théorème 8

/ L'inversion d'une matrice de Toeplitz triangulaire inférieure peut s'effectuer en $O(n)$ multiplications générales au plus. /

□ On désigne par $M(n)$ le nombre de multiplications générales nécessaires pour inverser une matrice de Toeplitz triangulaire inférieure d'ordre n .

On évalue $M(2n)$ et $M(2n-1)$ en fonction de $M(n)$.

On considère tout d'abord l'inversion d'une matrice de $\mathcal{L}_{\mathbb{I}}^{2n}$.

L'unique système à résoudre se partitionne naturellement ainsi :

$$\left(\begin{array}{cc|cc} t_0 & 0 & 0 & \\ & & & 0 \\ t_{n-1} & & t_0 & \\ \hline t_n & & t_1 & t_0 \\ t_{2n-1} & & t_n & t_{n-1} & t_0 \end{array} \right) \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \\ \hline x_n \\ \vdots \\ x_{2n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}$$

On va donc d'abord calculer (x_0, \dots, x_{n-1}) en $M(n)$ opérations multiplications.

L'inverse de la matrice $\begin{pmatrix} t_0 & 0 & 0 \\ & & \\ t_{n-1} & & t_0 \end{pmatrix}$ sera la matrice $\begin{pmatrix} x_0 & 0 & 0 \\ \vdots & x_0 & \\ \vdots & \vdots & \\ x_{n-1} & & x_0 \end{pmatrix}$.

Il reste ensuite à résoudre le système suivant :

$$\begin{pmatrix} t_0 & 0 & 0 \\ \vdots & & \\ t_{n-1} & 0 & t_0 \end{pmatrix} \begin{pmatrix} x_n \\ \vdots \\ x_{2n-1} \end{pmatrix} = - \begin{pmatrix} t_n & t_1 \\ \vdots & \vdots \\ t_{2n-1} & t_n \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

Posons :

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} t_n & \dots & t_1 \\ \vdots & & \\ t_{2n-1} & \dots & t_n \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

D'après le théorème 4, le calcul de (y_0, \dots, y_{n-1}) peut se faire en $2n-1$ multiplications au plus.

On aura ensuite :

$$\begin{pmatrix} x_0 \\ \vdots \\ x_{2n-1} \end{pmatrix} = \begin{pmatrix} x_0 & 0 & 0 \\ \vdots & \vdots & \\ x_{n-1} & x_0 & \end{pmatrix} \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Ce dernier calcul nécessite lui aussi $2n-1$ multiplications au plus.

Finalement, on obtient donc :

$$M(2n) \leq M(n) + 2n-1 + 2n-1$$

$$\text{Soit : } M(2n) \leq M(n) + 4n-2$$

Soit maintenant le cas d'une matrice $T \in \mathcal{L}_I^{2n-1}$.

On partitionne le système à résoudre de la manière suivante :

$$\begin{pmatrix} t_0 & 0 & & 0 \\ \vdots & & & \\ t_{n-1} & t_0 & & \\ \hline t_n & t_1 & t_0 & \\ \vdots & & & \\ t_{2n-2} & t_{n-1} & t_{n-2} & t_0 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \\ \hline x_n \\ \vdots \\ x_{2n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

La résolution du premier système nécessite $M(n)$ opérations multiplications. Connaissant (x_0, \dots, x_{n-1}) , il reste à résoudre :

$$\begin{pmatrix} t_0 & 0 & 0 \\ \vdots & & 0 \\ t_{n-2} & & t_0 \end{pmatrix} \begin{pmatrix} x_n \\ \vdots \\ x_{2n-2} \end{pmatrix} = - \begin{pmatrix} t_n & t_1 \\ \vdots & \\ t_{2n-2} & \dots & t_{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix} .$$

Le calcul du second membre peut se faire en $2n-1$ multiplications au plus. Le premier système donne :

$$\begin{pmatrix} t_0 & 0 & 0 \\ \vdots & & \\ t_{n-2} & \dots & t_0 \\ \hline t_{n-1} & \dots & t_0 \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-2} \\ \hline x_{n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

La résolution de ce système permet donc d'avoir aussi l'inverse de la matrice :

$$\begin{pmatrix} t_0 & 0 & 0 \\ \vdots & & \\ t_{n-2} & \dots & t_0 \end{pmatrix} \text{ qui est la matrice } \begin{pmatrix} x_0 & 0 & 0 \\ \vdots & & \\ x_{n-2} & \dots & x_0 \end{pmatrix} .$$

Par conséquent, le calcul de x_n, \dots, x_{2n-2} sera terminé avec $2n-3$ produits supplémentaires.

On a donc :

$$M(2n-1) \leq M(n) + 2n-1 + 2n-3$$

$$M(2n-1) \leq M(n) + 4n-4$$

Comme $M(1) = 1$ on en déduit la majoration suivante de $M(n)$:

$$M(n) < 4n-3$$

(On peut montrer en fait que l'on a : $M(n) \leq 4n - \log n$). \square

Théorème 9

Le calcul du quotient et du reste dans la division d'un polynôme de degré $2n$ par un polynôme de degré n peut s'effectuer en $O(n)$ multiplications générales.

Soient les deux polynômes :

$$A(x) = a_0 + \dots + a_{2n} x^{2n}$$

$$B(x) = b_0 + \dots + b_n x^n$$

On veut déterminer $Q(x)$ et $R(x)$ tels que :

$$A(x) = B(x) Q(x) + R(x)$$

$$\deg R(x) < n$$

Si l'on pose :

$$Q(x) = q_0 + \dots + q_n x^n$$

Les coefficients de $Q(x)$ sont solutions du système :

$$\begin{pmatrix} b_n & 0 & 0 \\ b_{n-1} & b_n & \\ \vdots & & \\ b_0 & \dots & b_n \end{pmatrix} \begin{pmatrix} q_n \\ \vdots \\ q_0 \end{pmatrix} = \begin{pmatrix} a_{2n} \\ \vdots \\ a_n \end{pmatrix}$$

La matrice de ce système est une matrice de $\begin{pmatrix} n+1 \\ I \end{pmatrix}$.

D'après le théorème 8, le calcul de son inverse peut s'effectuer en $4n+1$ multiplications au plus.

Il restera à effectuer ensuite le produit de cette inverse par le vecteur du second membre, ce qui peut se faire en $2n+1$ multiplications au plus.

$Q(x)$ est donc déterminé en $6n+2$ multiplications au plus.

Enfin, la détermination de $R(x)$ revient au calcul de $B(x).Q(x)$, calcul qui nécessite $2n+1$ multiplications au plus.

Donc, $Q(x)$ et $R(x)$, peuvent être calculés en au plus $8n+3$ multiplications.

Ces résultats sont à comparer à ceux de Strassen (4^n) et de Sieveking (28) et de Kung (21). \square

Remarque 1

En modifiant la démonstration du théorème 3, on peut montrer que le calcul des coefficients du polynome produit de deux polynomes de degré n et m peut s'effectuer en $n+m+1$ multiplications générales.

En effet, comme dans le théorème 1, on peut montrer que les matrices $m \times n$ suivantes, forment un espace de rang tensoriel $n+m+1$ ($m > n$).

$$\begin{pmatrix} x_0 & x_1 & \dots & x_n \\ x_1 & x_2 & \dots & x_{n+1} \\ \vdots & & & \\ x_n & \dots & \dots & x_m \\ x_{n+1} & & & \\ \vdots & & & \\ x_m & & & x_{n+m} \end{pmatrix}$$

En utilisant ce résultat, on peut montrer le théorème suivant :

Théorème 9 bis :

Le calcul du quotient et du reste dans la division d'un polynome de degré m par un polynome de degré n peut s'effectuer en moins de $n+7(m-n)+3$ multiplications générales.

Remarque 2

Si K possède une racine $2n+1$ ème de l'unité on peut alors calculer le produit de deux polynomes de degré n en $O(n \text{ Log } n)$ opérations arithmétiques totales par l'utilisation de la transformation de Fourier rapide (FFT) (cf. Cooley-Tuckey (18)).

Dans ces conditions, l'inversion d'une matrice de Toeplitz triangulaire inférieure, ainsi que le calcul du quotient et du reste dans la division de deux polynomes peuvent se faire en utilisant seulement $O(n \text{ Log } n)$ opérations arithmétiques en tout.

(On aura en effet $M(2n) \leq M(n) + O(n \text{ Log } n)$).

Remarque 3

Le fait que le calcul des coefficients du polynome résultant du produit de deux polynomes de degré n peut se faire en $O(n \text{ Log } n)$ opérations arithmétiques à l'aide de la FFT est clairement montré si on interprète ce calcul comme une convolution de deux vecteurs de K^{2n+1} .

En effet, posons :

$$\begin{aligned} X^t &= (x_0, \dots, x_n, 0, \dots, 0)^t \\ Y^t &= (y_0, \dots, y_n, 0, \dots, 0)^t \\ Z^t &= (z_0, \dots, z_{2n})^t \end{aligned}$$

où x_0, \dots, x_n , resp. (y_0, \dots, y_n) sont les coefficients des deux polynomes $p_1(t)$ resp. $p_2(t)$ et (z_0, \dots, z_{2n}) les coefficients du polynome

$$P_3(t) = P_1(t) P_2(t).$$

Alors on a :

$$Z = X * Y .$$

Désignons par W_1, \dots, W_{2n+1} les $2n+1$ racines nème de l'unité du corps K .

On peut également interpréter le calcul de $P_1(t)P_2(t)$ en $2n+1$ multiplications générales et $O(n \text{ Log } n)$ opérations arithmétiques comme le processus suivant :

a) Evaluation de $P_1(t)$ et $P_2(t)$ en les $2n+1$ points w_1, \dots, w_{2n+1}
(coût $O(n \log n)$ opérations arithmétiques par la FFT).

b) Calcul de $P_3(t)$ en les $2n+1$ points w_1, \dots, w_{2n+1} :

$$P_3(w_i) = P_1(w_i) \cdot P_2(w_i) \quad i=1, \dots, 2n+1$$

($2n+1$ multiplications "générales").

c) Calcul de $P_3(t)$ par interpolation en les $2n+1$ points

$$(w_1, P_3(w_1)), \dots, (w_{2n+1}, P_3(w_{2n+1})).$$

($O(n \log n)$ opérations arithmétiques) .

(cf. Borodin-Munro (17) et Horowitz (20)).

Nous terminons ce paragraphe par l'étude de l'évaluation des fonctions symétriques des racines d'un polynôme, problème pour lequel Strassen (29) a pu donner une borne inférieure non linéaire (en $n \log_2 \frac{n}{e}$) et pour lequel les résultats précédents donnent un calcul en $n \log_2 n+1$ multiplications "générales".

Calcul optimum des fonctions symétriques d'un polynôme.

Soit $\rho_n(t)$ un polynôme de degré n de $K[t]$. K est supposé algébriquement clos. On peut prendre en fait $K = \mathbb{C}$.

Soit alors x_1, \dots, x_n les n racines du polynôme $\rho_n(t)$.

On peut écrire :

$$(t-x_1)(t-x_2)\dots(t-x_n) = t^n - \sigma_1 t^{n-1} + \dots + (-1)^n \sigma_n .$$

$\sigma_1, \dots, \sigma_n$ sont les fonctions symétriques des racines du polynôme :

ce sont en fait des polynômes homogènes de degré $1, 2, \dots, n$ de $K[x_1, x_2, \dots, x_n]$.

On peut donc se poser le problème suivant :

Quel est le coût minimal de l'évaluation des n fonctions symétriques à partir de $K \cup \{x_1, \dots, x_n\}$?

Désignons par $C(n)$ ce coût minimal. D'après le résultat du théorème 5 on a :

$$C(2n) \leq 2C(n) + 2n-1$$

$$C(2n+1) \leq C(n) + C(n+1) + 2n$$

Par conséquent, on a :

$$C(n) \leq n \log_2 n + 1 .$$

Cette majoration est atteinte pour $n = 2^p$.

Strassen dans (29) a pu démontrer d'autre part, le résultat suivant (sur la minoration de $C(n)$).

Théorème 10

Le coût minimal de calcul des fonctions symétrique $\sigma_1, \dots, \sigma_n$ des racines d'un polynôme de degré n est supérieur à : $n \log_2 \frac{n}{e}$.

On a donc $n \log_2 \frac{n}{e} \leq C(n) \leq n \log_2 n + 1$

Le coût minimum est donc en $n \log_2 n$.

Strassen utilise des résultats de géométrie algébrique pour obtenir la minoration ci-dessus.

Son approche est la suivante (cf.(29)).

K est un corps infini, algébriquement clos.

Soit f_1, \dots, f_p , p éléments de $K(x_1, \dots, x_n)$. On cherche à évaluer $C(\mathcal{F})$ ($\mathcal{F} = \{f_1, \dots, f_p\}$).

Les p fractions rationnelles f_1, \dots, f_p définissent une application rationnelle de K^n dans K^p :

$$\varphi : (x_1, \dots, x_n) \in K^n \rightarrow (f_1(x_1, \dots, x_n), \dots, f_p(x_1, \dots, x_n))$$

Le domaine de définition de φ dans K^n est l'ensemble des points de K^n où aucun dénominateur de f_1, \dots, f_p ne s'annule.

graphe $\varphi = \{x_1, \dots, x_n, f_1(x), \dots, f_p(x)\}$, $(x^t = (x_1, \dots, x_n)$,
graphe $\varphi \subset K^{n+p}$.

Dans l'espace projectif P^{n+p} sur K , on considère la fermeture du graphe de φ : c'est une variété algébrique, $W(f_1, \dots, f_p)$ de degré d . ($d = \text{Degré } W(f_1, \dots, f_p)$).

Grâce au théorème de Bezout, sur le degré de l'intersection de deux variétés, on peut relier le coût minimal $C(f_1, \dots, f_p)$ de calcul des p fractions rationnelles, au degré de la variété $W(f_1, \dots, f_p)$.

On a le théorème suivant (Strassen)

Théorème 11

Le coût minimal du calcul de p fractions rationnelles de $K(x_1, \dots, x_n)$ est lié au degré de la variété définie par ces p fractions par l'inégalité :

$$C(f_1, \dots, f_p) \geq \text{Lg}_2 (\text{Degré } (W(f_1, \dots, f_p))) .$$

Le théorème 8, sur l'évaluation des fonctions symétriques $\sigma_1, \dots, \sigma_n$ découle du fait que l'on peut montrer que l'on a :

$$\text{Degré } W(\sigma_1, \dots, \sigma_n) \geq n !$$

4. ESPACES DES MATRICES AYANT DES SYMETRIES PARTICULIERES

a/ Matrices symétriques, contre-symétriques

Une matrice S de $\mathcal{M}_{n,n}(K)$ est symétrique si $A = A^t$. On note par S_n l'espace des matrices symétriques de $\mathcal{M}_{n,n}(K)$.

Théorème 12

/ L'espace S_n des matrices symétriques de $\mathcal{M}_{n,n}(K)$ possède une base tensorielle (son rang tensoriel est égal à sa dimension). /

⌋ Une matrice S de S_n peut s'écrire de façon générale sous la forme suivante :

$$S = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{12} & S_{22} & \dots & S_{2n} \\ \vdots & & & \\ S_{1n} & S_{2n} & \dots & S_{nn} \end{pmatrix} \quad \begin{matrix} S_{ij} \in K & i=1, \dots, n \\ & j=i, \dots, n . \end{matrix}$$

On a donc : $\dim S_n = \frac{n(n+1)}{2} .$

Il s'agit de trouver $\frac{n(n+1)}{2}$ matrices symétriques de rang un et linéairement indépendantes.

Soit e_{ij} ($i=1, \dots, n$, $j=1, \dots, n$) la matrice de $n, n(K)$ dont tous les éléments sont égaux à zéro sauf l'élément (i, j) égal à un. On va prendre les $\frac{n(n+1)}{2}$ matrices symétriques suivantes :

$$SS_{ii} = e_{ii} \quad (i=1, \dots, n)$$

et

$$SS_{ij} = e_{ii} + e_{ij} + e_{ji} + e_{jj} \quad (i=1, \dots, n, j=i+1, \dots, n).$$

Ces $\frac{n(n+1)}{2}$ matrices sont bien symétriques, de rang un, et linéairement indépendantes. Elles constituent donc bien une base tensorielle de l'espace S_n . Toute matrice S de S_n s'exprime de la manière suivante en fonction de ces matrices :

$$S = \sum_{\substack{i=1 \\ j=i+1}}^n s_{ij} SS_{ij} + \sum_{i=1}^n (s_{ii} - \sum_{j=i+1}^n s_{ij} - \sum_{j=1}^{i-1} s_{ji}) e_{ii} . \quad \square$$

Application

Le calcul des $\frac{n(n+1)}{2}$ formes bilinéaires suivantes peut se faire, de façon optimale en $\frac{n(n+1)}{2}$ multiplications générales :

$$x_i y_i \quad (i=1, \dots, n) \quad \text{et} \quad x_i y_j + x_j y_i \quad (i=1, \dots, n, j=i+1, \dots, n).$$

□ En effet, les $\frac{n(n+1)}{2}$ matrices qui définissent ces formes bilinéaires engendrent justement l'espace S_n . Le résultat découle donc de l'application du théorème 1. Les formules optimales sont les suivantes :

$$x_i y_j + x_j y_i = (x_i + x_j)(y_i + y_j) - x_i y_i - x_j y_j \\ (i=1, \dots, n, j=i+1, \dots, n).$$

Les $\frac{n(n+1)}{2}$ multiplications à effectuer sont donc les multiplications du type $x_i y_i$ ($i=1, \dots, n$) et les multiplications du type $(x_i + x_j)(y_i + y_j)$ ($i=1, \dots, n, j=i+1, \dots, n$) □

Ce résultat a été énoncé pour la première fois (à notre connaissance) par FIDUCCIA (9).

Généralisations

Soient n^2 éléments p_{ij} de K ($i=1, \dots, n$, $j=1, \dots, n$) avec $p_{ii} \neq 0$ et considérons l'ensemble S'_n des matrices de $\mathcal{M}_{n,n}(K)$ qui s'écrivent sous la forme suivante :

$$S' = \begin{pmatrix} p_{11} s_{11} & p_{12} s_{12} & \dots & p_{1n} s_{1n} \\ p_{21} s_{21} & p_{22} s_{22} & \dots & p_{2n} s_{2n} \\ \vdots & & & \\ p_{n1} s_{n1} & p_{n2} s_{n2} & \dots & p_{nn} s_{nn} \end{pmatrix} \quad (2)$$

On peut énoncer le résultat suivant :

Théorème 12 bis

/ Le rang tensoriel de l'espace S'_n des matrices de $\mathcal{M}_{n,n}(K)$ qui s'écrivent sous la forme (2) ci-dessus est égal à sa dimension (si $p_{ii} \neq 0$, $i=1, \dots, n$). /

□ Il suffit de remarquer que l'on peut écrire (on raisonne sur les formes bilinéaires à calculer) : si $p_{ji} \neq 0$:

$$p_{ij} x_i y_j + p_{ji} x_j y_i = p_{ji} (x_i + x_j) (y_i + \frac{p_{ij}}{p_{ji}} y_j) - p_{ji} x_i y_i - p_{ij} x_j y_j \quad (3)$$

(Dans le cas $p_{ji} = 0$ mais $p_{ij} \neq 0$ on gardera la forme $p_{ij} x_i y_j$).

Si $p_{ii} \neq 0$ $i=1, \dots, n$ on doit calculer les n multiplications $x_i y_i$ et le calcul de la forme (3) n'introduit qu'une multiplication générale en plus. □

Corollaire 1

Le sous-espace des matrices **symétriques** de \mathcal{S}_n qui possède $2q$ zéros (en des positions symétriques fixes mais non sur la diagonale) a pour rang tensoriel sa dimension, c'est-à-dire $\frac{n(n+1)}{2} - q$.

b/ Matrices centro-symétriques, centro-antisymétriques.

Une matrice A de $\mathcal{M}_{n,n}(K)$ est dite centro-symétrique si elle

vérifie $A = Q A Q$ ($Q = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{pmatrix}$).

Exemple pour $n=4$ la forme générale d'une matrice centro-symétrique est la suivante :

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_8 & a_7 & a_6 & a_5 \\ a_4 & a_3 & a_2 & a_1 \end{pmatrix} .$$

La dimension de l'ensemble des matrices centro-symétriques de $\mathcal{M}_{n,n}(K)$ est égale à $\lceil \frac{n^2}{2} \rceil$, et on peut énoncer le résultat suivant :

Théorème 13

Le rang tensoriel de l'ensemble des matrices centro-symétriques de $\mathcal{M}_{n,n}(K)$ est égal à la dimension de cet espace : $\lceil \frac{n^2}{2} \rceil$.

- Désignons par A_n une matrice centro-symétrique générale.
Si $n=2p$ on peut écrire :

$$A_{2p} = \begin{pmatrix} a_1 & a_2 & \dots & a_{2p} \\ a_{4p-2} & 0 & 0 & 0 & a_{2p+1} \\ \vdots & & & & \\ a_{2p+1} & 0 & \dots & 0 & a_{4p-2} \\ a_{2p} & \dots & \dots & \dots & a_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & & & 0 \\ \vdots & & A_{2p-2} & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Le rang tensoriel de la première matrice de cette décomposition est égal à $4p-2$. En effet, on peut écrire :

$$(3) \begin{pmatrix} 0 & a_i & 0 & \dots & 0 & a_{2p-i+1} & 0 & \dots & 0 \\ 0 & & & & & & & & \\ \vdots & & & & & & & & \\ 0 & a_{2p-i+1} & \dots & a_i & 0 & \dots & 0 & & \end{pmatrix} \\
 = (a_i + a_{2p-i+1})/2 \begin{pmatrix} 0 & \dots & 1 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & & & & & & & & \\ & & & & & & & & \\ & & & & 1 & 0 & \dots & 1 & 0 & \dots & 0 \end{pmatrix} \\
 + (a_i - a_{2p-i+1})/2 \begin{pmatrix} 0 & \dots & 1 & 0 & \dots & 0 & 1 & \dots & 0 \\ & & & & & & & & \\ & & & & & & & & \\ 0 & & & 1 & 0 & & 1 & \dots & 0 \end{pmatrix} .$$

On a donc :

$$\text{Rt } A_{2p} \leq \text{Rt } A_{2p-2} + 4p-2$$

c'est-à-dire :

$$\text{Rt } A_{2p} \leq \sum_{i=1}^p 4(i-2)$$

$$\text{Rt } A_p \leq 2p^2 .$$

Comme la dimension de cet espace est égale à $\lceil \frac{n^2}{2} \rceil$, c'est-à-dire à $2p^2$ ici, on a bien :

$$\text{Rt } A_p = 2p^2 ,$$

et la décomposition explicitée ci-dessus donne les $2p^2$ matrices centro-symétriques qui génèrent l'espace.

Si $n = 2p+1$ le raisonnement est le même, mais il faut considérer aussi les deux matrices de rang un suivantes :

$$p+1 \rightarrow \begin{pmatrix} & 0 & & & \\ 1 & 0 & \dots & 0 & 1 \\ & 0 & & & \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} & 1 & & & \\ & 0 & & & \\ & \vdots & & & \\ & 0 & & & \\ & 1 & & & \\ \uparrow & p+1 & & & \end{pmatrix} .$$

On a donc :

$$\text{Rt } B_{2p+1} \geq \text{Rt } B_{2p-1} + 4p \quad ,$$

Soit :

$$\text{Rt } B_{2p+1} \leq 2p^2 + 2p+1 .$$

On a d'autre part :

$$\text{Rt } B_{2p+1} \geq \left\lceil \frac{(2p+1)^2}{2} \right\rceil$$

$$\text{Rt } B_{2p+1} \geq 2p^2 + 2p+1$$

Par conséquent :

$$\text{Rt } B_{2p+1} = 2p^2 + 2p + 1 . \quad \square$$

Généralisation

Comme le montre la décomposition (3), le résultat du théorème 11 tient au fait que toute matrice centro-symétrique se décompose en une somme de matrices du type suivant :

$$A = \begin{pmatrix} & 0 & & & 0 \\ & 0 & a_i & \dots & a_{n-i+1} \\ & & & & \\ & & a_{n-i+1} & a_i & \\ & 0 & & & 0 \end{pmatrix} .$$

Ces matrices sont de rang deux et regroupent deux paramètres (si l'ordre de la matrice est impair il faut rajouter les paramètres qui se trouvent sur la ligne (et la colonne) du milieu).

on voit que la matrice $\begin{pmatrix} p_1 a_1 & p_2 a_2 \\ p_2' a_2 & p_1' a_1 \end{pmatrix}$ ne peut être décomposée en

une somme de deux matrices de rang un de ce type (il faudrait choisir a_1 et a_2 de façon à annuler le déterminant qui vaut :

$p_1 p_1' a_1^2 - p_2 p_2' a_2^2$, ce qui est impossible sur \mathbb{R} ou \mathbb{Q} si $p_1 p_1' p_2 p_2' < 0$).

Par contre, si $p_1 p_1'$ et $p_2 p_2'$ sont de même signe alors la décomposition (4) est possible : dans le cas $p_1 p_1' < 0$ et $p_2 p_2' < 0$ il faudra prendre $\sqrt{-p_2 p_2'}$ et $\sqrt{-p_1 p_1'}$ dans la formule (4).

Théorème 14

/ Le rang tensoriel de l'ensemble des matrices centro-antisymétriques de $\mathcal{M}_{n,n}(\mathbb{K})$ est égal à la dimension de cet espace, c'est-à-dire à $\lceil \frac{n^2}{2} \rceil$. /

□ une matrice A est dite centro-antisymétrique si elle vérifie $A = -Q A Q$ (elle est anti-symétrique par rapport à son centre). Pour $n=4$ la matrice centro-symétrique générale s'écrit sous la forme suivante :

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ -a_8 & -a_7 & -a_6 & -a_5 \\ -a_4 & -a_3 & -a_2 & -a_1 \end{pmatrix}$$

Il est clair que dans ce cas les conditions (a) et (b) sont vérifiées par les matrices du type A' qui interviennent dans la décomposition de A comme en (3). Pour $n=4$ on écrira :

$$A = \begin{pmatrix} a_1 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -a_4 & 0 & 0 & -a_1 \end{pmatrix} + \begin{pmatrix} 0 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -a_3 & -a_2 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ a_5 & 0 & 0 & a_8 \\ -a_8 & 0 & 0 & -a_5 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a_6 & a_7 & 0 \\ 0 & -a_7 & -a_6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ,$$

Chacune de ses quatre matrices se décompose en une somme de deux matrices de rang un puisque l'on a :

$$\begin{pmatrix} a & b \\ -b & -a \end{pmatrix} = \frac{a+b}{2} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} + \frac{a-b}{2} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} . \quad \square$$

c/ Matrices horizontales (anti) symétriques,
verticales (anti) symétriques .

Définition

A sera dite :

- (α) horizontale symétrique si : $A = QA$,
- (β) verticale symétrique si : $A = AQ$,
- (γ) horizontale antisymétrique si : $A = -QA$,
- (δ) verticale antisymétrique si : $A = -AQ$.

On peut énoncer le théorème suivant :

Théorème 15

L'ensemble des matrices de $\mathcal{M}_{n,n}(K)$ de type α (ou (β) , (γ) , (δ)) a un rang tensoriel égal à sa dimension.

- Remarquons tout d'abord que si A est horizontale symétrique (resp. horizontale anti-symétrique) alors A^t est verticale symétrique (resp. verticale anti-symétrique). Par conséquent, comme la transposée d'une matrice a le même rang tensoriel que cette matrice, il suffit de montrer le théorème pour les matrices horizontales symétriques et horizontales anti-symétriques. Le résultat est alors immédiat dès que l'on examine la forme de ces matrices :

Pour $n=4$

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_5 & a_6 & a_7 & a_8 \\ a_1 & a_2 & a_3 & a_4 \end{pmatrix} \quad \text{est une matrice horizontale symétrique.}$$

$$A' = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ -a_5 & -a_6 & -a_7 & -a_8 \\ -a_1 & -a_2 & -a_3 & -a_4 \end{pmatrix} \quad \text{est une matrice horizontale antisymétrique.}$$

Si l'on fait tous les coefficients égaux à a sauf un égal à un on obtient bien une matrice de rang un. Donc l'espace des matrices horizontales symétriques ou horizontales anti-symétriques a un rang tensoriel égal à sa dimension (qui vaut $\frac{n^2}{2}$ si n est pair et $\frac{n(n+1)}{2}$ si n est impair).

Le résultat peut évidemment se généraliser aux matrices qui ont la forme suivante (exemple: pour $n = 4$) :

$$A = \begin{pmatrix} p_1 a_1 & p_2 a_2 & p_3 a_3 & p_4 a_4 \\ p_5 a_5 & p_6 a_6 & p_7 a_7 & p_8 a_8 \\ p'_5 a_5 & p'_6 a_6 & p'_7 a_7 & p'_8 a_8 \\ p'_1 a_1 & p'_2 a_2 & p'_3 a_3 & p'_4 a_4 \end{pmatrix} \quad p_i, p'_i \in K. \quad \square$$

d/ Matrices roto-symétriques droites (ou gauches).

Définition

Une matrice est dite roto-symétrique droite si elle vérifie $A = A^T Q$, elle est dite roto-symétrique gauche si elle vérifie $A = Q A^T$.

Exemples

Pour $n = 4$, une matrice roto-symétrique droite s'écrira sous la forme générale suivante :

$$A_4 = \begin{pmatrix} a_1 & a_2 & a_3 & a_1 \\ a_3 & a_4 & a_4 & a_2 \\ a_2 & a_4 & a_4 & a_3 \\ a_1 & a_3 & a_2 & a_4 \end{pmatrix},$$

et sa transposée sera une matrice roto-symétrique gauche.

On peut écrire A_4 sous la forme suivante :

$$A_4 = \begin{pmatrix} a_1 & 0 & 0 & a_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_1 & 0 & 0 & a_1 \end{pmatrix} + \begin{pmatrix} 0 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_3 & a_2 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ a_3 & 0 & 0 & a_2 \\ a_2 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a_4 & a_4 & 0 \\ 0 & a_4 & a_4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Ceci permet d'écrire A_4 comme une combinaison de six matrices de rang un (cf. a). D'autre part, on voit facilement qu'il en peut exister quatre matrices roto-symétriques droites de rang un et linéairement intépendantes.

On a donc :

$$5 \leq \text{Rt } B_4 \leq 6.$$

Ceci se généralise facilement pour l'ensemble des matrices roto-symétriques droites ou gauches.

Théorème 16

L'ensemble des matrices roto-symétriques droites (ou gauches) de $M_{n,n}(\mathbb{K})$ a un rang tensoriel $\text{Rt } B_n$ vérifiant :

$$\left\lceil \frac{n^2}{4} \right\rceil + 1 \leq \text{Rt } B_n \leq \left\lceil \frac{n^2}{2} \right\rceil - \left\lfloor \frac{n}{2} \right\rfloor.$$

□ Il suffit de remarquer que la matrice B_n roto-symétrique la plus générale s'écrit sous la forme :

$$B_n = \begin{pmatrix} a_1 & a_2 & \dots & a_{n-1} & a_1 \\ & a_{n-1} & & & a_2 \\ & \vdots & & B_{n-2} & \vdots \\ & a_2 & & & a_3 \\ a_1 & a_{n-1} & \dots & a_2 & a_1 \end{pmatrix} \quad a_i \in K \quad (i=1, \dots, n-1)$$

B_{n-2} désignant la matrice roto-symétrique la plus générale de dimension $(n-2, n-2)$.

La borne inférieure provient de ce que la dimension de l'espace des matrices roto-symétriques droites de $\mathcal{M}_{n,n}(K)$ est égale à $\lceil \frac{n^2}{4} \rceil$ et que l'on ne peut trouver de bases tensorielles pour cet espace. La borne supérieure s'obtient en raisonnant comme pour les matrices centro-symétriques. \square

e/ Matrices à éléments complexes.

Dans ce paragraphe on ne considère que les matrices de l'ensemble $\mathcal{M}_{n,n}(\mathbb{C})$.

Notations

Soit x un élément de \mathbb{C} . On désigne par \bar{x} son conjugué. Soit $A \in \mathcal{M}_{n,n}(\mathbb{C})$.

On désigne par :

\bar{A} la matrice dont tous les éléments sont les conjugués de ceux de A .

A^* la matrice transposée de la conjuguée de A ($A^* = \bar{A}^t$).

Définitions

On peut définir dans $\mathcal{M}_{n,n}(K)$ les nouveaux sous-espaces de matrices suivants :

- espaces des matrices hermitiennes ($A = A^*$),
- espaces des matrices anti-hermitiennes ($A = -A^*$),
- espaces des matrices contre-hermitiennes (anti-hermitiennes)
($A = K A^* K$, $A = -K A^* K$),
- espaces des matrices roto-hermitiennes droites (resp. gauches)
($A = A^* Q$ resp. $A = Q A^*$),
- espaces des matrices roto-anti-hermitiennes droites (resp. gauches)
($A = -A^* Q$, resp. $A = -Q A^*$),

- espaces des matrices centro-hermitiennes (resp. centro-anti-hermitiennes)
($A = Q \bar{A} Q$ resp. $A = -Q \bar{A} Q$) ,
- espaces des matrices verticales hermitiennes (resp. verticales anti-hermitiennes) ($A = \bar{A} Q$ resp. $A = -\bar{A} Q$) ,
- espaces des matrices horizontales hermitiennes (resp. horizontales anti-hermitiennes) ($A = Q \bar{A}$ resp. $A = -Q \bar{A}$) .

On peut étudier le rang tensoriel de tous ces espaces. Pour obtenir une majoration de ce rang tensoriel il suffit de multiplier par deux celle donnée pour les espaces correspondants précédemment étudiés (remplacer "hermitienne" par "symétrique"). Mais les bornes inférieures restent inchangées.

Le chapitre suivant va montrer comment on peut utiliser la connaissance du rang tensoriel de certains espaces de matrices pour obtenir une majoration du rang tensoriel de matrices plus complexes.

REFERENCES DU CHAPITRE B IV

- (17) BORODIN, A., MUNRO "Evaluating polynomials at many points.
Information processing letters (1971), 66-68.
- (18) COOLEY, J.W., TUKER, J.W. "An algorithm for the machine calculation
of complex Fourier series.
Math. Comput. 19 (90), 297-301 (april 1965).
- (19) FIDUCCIA, CM "Polynomial evaluation via the division algorithm.
The fast Fourier transform revisited".
Proceeding 4th annual ACM symposium on theory of Computing,
88-93 (1972).
- (20) HOROWITZ, F. "A fast method for interpolation using precondition g"
Information processing letters 1,4, 157-163 (1972).
- (21) KUNG HF " On computing reciprocals of power series"
Numer. Math. 22, 341-348 (1974).
- (22) LAFON, J.C. "Bases tensorielles des matrices de Toeplitz (et de
Hankel). Applications".
Numerische Mathematik, 23, 349-361 (1975).
- (23) LAFON J.C. "Complexite du calcul de certaines familles de formes
bilinéaires".
Séminaire, Grenoble n° 208 (octobre 1974).
- (24) MOENCK, R., BORODIN, AB "Fast modular transforms via division".
Conference Redord, IECE 13th annual symposium on switching and
automata theory, 90-96 (1972).
- (25) MORGENSTERN, J. "Note on a lower bound of the linear complexity of
the fast fourier transform".
J. ACM 20;2, 305-306 (1973).

- (26) NICHOLSON F.J "Algebraic theory of finite fourier transforms".
J. Computer and System sciences, 515, 524-547 (1971).
- (27) SHONHAGE, A., STRASSEN R. "Schnell multiplikation grosser zahlen
Computing 7, 281-292 (1971).
- (28) SIEVEKING, M. "An algorithm for division of power series!"
Computing 10, 153-156 (1972).
- (29) STRASSEN, V. "Die Berechnungskomplexität von elementarsymmetrischen
Funktionen und von interpolations-koeffizienten".
Numer. Math. 20, 238-251 (1973).
- (4") STRASSEN, V. "Vermeidung von Divisionen".
Crelle J. für die Reine und Angew. Mathematik (1973).
- (30) WINOGRAD, S. "Some remarks on fast multiplication of polynomials".
In Complexity of sequential and parallel numerical algorithms.
Proc. Symposium Pittsburgh, (May 1973).
Edited by Traub. Academic Press.

CHAPITRE B V

MINORATION DU RANG TENSORIEL

RESULTATS D'OPTIMALITE

PLAN

1. Principes généraux de minoration du rang tensoriel.
 - a/ Utilisation des propriétés du rang tensoriel.
 - b/ Utilisation des transformations invariantes.
2. Produit de deux matrices.
 - a/ Borne inférieure dans le cas général.
 - b/ Optimalité du résultat de STRASSEN.
3. Produit optimal de deux quaternions.
4. Produit vectoriel de deux vecteurs-produit de LIE de deux matrices.
5. Rang tensoriel de l'ensemble des matrices anti-symétriques.

Conclusion .

INTRODUCTION

Dans le chapitre précédent, on a pu obtenir des formules de calculs optimales pour des problèmes $P(B_1, \dots, B_p)$ tels que l'espace $\{B_i\}$ admettait une base constituée de matrices de rang un. Dans ce cas, le nombre minimal de multiplications "générales" utilisées par ces formules est égal à la dimension de l'espace $\{B_i\}$ et ce résultat est valable même si $K[X, Y]$ est commutatif. Le théorème 1 du précédent chapitre, constitue donc notre premier résultat d'optimalité.

Il s'agit maintenant d'étudier les cas où l'on a :

$$\text{Rt } \{B_i\} > \dim \{B_i\} .$$

Il est alors plus difficile d'obtenir la valeur exacte du rang tensoriel de l'espace $\{B_i\}$. Cette valeur, q par exemple, est déterminée quand, d'une part on connaît q matrices de rang un avec lesquelles on peut exprimer les matrices B_i ($i=1, \dots, p$), et que d'autre part, on peut démontrer que le nombre minimal de matrices de rang un nécessaires pour exprimer les matrices B_i ($i=1, \dots, p$) est bien égal à q .

Dans le chapitre BIII, on a vu comment essayer de diminuer le rang tensoriel connu d'un ensemble de matrices. Comme il n'existe pas, pour le moment, de méthode effective de calcul du rang tensoriel de p matrices B_i (sauf pour le cas de deux matrices de $\mathcal{M}_{n,n}(C)$), on en est réduit à utiliser de manière astucieuse ces principes pour l'obtention de nouvelles formules de calcul de $P(B_1, \dots, B_p)$, c'est-à-dire, en fait, pour obtenir une majoration, la plus faible possible, du rang tensoriel des matrices B_i ($i=1, \dots, p$).

Dans le premier paragraphe de ce chapitre, on va présenter les méthodes générales utilisables pour minorer le rang tensoriel de p matrices B_i .

Dans les paragraphes suivants on appliquera ces méthodes successivement au calcul du produit de deux matrices, au calcul du produit de deux quaternions, au calcul du produit vectoriel de deux vecteurs et du produit de Lie de deux matrices, au calcul du rang tensoriel des matrices anti-symétriques.

1 . PRINCIPES GENERAUX DE MINORATION DU RANG TENSORIEL

a/ Utilisation des propriétés du rang tensoriel

Soient B_i ($i=1, \dots, p$) p matrices de $\mathcal{M}_{m,n}(K)$.

Le rang tensoriel de ces matrices ne dépendant que du sous-espace $\{B_i\}$ de $\mathcal{M}_{m,n}(K)$ qu'elles génèrent, on peut toujours supposer que ces matrices sont linéairement indépendantes ($\dim \{B_i\} = p$).

Le rang tensoriel de l'espace $\{B_i\}$ ($Rt \{B_i\}$) est supérieur ou égal à sa dimension, ainsi qu'au maximum du rang des matrices qui lui appartiennent.

On peut donc écrire :

$$(1) \quad \text{Max} (p, \text{Max}_{B \in \{B_i\}} (\text{Rang} (B))) \leq Rt \{B_i\} .$$

Soit $B(Z) = \sum_{i=1}^p Z_i B_i$ la matrice qui sert à paramétriser l'espace $\{B_i\}$.

En général on étudie le rang tensoriel de $\{B_i\}$ sur la matrice $B(Z)$ car on peut mieux visualiser certaines propriétés. De ce point de vue, il faut également considérer les matrices $B'(X)$ et $B''(Y)$ qui ont le même rang tensoriel que $B(Z)$ puisqu'elles sont associées à la même forme trilineaire $X^t B(Z) Y$. En particulier, on pourrait également écrire :

$$(1') \quad \text{Max} (m, \text{Max}_{X \in K^m} (\text{Rang} B'(X))) \leq Rt \{B_i\} ,$$

$$(1'') \quad \text{Max} (n, \text{Max}_{Y \in K^n} (\text{Rang} B''(Y))) \leq Rt \{B_i\} .$$

On ne retiendra évidemment que la plus grande de ces trois minoration possibles.

Le théorème suivant constitue, avec les remarques précédentes et avec l'utilisation (exposée ensuite) des transformations qui laissent invariante une forme trilineaire, le principal moyen d'obtention d'une minoration du rang tensoriel.

Théorème 1

/ Soit q le rang tensoriel de la matrice $B(Z) = \sum_{i=1}^p Z_i B_i$ (les p matrices

B_i étant supposées indépendantes). On désigne par T_{i_1, \dots, i_k} ($k < p$),

l'ensemble des vecteurs à p composantes obtenus à partir du vecteur de composantes Z_1, \dots, Z_p en substituant à Z_{i_1}, \dots, Z_{i_k} des combinaisons

linéaires des $p-k$ composantes inchangées. (i_1, \dots, i_k) désigne un k -uplet d'indices distincts compris entre 1 et p .

Alors, on a la minoration suivante de q :

$$q \geq \max_{k < p} [\max_{(i_1, \dots, i_k)} [\min_{Z' \in T_{i_1, \dots, i_k}} \text{Rt}(B(Z'))] + k] \quad . \quad /$$

Remarque

Dans l'énoncé de ce théorème, on peut évidemment remplacer $B(Z)$ soit par $B'(X)$, soit par $B''(Y)$.

□ Ecrivons la matrice $B(Z)$ sous la forme d'une combinaison linéaire de q matrices de rang un :

$$B(Z) = \sum_{j=1}^q (\lambda_j^t Z) U_j V_j^t \quad U_j \in K^m, V_j \in K^n, \lambda_j \in K^p \quad j=1, \dots, q.$$

Les matrices B_i ($i=1, \dots, p$) étant supposées linéairement indépendantes, on a : $\dim \{B_i\} = p$ et donc (cf. (1)) : $q \geq p$.

Il existe donc nécessairement p vecteurs linéairement indépendants parmi les q vecteurs $\lambda_1, \dots, \lambda_q$ de K^p .

On peut toujours supposer, sans restreindre la généralité du raisonnement, que les vecteurs $\lambda_1, \dots, \lambda_p$ sont ces p vecteurs linéairement indépendants. Pour tout entier k strictement inférieur à p , on peut résoudre le système suivant (de k équations à p inconnues Z_1, \dots, Z_p) :

$$(1) \quad \lambda_i^t Z = 0 \quad i=1, \dots, k \quad .$$

La résolution de ce système permet d'exprimer k de ces inconnues en fonction des $p-k$ inconnues restantes.

Soit $Z' = (Z'_1, \dots, Z'_p)$ le vecteur obtenu à partir de Z en substituant à ces k inconnues les combinaisons linéaires des $p-k$ restantes. On a :

$$\lambda_i^t Z' = 0 \quad i=1, \dots, k .$$

On peut donc écrire :

$$B(Z') = \sum_{j=k+1}^q (\lambda_j^t Z') U_j V_j^t$$

On a donc :

$$Rt (B(Z')) \leq q-k$$

C'est-à-dire :

$$(2) \quad q \geq Rt (B(Z')) + k$$

Désignons par Z_{i_1}, \dots, Z_{i_k} les k inconnues principales du système (1).

On a donc : $Z' \in T_{i_1, \dots, i_k}$.

Le raisonnement précédent montre seulement qu'il existe un vecteur Z' de T_{i_1, \dots, i_k} particulier pour lequel l'inégalité (2) soit vérifiée.

On peut donc écrire :

$$(3) \quad q \geq \min_{Z' \in T_{i_1, \dots, i_k}} (Rt (B(Z'))) + k .$$

Pour obtenir l'inégalité (3) il suffit en réalité d'annuler k des quantités $\lambda_1^t Z, \dots, \lambda_p^t Z$. La matrice $(p \times p)$ dont les lignes sont formées par $\lambda_1^t, \lambda_2^t, \dots, \lambda_p^t$ étant régulière, quel que soit le choix des indices i_1, \dots, i_k des colonnes, on peut trouver k indices de lignes j_1, \dots, j_k tels que la sous matrice $(k \times k)$ dont les termes se trouvent à l'intersection de ces k lignes et de ces k colonnes soit régulière.

On peut donc résoudre le système :

$$\lambda_{j_1}^t Z = \lambda_{j_2}^t Z = \dots = \lambda_{j_k}^t Z = 0 ,$$

en prenant Z_{i_1}, \dots, Z_{i_k} comme inconnues principales.

Par conséquent, pour tous les choix possibles des k indices i_1, \dots, i_p parmi l'ensemble $\{1, 2, \dots, p\}$, on peut obtenir l'inégalité :

$$q \geq \underset{Z' \in T_{i_1, \dots, i_k}}{\text{Min}} (\text{Rt}(B(Z')) + k) .$$

On peut donc écrire en réalité :

$$(4) \quad q \geq \underset{(i_1, \dots, i_k)}{\text{Max}} \left(\underset{Z' \in T_{i_1, \dots, i_k}}{\text{Min}} (\text{Rt}(B(Z')) + k) \right) .$$

Comme on peut écrire l'inégalité (4) pour toutes les valeurs de k comprises entre 0 et $p-1$, on obtient le résultat du théorème 1 :

$$q \geq \underset{k < P}{\text{Max}} \left[\underset{(i_1, \dots, i_k)}{\text{Max}} \left[\underset{Z' \in T_{i_1, \dots, i_k}}{\text{Min}} (\text{Rt}(B(Z')) + k) \right] \right] . \square$$

Exemple :

Montrons que le produit de deux matrices 2,2 triangulaires inférieures ne peut se faire en moins de quatre multiplications.

$$\text{Si } A = \begin{pmatrix} a_1 & 0 \\ a_2 & a_3 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} b_1 & 0 \\ b_2 & b_3 \end{pmatrix} \quad \text{sont deux}$$

matrices triangulaires inférieures quelconques, effectuer leur produit revient à calculer les trois formes bilinéaires suivantes :

$$a_1 b_1, a_2 b_1 + a_3 b_2, a_3 b_3 .$$

La matrice $B(Z)$ de cet exemple est la suivante :

$$B(Z) = \begin{pmatrix} Z_1 & 0 & 0 \\ Z_2 & 0 & 0 \\ 0 & Z_2 & Z_3 \end{pmatrix}$$

Appliquons le théorème 1 avec $k = 2$ et le choix $i_1 = 1$ et $i_3 = 3$ (on exprime Z_1 et Z_3 en fonction de Z_2) soit :

$$Z_1 = \alpha_1 Z_2, \quad Z_3 = \alpha_3 Z_2 \quad (\alpha_1, \alpha_3 \in K).$$

$$\text{Rt } B(Z) \geq \min_{\alpha_1, \alpha_3} \left[\text{Rt} \begin{pmatrix} \alpha_1 Z_2 & 0 & 0 \\ Z_2 & 0 & 0 \\ 0 & Z_2 & \alpha_3 Z_2 \end{pmatrix} \right] + 2$$

pour $Z_2 \neq 0$ la nouvelle matrice est de rang au moins égal à 2 donc :

$$\text{Rt } B(Z) \geq 2 + 2 = 4.$$

b/ Utilisations des transformations invariantes

Définition 1

Transformation laissant invariante une forme trilinéaire.

Soit : $F(X, Y, Z) = \sum_{i, j, k}^{m, n, p} f^{i, j, k} X_i Y_j Z_k$, une forme trilinéaire en

$X_1, \dots, X_m, Y_1, \dots, Y_n, Z_1, \dots, Z_p$.

On pose $X^t = (X_1, \dots, X_m)$, $Y^t = (Y_1, \dots, Y_n)$, $Z^t = (Z_1, \dots, Z_p)$.

Une transformation T de cette forme trilinéaire est définie par la donnée de trois matrices A, B, C régulières de tailles m, n et p respectivement. On posera : $T = (A, B, C)$.

La transformation $T = (A, B, C)$ sera dite "F invariante" si elle laisse invariante F , c'est-à-dire si on a l'identité :

$$F(AX, BY, CZ) = \sum_{i, j, k}^{m, n, p} f^{i, j, k} X_i Y_j Z_k ;$$

Exemple 1

On considère la forme trilinéaire associée au calcul du produit de deux matrices de taille n (cf. ex. 2(III)) :

$$F(X, Y, Z) = X^t B_n(Z) Y ,$$

$$\text{avec } X^t = (X_1, \dots, X_{n^2}) , \quad Y^t = (Y_1, \dots, Y_{n^2}) , \quad Z^t = (Z_1, \dots, Z_{n^2}) .$$

On va chercher une transformation laissant invariante F .

Soit A et B les deux matrices de taille n dont F représente le calcul des n^2 éléments de leur produit :

$$A = \begin{pmatrix} X_1 & X_{n+1} & \dots & X_{n^2-n+1} \\ \vdots & \vdots & & \vdots \\ X_n & X_{2n} & & X_{n^2} \end{pmatrix} , \quad B = \begin{pmatrix} Y_1 & Y_2 & \dots & Y_n \\ \vdots & \vdots & & \vdots \\ Y_{n^2-n+1} & & \dots & Y_{n^2} \end{pmatrix} .$$

Soit T une matrice de taille n régulière.

On a évidemment :

$$AB = (AT)(T^{-1}B)$$

Prenons :

$$AT = \begin{pmatrix} X'_1 & X'_{n^2-n+1} \\ \vdots & \vdots \\ X'_n & X'_{n^2} \end{pmatrix} , \quad T^{-1}B = \begin{pmatrix} Y'_1 & \dots & Y'_n \\ \vdots & & \vdots \\ Y'_{n^2-n+1} & & Y'_{n^2} \end{pmatrix}$$

$$X'^t = (X'_1, \dots, X'_{n^2}) , \quad Y'^t = (Y'_1, \dots, Y'_{n^2}) .$$

On peut écrire :

$$X' = R X \quad \text{et} \quad Y' = S Y \quad R, S \in \mathcal{M}_{n^2, n^2}(K) .$$

On voit facilement, en explicitant les termes de la matrice AT (resp. $T^{-1}B$) en fonction des termes de la matrice A (resp. B) que l'on

peut écrire :

$$R = \begin{pmatrix} T^t & 0 & \dots & 0 \\ 0 & T^t & & \\ \cdot & & \cdot & \\ \cdot & & & 0 \\ 0 & \dots & 0 & T^t \end{pmatrix} P, \quad S = \begin{pmatrix} (T^{-1})^t & 0 & \dots & 0 \\ 0 & (T^{-1})^t & & \\ \cdot & & \cdot & \\ \cdot & & & 0 \\ 0 & \dots & 0 & (T^{-1})^t \end{pmatrix} P$$

P étant une matrice de permutation de $\mathcal{M}_{n^2, n^2}(K)$.

Les deux matrices R et S étant inversibles, il est clair, de par cette construction, que la transformation (R^{-1}, S^{-1}, I) laisse invariante la forme $X^t B_n(Z) Y$.

Application à la minoration du rang tensoriel

Soit $F(X, Y, Z) = \sum_{i, j, k}^{m, n, p} f^{i, j, k} X_i Y_j Z_k$ une forme trilinéaire de rang

tensoriel égal à q.

On peut écrire :

$$F(X, Y, Z) = \sum_{j=1}^q (U_j^t X)(V_j^t Y)(W_j^t Z)$$

Soit $T = (A, B, C)$ une transformation qui laisse invariante la forme F :

$$F(AX, BY, CZ) = \sum_{j=1}^q f^{i, j, k} X_i Y_j Z_k.$$

Mais on a aussi :

$$(a) \quad F(AX, BY, CZ) = \sum_{j=1}^q (U_j^t AX)(V_j^t BY)(W_j^t CZ).$$

Si l'on peut choisir la transformation T de façon à faire apparaître dans (a) un produit par une forme très particulière, on pourra utiliser T pour obtenir une minoration du rang tensoriel de F.

Exemple :

Supposons que l'on puisse choisir A de telle façon que l'on ait :

$$U_1^t A = e_1$$

On aura alors :

$$(b) \quad F(AX, BY, CZ) = X_1 (V_1^t BY) (W_1^t CZ) + \sum_{j=2}^q (U_j^t AX) (V_j^t BY) (W_j^t CZ).$$

En faisant $X_1 = 0$, on obtient une nouvelle forme $F'(X', Y, Z)$ avec $X'^t = (X_2, \dots, X_m)$ de rang tensoriel au plus égal à $q-1$ d'après (b). Mais, du fait de l'invariance de F par la transformation T, on peut écrire explicitement la nouvelle forme F' :

$$F'(X', Y, Z) = \sum_{i=2, j=1, k=1}^{m, n, p} f^{i, j, k} X_i Y_j Z_k$$

et on aura :

$$q \geq \text{Rt}(F') + 1 .$$

Pour minorer le rang tensoriel de F il suffit donc de minorer le rang tensoriel de F' ce qui peut être plus facile.

On a utilisé cette technique pour démontrer l'optimalité de la méthode de STRASSEN pour calculer le produit de deux matrices deux deux, du produit de deux quaternions en huit multiplications réelles, du calcul du produit vectoriel de deux vecteurs de R^3 en cinq multiplications, ainsi que pour la détermination du rang tensoriel de l'espace des matrices antisymétriques.

2 . COMPLEXITE DU PRODUIT DE DEUX MATRICES

a/ Borne inférieure dans le cas général.

Théorème 2

/ Le calcul de la matrice produit de deux matrices de taille (m,n) et (n,p) nécessite au minimum le nombre de multiplications :

$$\text{Max } (p(m+n-1), n(m+p-1), m(n+p-1)). \quad /$$

□ Posons :

$$A = \begin{pmatrix} X_1 & \dots & X_{mn-m+1} \\ X_2 & & \\ \vdots & & \\ X_m & \dots & X_{mn} \end{pmatrix}, \quad B = \begin{pmatrix} Y_1 & Y_2 & \dots & Y_p \\ Y_{p+1} & & \dots & Y_{2p} \\ \vdots & & & \\ Y_{np-p+1} & \dots & \dots & Y_{np} \end{pmatrix}.$$

La matrice AB appartient à $\mathcal{M}_{m,p}(K)$. Le calcul du produit AB revient au calcul de mp formes bilinéaires en X_1, \dots, X_{mn} et Y_1, \dots, Y_{np} .

Posons $Z^t = (Z_1, \dots, Z_{mp})$.

La matrice B(Z) qui représente les mp formes bilinéaires à calculer peut s'écrire sous la forme suivante :

$$B(Z) = \left(\begin{array}{cccc} C(Z) & & & \\ & C(Z) & & 0 \\ & & 0 & \ddots \\ & & & & C(Z) \end{array} \right) \left. \vphantom{\begin{array}{cccc} C(Z) & & & \\ & C(Z) & & 0 \\ & & 0 & \ddots \\ & & & & C(Z) \end{array}} \right\} \begin{array}{l} n \text{ blocs} \\ n \text{ blocs} \end{array}$$

La matrice $C(Z)$, ayant m lignes et p colonnes, a la forme suivante :

$$C(Z) = \begin{pmatrix} Z_1 & \dots & Z_p \\ Z_{p+1} & \dots & Z_{2p} \\ \vdots & & \\ Z_{mp-p+1} & \dots & Z_{mp} \end{pmatrix} .$$

On a évidemment :

$$\text{Rt}(B(Z)) \geq m p$$

(On a en réalité $\text{Rt}(B(Z)) \geq \text{Max}(mn, np, pn)$ puisque le rang tensoriel de $B(Z)$ est égal à celui de $B'(X)$ et à celui de $B''(Y)$).

On va maintenant appliquer le théorème 1 de façon à obtenir une meilleure minoration du rang tensoriel de $B(Z)$.

On peut tout d'abord appliquer le théorème 1 avec :

$$k = mp - p .$$

On suppose que Z_{p+1}, \dots, Z_{mp} sont exprimés sous forme de combinaisons linéaires de Z_1, \dots, Z_p .

La matrice déduite de $B(Z)$ par le remplacement des Z_{p+1}, \dots, Z_{mp} par ces formes linéaires en Z_1, \dots, Z_p a un rang tensoriel supérieur ou égal à np :

en effet, il n'existe pas de combinaison linéaire de ses np colonnes qui donne le vecteur nul pour tous les choix possibles de Z_1, \dots, Z_p (Propriété 10, III).

On a donc en vertu du théorème 1 :

$$\text{Rt}(B(Z)) \geq mp - p + np$$

Soit $\text{Rt}(B(Z)) \geq p[m+n-1]$.

Maintenant il faut remarquer que le raisonnement que l'on vient de faire sur $B(Z)$ est aussi valable sur $B'(X)$ et sur $B''(Y)$ (et aussi sur les transposées de ces trois matrices). Comme ces six matrices ont le même rang tensoriel on en déduit alors la minoration suivante du rang tensoriel de $B(Z)$:

$$Rt(B(Z)) \geq \text{Max}(p(m+n-1), n(m+p-1), m(n+p-1)),$$

D'où le résultat du théorème. \square

Remarque

La considération des matrices $B(Z)$, $B'(X)$, $B''(Y)$ et de leurs transposées revient à dire que la minoration donnée ci-dessus sur le rang tensoriel de la matrice $B(Z)$ est une minoration du nombre de multiplications "générales" nécessaires pour calculer les éléments du produit d'une matrice de taille $m \times n$ par une matrice de taille $n \times p$, mais aussi une minoration du nombre de multiplications générales nécessaires pour calculer le produit de deux matrices dans les différents cas suivants :

matrice $m \times p$ par une matrice de taille $p \times n$
 matrice $n \times p$ par une matrice de taille $p \times m$
 matrice $n \times m$ par une matrice de taille $m \times p$
 matrice $p \times n$ par une matrice de taille $n \times m$
 matrice $p \times m$ par une matrice de taille $m \times n$.

Corollaire

Le produit de deux matrices de $M_{n,n}(K)$ nécessite au moins $2n^2 - n$ multiplications "générales".

Le résultat découle immédiatement du théorème précédent. Dans le cas $n = 2$, la borne ainsi obtenue vaut six et ne permet donc pas d'affirmer l'optimalité de la méthode de STRASSEN. Pour démontrer cette optimalité, on utilisera dans le théorème suivant la notion de transformations invariantes déjà exposée.

b/ Cas particulier du produit de deux matrices 2,2,...

Théorème 3

/ Le produit de deux matrices 2×2 ne peut se faire en moins de sept multiplications "générales". /

Il faut prouver que le rang tensoriel de la forme trilinéaire $X^t B_2(Z) Y$ est au moins égal à sept. Dans le cas présent, on a :

$$B_2(Z) = \begin{pmatrix} Z_1 & Z_2 & 0 & 0 \\ Z_3 & Z_4 & 0 & 0 \\ 0 & 0 & Z_1 & Z_2 \\ 0 & 0 & Z_3 & Z_4 \end{pmatrix} .$$

Soit $F(X,Z,Y) = X^t B_2(Z) Y$.

On a donc

$$F(X,Z,Y) = (X_1 Z_1 + X_2 Z_3) Y_1 + (X_1 Z_2 + X_2 Z_4) Y_2 \\ + (X_3 Z_1 + X_4 Z_3) Y_3 + (X_3 Z_2 + X_4 Z_4) Y_4 .$$

Soit q le rang tensoriel de cette forme trilinéaire, et écrivons $F(X,Z,Y)$ sous la forme :

$$F(X,Z,Y) = \sum_{j=1}^q (U_j^t X) (W_j^t Z) (V_j^t Y) \quad U_j, V_j, W_j \in K^4 \quad j=1, \dots, q.$$

On pose :

$$U_j^t = (U_j^1, U_j^2, U_j^3, U_j^4) \quad (j=1, \dots, q) .$$

Soit T une matrice régulière

$$T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} .$$

Dans ce cas, la matrice R de l'exemple 1 va s'écrire sous la forme suivante :

$$R = \begin{pmatrix} t_{11} & 0 & t_{21} & 0 \\ 0 & t_{11} & 0 & t_{21} \\ t_{12} & 0 & t_{22} & 0 \\ 0 & t_{12} & 0 & t_{22} \end{pmatrix} .$$

La transformation $T = (R^{-1}, S^{-1}, I)$ laisse $F(X, Z, Y)$ invariante.

On aura donc :

$$F(X, Z, Y) = \sum_{j=1}^q (U_j^t R^{-1} X) (W_j^t Z) (V_j^t S^{-1} Y) .$$

On va choisir T de manière à obtenir un terme très simple en $U_1^t R^{-1} X$. Il est immédiat de constater que la matrice R^{-1} a la même forme que la matrice R (ses éléments sont ceux de la matrice T^{-1}).

Posons :

$$R^{-1} = \begin{pmatrix} R_1 & 0 & R_3 & 0 \\ 0 & R_1 & 0 & R_3 \\ R_2 & 0 & R_4 & 0 \\ 0 & R_2 & 0 & R_4 \end{pmatrix} , \quad U_1^t = (U_1^1, U_1^2, U_1^3, U_1^4) .$$

Choisir T revient en fait à choisir R^{-1} . Quelle forme très simple peut-on donner à $U_1^t R^{-1}$?

On a :

$$U_1^t R^{-1} = (U_1^1 R_{11} + U_1^3 R_{12} , U_1^2 R_{11} + U_1^4 R_{12} , U_1^1 R_{13} + U_1^3 R_{14} , U_1^2 R_{13} + U_1^4 R_{14}) .$$

Il nous faut distinguer dans la suite deux cas suivant que la matrice

$$\begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix} \text{ est régulière ou non.}$$

Cas 1

On suppose $\begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix}$ régulière. On peut alors choisir, de manière

unique, R_1, R_2, R_3, R_4 tels que la matrice R^{-1} soit régulière, et telle que :

$$U_1^t R^{-1} = (1, 0, 0, 1) .$$

En effet, pour cela, il faut résoudre les deux systèmes :

$$\begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix} \begin{pmatrix} R_3 \\ R_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} .$$

On aura donc la seule solution :

$$\begin{pmatrix} R_1 & R_3 \\ R_2 & R_4 \end{pmatrix} = \begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix}^{-1} .$$

En utilisant la transformation (R^{-1}, S^{-1}, I) correspondante on obtiendra donc :

$$(a) \quad F(X, Z, Y) = (X_1 + X_4)(W_1^t Z)(V_1^t S^{-1} Y) + \sum_{j=2}^q (U_j^t R^{-1} X)(W_j^t Z)(V_j^t S^{-1} Y) .$$

Avec :

$$\begin{aligned} F(X, Z, Y) &= (X_1 Z_1 + X_2 Z_3) Y_1 + (X_1 Z_2 + X_2 Z_4) Y_2 \\ &\quad + (X_3 Z_1 + X_4 Z_3) Y_3 + (X_3 Z_2 + X_4 Z_4) Y_4 . \end{aligned}$$

Si l'on fait maintenant $X_1 = -X_4$ dans (a) on voit que le rang tensoriel de la nouvelle forme trilinéaire est au plus égal à $q-1$. Cette forme trilinéaire est la suivante :

$$\begin{aligned} F'(X, Z, Y) &= (-X_4 Z_1 + X_2 Z_3) Y_1 + (-X_4 Z_2 + X_2 Z_4) Y_2 \\ &\quad + (X_3 Z_1 + X_4 Z_3) Y_3 + (X_3 Z_2 + X_4 Z_4) Y_4 . \end{aligned}$$

La matrice $B_1(Z)$ associée a la forme suivante :

$$B_1(Z) = \begin{pmatrix} Z_3 & Z_4 & 0 & 0 \\ 0 & 0 & Z_1 & Z_2 \\ -Z_1 & -Z_2 & Z_3 & Z_4 \end{pmatrix} .$$

On va montrer que le rang tensoriel de cette matrice est au moins égal à six (on aura donc $q-1 \geq 6$, c'est-à-dire $q \geq 7$). Pour démontrer ceci, on va appliquer le théorème 1 :

$$\text{Soit } B_1(Z) = \sum_{j=1}^{q'} (\lambda_j^t Z) U_j V_j^t \quad U_j \in K^3, V_j \in K^4.$$

On peut exprimer Z_3 et Z_4 en fonction linéaire de Z_1 et Z_2 en annulant deux des formes $\lambda_1^t Z, \lambda_2^t Z, \lambda_3^t Z, \lambda_4^t Z$ ($\lambda_1, \dots, \lambda_4$ sont supposés être linéairement indépendants).

On aura par exemple :

$$Z_3 = \alpha_3 Z_1 + \beta_3 Z_2 \quad \text{et} \quad Z_4 = \alpha_4 Z_1 + \beta_4 Z_2.$$

La matrice $B_1(Z')$ obtenue à partir de $B_1(Z)$ en remplaçant Z_3 et Z_4 par ces deux formes linéaires en Z_1 et Z_2 a un rang tensoriel au plus égal à $q'-2$. Cette matrice a la forme suivante :

$$B_1(Z') = \begin{pmatrix} \alpha_3 Z_1 + \beta_3 Z_2 & \alpha_4 Z_1 + \beta_4 Z_2 & 0 & 0 \\ 0 & 0 & Z_1 & Z_2 \\ -Z_1 & -Z_2 & \alpha_3 Z_1 + \beta_3 Z_2 & \alpha_4 Z_1 + \beta_4 Z_2 \end{pmatrix}.$$

Cette matrice a un rang tensoriel supérieur ou égal à quatre puisque ses quatre colonnes sont K linéairement indépendantes.

On a donc bien :

$$\text{Rt } B_1(Z') \geq 6,$$

et donc

$$\text{Rt } B(Z) \geq 7.$$

Cas 2

On suppose cette fois la matrice $\begin{pmatrix} U_1^1 & U_1^3 \\ U_1^2 & U_1^4 \end{pmatrix}$ singulière.

On peut donc prendre : $U_1^2 = \lambda U_1^1$, $U_1^4 = \lambda U_1^3$.

D'autre part, on peut toujours supposer le vecteur U_1 tel que l'on ait $U_1^1 \neq 0$ (un des vecteurs U_j , $j=1, \dots, q$ doit avoir U_j^1 non nul). Dans ce cas, on peut alors choisir R de telle manière que l'on ait :

$$U_1^t R^{-1} = (1, \lambda, 0, 0) \cdot$$

En effet, il faut pour cela résoudre les deux équations suivantes :

$$U_1^1 R_1 + U_1^3 R_2 = 1$$

$$U_1^1 R_3 + U_1^3 R_4 = 0$$

On peut donc prendre comme solution :

$$R_1 = (U_1^1)^{-1}(1 - U_1^3 R_2) \quad \text{et} \quad R_3 = -(U_1^1)^{-1} U_1^3 R_4.$$

La matrice R est bien régulière car on a :

$$\begin{aligned} R_1 R_4 - R_2 R_3 &= (U_1^1)^{-1}(1 - U_1^3 R_2) R_4 + (U_1^1)^{-1} U_1^3 R_2 R_4 \\ &= (U_1^1)^{-1} \end{aligned}$$

En utilisant la transformation (R^{-1}, S^{-1}, I) correspondant à ce choix on va donc obtenir cette fois :

$$F(X, Z, Y) = (X_1 + \lambda X_2)(W_1^t Z)(V_1^t S^{-1} Y) + \sum_{j=2}^q (U_j^t R^{-1} X)(W_j^t Z)(V_j^t S^{-1} Y).$$

En faisant $X_1 = -\lambda X_2$ dans cette forme trilinéaire F , on obtient une nouvelle forme F' de rang tensoriel au plus égal à $q-1$.

Il s'agit de montrer que le rang tensoriel de cette forme F' est au moins égal à six (on aura donc $q - 1 \geq 6$ soit $q \geq 7$).

La nouvelle forme trilinéaire F' est la suivante :

$$\begin{aligned} F'(X, Z, Y) &= (-\lambda X_2 Z_1 + X_2 Z_3) Y_1 + (-\lambda X_2 Z_2 + X_2 Z_4) Y_2 \\ &\quad + (X_3 Z_1 + X_4 Z_3) Y_3 + (X_3 Z_2 + X_4 Z_4) Y_4. \end{aligned}$$

La matrice $B''(Y)$ associée à cette forme trilinéaire est la suivante :

$$B''(Y) = \begin{pmatrix} -\lambda Y_1 & -\lambda Y_2 & Y_1 & Y_2 \\ Y_3 & Y_4 & 0 & 0 \\ 0 & 0 & Y_3 & Y_4 \end{pmatrix} .$$

On va montrer que le rang tensoriel de $B''(Y)$ est au moins égal à six. Pour démontrer ceci, on va appliquer le théorème 1.

$$\text{Soit } B''(Y) = \sum_{j=1}^{q'} (\lambda_j^t Y) U_j V_j^t \quad U_j \in K^3 \quad V_j \in K^4 \quad j=1, \dots, q' .$$

On peut exprimer Y_1 et Y_2 en fonction de Y_3 et Y_4 en annulant deux des formes $\lambda_1^t Y, \dots, \lambda_4^t Y$ ($\lambda_1, \dots, \lambda_4$ sont supposés être linéairement indépendants). La matrice obtenue à partir de $B''(Y)$ en remplaçant Y_1 et Y_2 par ces formes linéaires de Y_3 et Y_4 aura un rang tensoriel au plus égal à $q'-2$. Mais les quatre colonnes de cette matrice restent visiblement K linéairement indépendantes, son rang tensoriel est donc supérieur ou égal à quatre, et on a donc bien :

$$q'-2 \geq 4$$

$$\text{soit } q' \geq 6$$

$$\text{et donc } q \geq 7 \quad . \quad \square$$

3 . PRODUIT DE DEUX QUATERNIONS

Soit les éléments i, j, k vérifiant les relations suivantes :

$$i^2 = j^2 = k^2 = -1, \quad ij = -ji = k, \quad jk = -kj = i$$

$$\text{et } ki = -ik = j .$$

Si $x_n, y_n \in \mathbb{R}$ pour $n = 1, 2, 3, 4$, alors $X = x_1 + x_2 i + x_3 j + x_4 k$

et $Y = Y_1 + Y_2 i + Y_3 j + Y_4 k$ sont appelés quaternions (sur \mathbb{R}) .

et leur produit est le quaternion-XY :

$$\begin{aligned}
 XY = & (x_1y_1 - x_2y_2 - x_3y_3 - x_4y_4) \\
 & + (x_1y_2 + x_2y_1 + x_3y_4 - x_4y_3)i \\
 & + (x_1y_3 - x_2y_4 + x_3y_1 + x_4y_2)j \\
 & + (x_1y_4 + x_2y_3 - x_3y_2 + x_4y_1)k .
 \end{aligned} \tag{1}$$

Les quaternions réels satisfont $N(X) \equiv x_1^2 + x_2^2 + x_3^2 + x_4^2$

($N(X) > 0$ si $X \neq 0$) et $N(XY) = N(X)N(Y)$. Les quaternions sont utiles dans certains calculs mathématiques ou physiques.

Bien que les quaternions les plus souvent utilisés soient définis sur \mathbb{R} , le produit de deux quaternions peut être calculé quand les x_n et les y_n sont des éléments d'un anneau quelconque. On va étudier la complexité, au point de vue nombre de multiplications "générales" utilisées, du produit de deux quaternions arbitraires.

D'après les formules (1) il s'agit d'étudier la complexité de l'évaluation simultanée des quatre formes bilinéaires suivantes :

$$X^t B_1 Y = x_1y_1 - x_2y_2 - x_3y_3 - x_4y_4 ,$$

$$X^t B_2 Y = x_1y_2 + x_2y_1 + x_3y_4 - x_4y_3 ,$$

$$X^t B_3 Y = x_1y_3 - x_2y_4 + x_3y_1 + x_4y_2 ,$$

$$X^t B_4 Y = x_1y_4 + x_2y_3 - x_3y_2 + x_4y_1 .$$

Soit $Z^t = (Z_1, Z_2, Z_3, Z_4)$.

La matrice $B(Z) = \sum_{i=1}^4 Z_i B_i$ a la forme suivante :

$$B(Z) = \begin{pmatrix} Z_1 & Z_2 & Z_3 & Z_4 \\ Z_2 & -Z_1 & Z_4 & -Z_3 \\ Z_3 & -Z_4 & -Z_1 & Z_2 \\ Z_4 & Z_3 & -Z_2 & -Z_1 \end{pmatrix} .$$

Sous la forme (1) le calcul du produit de deux quaternions arbitraires nécessite seize multiplications générales. Nous allons tout d'abord montrer que ce calcul peut se faire en huit multiplications "générales" seulement.

Théorème 4.

/ Le produit de deux quaternions réels peut se faire avec huit multiplications générales. /

□ On va chercher une décomposition de la matrice $B(Z)$ en somme de huit matrices de rang un, ce qui nous donnera donc une formule de calcul du produit de deux quaternions en huit multiplications "générales".

On peut écrire :

$$B(Z) = B'(Z) + B''(Z) ,$$

avec :

$$B'(Z) = \begin{pmatrix} -Z_1 & Z_2 & Z_3 & Z_4 \\ Z_2 & -Z_1 & Z_4 & Z_3 \\ Z_3 & Z_4 & -Z_1 & Z_2 \\ Z_4 & Z_3 & Z_2 & -Z_1 \end{pmatrix} \quad \text{et} \quad B''(Z) = \begin{pmatrix} 2Z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2Z_3 \\ 0 & -2Z_4 & 0 & 0 \\ 0 & 0 & -2Z_2 & 0 \end{pmatrix} .$$

$B''(Z)$ peut se mettre, de manière évidente, sous la forme d'une somme de quatre matrices de rang un, il en est de même pour $B'(Z)$ car on peut écrire :

$$B'(Z) = \frac{1}{4}(-Z_1+Z_2+Z_3+Z_4) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} + \frac{1}{4}(-Z_1+Z_2-Z_3-Z_4) \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix} \\ + \frac{1}{4}(-Z_1-Z_2+Z_3-Z_4) \begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix} + \frac{1}{4}(-Z_1-Z_2-Z_3+Z_4) \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} .$$

Soit $Z = Z_1 + Z_2 i + Z_3 j + Z_4 k$ le quaternion produit de X par Y . Les formules de calcul de Z_1, \dots, Z_4 en huit multiplications sont les suivantes :

$$Z_1 = 2 \text{ [I]} - 1/4 (\text{[V]} + \text{[VI]} + \text{[VII]} + \text{[VIII]}) ,$$

$$Z_2 = -2 \text{ [II]} + 1/4 (\text{[V]} + \text{[VI]} - \text{[VII]} - \text{[VIII]}) ,$$

$$Z_3 = -2 \text{ [III]} + 1/4 (\text{[V]} - \text{[VI]} + \text{[VII]} - \text{[VIII]}) ,$$

$$Z_4 = -2 \text{ [IV]} + 1/4 (\text{[V]} - \text{[VI]} - \text{[VII]} + \text{[VIII]}) .$$

Avec

$$\text{[I]} = x_1 y_1 , \quad \text{[II]} = x_4 y_3 , \quad \text{[III]} = x_2 y_4 , \quad \text{[IV]} = x_3 y_2 ,$$

$$\text{[V]} = (x_1 + x_2 + x_3 + x_4)(y_1 + y_2 + y_3 + y_4) ,$$

$$\text{[VI]} = (x_1 + x_2 - x_3 - x_4)(y_1 + y_2 - y_3 - y_4) ,$$

$$\text{[VII]} = (x_1 - x_2 + x_3 - x_4)(y_1 - y_2 + y_3 - y_4) ,$$

$$\text{[VIII]} = (x_1 - x_2 - x_3 + x_4)(y_1 - y_2 - y_3 + y_4) . \quad \square$$

Nous allons maintenant prouver l'optimalité de ce résultat quand les x_i et les y_i appartiennent à un anneau non commutatif et que le corps K des scalaires utilisés possède la propriété suivante :

$$a^2 + b^2 + c^2 + d^2 = 0 \text{ si et seulement si } a = b = c = d = 0 \text{ (propriété } P^* \text{)} .$$

Le corps des réels et celui des rationnels vérifient cette propriété.

Par contre le corps des complexes et tous les corps finis ne la vérifient pas.

Théorème 5

/ Tout algorithme de calcul du produit de deux quaternions sur un anneau non commutatif et qui n'utilise que des scalaires d'un corps K possédant la propriété P^* , doit utiliser au minimum huit multiplications générales. (Les formules précédentes sont donc optimales). /

□ Comme on est dans le cas où l'on n'utilise pas la commutativité, on sait que le nombre minimum de multiplications "générales" nécessaires pour calculer le produit de deux quaternions est égal au rang tensoriel de la forme trilinéaire F associée à ce calcul.

On a :

$$F(X,Z,Y) = X^t B(Z) Y, \text{ avec } B(Z) = \begin{pmatrix} Z_1 & Z_2 & Z_3 & Z_4 \\ Z_2 & -Z_1 & Z_4 & -Z_3 \\ Z_3 & -Z_4 & -Z_1 & Z_2 \\ Z_4 & Z_3 & -Z_2 & -Z_1 \end{pmatrix} .$$

Soit q le rang tensoriel de $F(X,Z,Y)$, on peut écrire :

$$F(X,Z,Y) = \sum_{j=1}^q (U_j^t X)(W_j^t Z)(V_j^t Y) .$$

On a nécessairement $q \geq 4$.

Pour montrer que l'on a toujours $q \geq 8$, on va utiliser tout d'abord des transformations qui laissent F invariante pour se ramener au cas où l'on a :

$$F(X,Z,Y) = x_1 (W_1^t Z)(V_1^t Y) + (U_2^t X)(W_2^t Z)y_1 + \sum_{k=3}^q (U_k^t X)(W_k^t Z)(V_k^t Y) .$$

Considérons les deux matrices orthogonales suivantes :

$$A = \frac{1}{d(a)} \begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & -a_4 & a_3 \\ a_3 & a_4 & a_1 & -a_2 \\ a_4 & -a_3 & a_2 & a_1 \end{pmatrix} ; \quad B = \frac{1}{d(b)} \begin{pmatrix} b_1 & -b_2 & -b_3 & -b_4 \\ b_2 & b_1 & b_4 & -b_3 \\ b_3 & -b_4 & b_1 & b_2 \\ b_4 & b_3 & -b_2 & b_1 \end{pmatrix} ,$$

$$d(a) = a_1^2 + a_2^2 + a_3^2 + a_4^2 \quad ; \quad d(b) = b_1^2 + b_2^2 + b_3^2 + b_4^2 \quad (a_i, b_i \in K) .$$

On suppose évidemment $d(a) \neq 0$ et $d(b) \neq 0$, ce qui revient à supposer, puisque K doit posséder la propriété P^* , que les quatre scalaires a_i ($i=1, \dots, 4$) resp. $(b_i, i=1, \dots, 4)$ ne sont pas tous nuls.

Un calcul direct montre que $T_A = (A, d(a)A, I)$ et $T_B = (I, d(B)B, B)$ sont deux transformations qui laissent invariante la forme F .

Appliquons tout d'abord à F la transformation T_A .

On obtient :

$$F(X, Z, Y) = d(a) \sum_{j=1}^q (U_j^t A X) (W_j^t A Z) (V_j^t Y) .$$

Soit $U_1^t = (U_1^1 U_1^2 U_1^3 U_1^4)$. Le vecteur U_1 n'est pas le vecteur nul puisque q est supposé être le rang tensoriel de la forme F . Choisissons alors la matrice A dont les éléments sont justement les composantes de U_1 :

$$(a_1 a_2 a_3 a_4) = (U_1^1 U_1^2 U_1^3 U_1^4) .$$

On aura bien : $d(a) \neq 0$.

Avec ce choix de la matrice A , il est immédiat de vérifier que l'on a :

$$U_1^t A X = x_1 .$$

On a donc :

$$F(X, Z, Y) = x_1 (W_1^t Z) (V_1^t Y) + \sum_{j=2}^q (U_j^t X) (W_j^t Z) (V_j^t Y) .$$

Appliquons maintenant la transformation T_B à F écrite comme ci-dessus.

On obtient :

$$F(X, Z, Y) = d(b) [x_1 (W_1^t B Z) (V_1^t B Y) + \sum_{j=2}^q (U_j^t X) (W_j^t B Z) (V_j^t B Y)]$$

Le vecteur V_2 ne peut être le vecteur nul (sinon le rang tensoriel de F serait inférieur à q), on peut donc choisir la matrice B avec les éléments b_i ($i=1, \dots, 4$) tels que :

$$(b_1 b_2 b_3 b_4) = (V_2^1 V_2^2 V_2^3 V_2^4) .$$

Avec ce choix de B il est facile de constater que l'on a :

$$V_2^t B Y = y_1 .$$

On obtient donc :

$$F(X,Z,Y) = x_1(W_1^t Z)(V_1^t Y) + (U_2^t X)(W_2^t Z)y_1 + \sum_{j=3}^q (U_j^t X)(W_j^t Z)(V_j^t Y) .$$

Si dans l'expression de $F(X,Y,Z)$ on remplace x_1 et y_1 par zéro, on va donc obtenir une nouvelle forme F' dont le rang tensoriel sera au plus égal à $q-2$.

Cette forme trilinéaire F' est la suivante :

$$F'(X,Z,Y) = (-x_2 y_2 - x_3 y_3 - x_4 y_4)z_1 + (x_3 y_4 - x_4 y_3)z_2 \\ + (-x_2 y_4 + x_4 y_2)z_3 + (x_2 y_3 - x_3 y_2)z_4 .$$

Cette forme trilinéaire est associée au calcul du produit de $(x_2 i + x_3 j + x_4 k)$ par $(y_2 i + y_3 j + y_4 k)$.

Si l'on peut montrer que le rang tensoriel de cette forme F' est au moins égal à six on en déduira alors :

$$q-2 \geq 6 \quad \text{c'est-à-dire} \quad q \geq 8 .$$

Tout revient donc à démontrer maintenant que le rang tensoriel de la matrice $B'(Z)$ suivante est au moins six :

$$B'(Z) = \begin{pmatrix} -Z_1 & Z_4 & -Z_3 \\ -Z_4 & -Z_1 & Z_2 \\ Z_3 & -Z_2 & -Z_1 \end{pmatrix} .$$

Soit q' le rang tensoriel de $B'(Z)$ ($q' \geq 4$).

$$B'(Z) = \sum_{j=1}^{q'} (W_j^t Z) U_j V_j^t \quad U_j \in K^3, V_j \in K^3, W_j \in K^4 .$$

On peut toujours supposer W_1, W_2, W_3, W_4 linéairement indépendants . Il existe une solution $Z^0 = (Z_1^0, Z_2^0, Z_3^0, Z_4^0)$ avec $Z_1^0 \neq 0$ au système de trois équations à quatre inconnues z_i ($i=1, \dots, 4$):

$$W_i^t Z = 0 \quad (i=1, 2, 3) .$$

(On peut toujours supposer que W_1, W_2, W_3 sont tels que la matrice 3×3 constituée par les trois dernières composantes de ces vecteurs est régulière, si bien que l'on peut résoudre le système en exprimant Z_2, Z_3 et Z_4 en fonction (linéaire) de Z_1). Pour ce choix de Z , on va obtenir :

$$B'(Z^0) = \sum_{j=4}^q (W_j^t Z_0) U_j V_j^t$$

Mais le déterminant de $B'(Z^0)$ vaut : $-Z_1^0(Z_1^{0^2} + Z_2^{0^2} + Z_3^{0^2} + Z_4^{0^2})$.

Puisque $Z_1^0 \neq 0$, et comme K possède la propriété P^* , ce déterminant est non nul. Par conséquent, la matrice $B'(Z_0)$ est de rang trois.

On a donc :

$$q' - 3 \geq 3,$$

c'est-à-dire

$$q' \geq 6.$$

On a donc bien finalement $q \geq 8$. \square

Remarque 1 :

Nous venons de démontrer dans le théorème précédent que le rang tensoriel de la forme trilinéaire $F'(X, Z, Y)$ était au minimum égal à six. L'expression de cette forme F' est la suivante :

$$\begin{aligned} F'(X, Z, Y) = & (-x_2 y_2 - x_3 y_3 - x_4 y_4) Z_1 + (x_3 y_4 - x_4 y_3) Z_2 \\ & + (-x_2 y_4 + x_4 y_2) Z_3 + (x_2 y_3 - x_3 y_2) Z_4. \end{aligned}$$

Posons $V^t = (x_2, x_3, x_4)$, $W^t = (y_2, y_3, y_4)$.

La forme trilinéaire $F'(\lambda, Y, Z)$ peut être considérée comme associée au calcul simultané du produit scalaire $V \cdot W$ de ces deux vecteurs, et de leur produit vectoriel $V \wedge W$.

On peut donc énoncer le résultat suivant :

Corollaire 2.

Le calcul simultané du produit scalaire et du produit vectoriel de deux vecteurs quelconques de R^3 nécessite au moins six multiplications.

Remarque 2

Dans la démonstration du théorème précédent, on a utilisé deux types de transformations qui laissent $F(X,Z,Y)$ invariante, à savoir les transformations T_A et T_B

$$T_A = (A, d(a)A, I) ,$$

$$T_B = (I, d(B)B, B) .$$

Il existe d'autres transformations qui laissent invariante cette forme trilineaire (l'ensemble de toutes ces transformations forme un groupe pour la composition).

Exemple

$$\text{Soit } A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix} , \quad B = \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix} , \quad C = 1/4 AB$$

$T = (A, C, B)$ est une transformation qui laisse invariante $F(X, Z, Y)$ et qui n'est pas du type T_A ou T_B .

Si on applique cette transformation T à la décomposition en huit matrices de rang un précédemment trouvée, on obtient les nouvelles formules suivantes (pour le calcul des quatre composantes z_i ($i=1, \dots, 4$) du quaternion XY) :

$$Z_1 = [II] + 1/2(-[V] - [VI] + [VII] + [VIII]) ,$$

$$Z_2 = [I] - 1/2 ([IV] + [VI] + [VII] + [VIII]) ,$$

$$Z_3 = -[III] + 1/2 ([IV] - [VI] + [VII] - [VIII]) ,$$

$$Z_4 = -[IV] + 1/2 ([V] - [VI] - [VII] + [VIII]) .$$

Avec :

$$\begin{aligned}
[\text{I}] &= (x_1+x_2)(y_1+y_2) \quad , & [\text{V}] &= (x_2+x_4)(y_2+y_3) \quad , \\
[\text{II}] &= (x_4-x_3)(y_3-y_4) \quad , & [\text{VI}] &= (x_2-x_4)(y_2-y_3) \quad , \\
[\text{III}] &= (x_2-x_1)(y_3-y_4) \quad , & [\text{VII}] &= (x_1+x_3)(y_1-y_4) \quad , \\
[\text{IV}] &= (x_3+x_4)(y_2-y_1) \quad , & [\text{VIII}] &= (x_1-x_3)(y_1+y_4) \quad .
\end{aligned}$$

Ces formules, ainsi que les précédentes sont optimales quant aux nombre de multiplications générales utilisées, quand on n'utilise pas la commutativité (entre les x_i et les y_j).

On a maintenant montrer que le nombre minimal de multiplications générales nécessaires pour calculer le produit de deux quaternions, en utilisant la commutativité est supérieur ou égal à sept.

Théorème 6

/ Sept multiplications au moins sont nécessaires pour calculer le produit de deux quaternions en utilisant la commutativité quand le corps K des scalaires possède la propriété P_* . /

□ On suppose donc $K[x_1, \dots, x_4, y_1, \dots, y_4]$ commutatif.

Dans ce cas, on sait que le nombre minimum q de multiplications "générales" nécessaires pour calculer le produit de deux quaternions est égal au minimum du rang tensoriel des matrices (8×8) $N(Z)$ qui vérifient :

$$N(Z) + N(Z)^t = \begin{pmatrix} 0 & B(Z) \\ B(Z)^t & 0 \end{pmatrix} ,$$

$$\text{avec } B(Z) = \begin{pmatrix} Z_1 & Z_2 & Z_3 & Z_4 \\ Z_2 & -Z_1 & Z_4 & -Z_3 \\ Z_3 & -Z_4 & -Z_1 & Z_2 \\ Z_4 & Z_3 & -Z_2 & -Z_1 \end{pmatrix} .$$

On va montrer que le rang tensoriel de ces matrices $N(Z)$ est toujours minoré par sept.

Soit q le rang tensoriel d'une telle matrice $N(Z)$.

Alors, on peut écrire :

$$N(Z) = \sum_{j=1}^q (W_j^t Z) U_j V_j^t \quad U_j \in K^8, V_j \in K^8, W_j \in K^4, j=1, \dots, q.$$

On peut toujours supposer que W_1, W_2, W_3, W_4 sont linéairement indépendants et que les trois dernières composantes de W_1, W_2 et W_3 forment une matrice (3×3) régulière.

Considérons le système d'équations suivant :

$$W_1^t Z = 0, \quad W_2^t W = 0, \quad W_3^t Z = 0.$$

On peut trouver $Z^0 \neq 0$ ($Z^0 = (Z_1^0, Z_2^0, Z_3^0, Z_4^0)$) avec $Z_1^0 \neq 0$ qui soit une solution de ce système.

Pour ce choix de Z , on a :

$$N(Z^0) = \sum_{j=4}^q (W_j^t Z^0) U_j V_j^t.$$

$N(Z^0)$ est exprimée comme une combinaison linéaire de $q-3$ matrices de rang un, donc :

$$\text{Rang}(N(Z^0)) \leq q-3.$$

Mais nous avons aussi :

$$N(Z^0) + N(Z^0)^t = \begin{pmatrix} 0 & B(Z^0) \\ B(Z^0)^t & 0 \end{pmatrix}.$$

Le déterminant de $B(Z^0)$ vaut :

$$-((Z_1^0)^2 + (Z_2^0)^2 + (Z_3^0)^2 + (Z_4^0)^2)^2.$$

Comme K possède la propriété P^* il ne peut s'annuler car Z_1^0 est non nul. La matrice $N(Z^0) + N(Z^0)^t$ est donc régulière et donc :

$$\text{Rang}(N(Z^0) + N(Z^0)^t) = 8.$$

Mais :

$$\text{Rang} (N(Z^0) + N(Z^0)^t) \leq 2 \text{Rang} (N(Z^0)) \leq 2 (q - 3) ,$$

donc $8 \leq 2 (q-3) ,$

ce qui entraîne $q \geq 7 . \square$

Remarque 3

Les démonstrations des bornes inférieures données dans le cas non-commutatif et dans le cas commutatif, dépendent de l'hypothèse que K possède la propriété P^* , c'est-à-dire que l'équation $a^2 + b^2 + c^2 + d^2 = 0$ n'a pas de solution non triviale dans K . Cette hypothèse est vérifiée si le corps K est le corps des rationnels ou celui des réels, par contre elle ne l'est plus si K est le corps des complexes C . L'exemple suivant va montrer que le théorème 5 n'est plus vrai dans ce cas.

Au quaternion $X = x_1 + x_2 i + x_3 j + x_4 k$ on peut associer la matrice 2×2 sur C suivante :

$$\begin{pmatrix} X_1 & X_2 \\ -\bar{X}_2 & \bar{X}_1 \end{pmatrix} \quad \begin{array}{l} X_1 = x_1 + x_2 i , \\ X_2 = x_3 + x_4 i . \end{array}$$

Cette correspondance est un isomorphisme entre l'ensemble des quaternions et l'ensemble des matrices de ce type.

Le produit de deux quaternions correspond au produit des deux matrices associées . D'après le résultat de STRASSEN ce produit peut être calculé en sept multiplications complexes (en fait six complexes et une réelle).

La même construction donne un algorithme pour multiplier deux quaternions sur un anneau quelconque avec sept multiplications "complexes", où une multiplication "complexe" signifie la construction de $(x_1 y_1 - x_2 y_2) + (x_1 y_2 + x_2 y_1) i$ à partir de $x_1 + x_2 i$ et $y_1 + y_2 i$ (x_1, x_2, y_1, y_2) étant des éléments de l'anneau.

Pour ce problème il reste donc soit à améliorer la borne inférieure sept dans le cas commutatif, soit à trouver une formule utilisant la commutativité et qui calcule le produit de deux quaternions en sept multiplications. Enfin, le cas où le corps K ne vérifie pas la propriété P^* mérite une étude spéciale.

Nous terminons cette étude de la complexité du produit de deux quaternions par le résultat suivant dû à DE GROOTE (33) et qui est à l'origine de notre intérêt pour cette question.

Théorème 7

/ Le calcul simultané des deux produits XY et YX des deux quaternions X et Y en dix multiplications est optimal si K possède la propriété P^* . /

□ On suppose évidemment ici $K[x_1, \dots, x_4, y_1, \dots, y_4]$ commutatif. Au lieu de calculer XY et YX on peut calculer $1/2(XY+YX)$ et $1/2(XY-YX)$. Le calcul nécessite l'évaluation de sept formes bilinéaires représentées par la matrice $B(Z)$ ($Z^t = (z_1, \dots, z_7)$) suivante :

$$B(Z) = \begin{pmatrix} z_1 & z_2 & z_3 & z_4 \\ z_2 & -z_1 & z_5 & -z_6 \\ z_3 & -z_5 & -z_1 & z_7 \\ z_4 & z_6 & -z_7 & -z_1 \end{pmatrix} .$$

Cette matrice a bien un rang tensoriel au plus égal à dix car on peut l'écrire sous la forme :

$$B(Z) = \begin{pmatrix} z_2 & z_2 & 0 & 0 \\ z_2 & z_2 & 0 & 0 \\ 0 & 0 & z_7 & z_7 \\ 0 & 0 & -z_7 & -z_7 \end{pmatrix} + \begin{pmatrix} z_3 & 0 & z_3 & 0 \\ 0 & -z_6 & 0 & -z_6 \\ z_3 & 0 & z_3 & 0 \\ 0 & z_6 & 0 & z_6 \end{pmatrix} + \begin{pmatrix} z_4 & 0 & 0 & z_4 \\ 0 & z_5 & z_5 & 0 \\ 0 & -z_5 & -z_5 & 0 \\ z_4 & 0 & 0 & z_4 \end{pmatrix} + D$$

4 . PRODUIT VECTORIEL DE DEUX VECTEURS ,
PRODUIT DE LIE DE DEUX MATRICES 2×2 .

Soit U et V deux vecteurs de \mathbb{R}^3 .

$$U^t = (x_1, x_2, x_3) \quad , \quad V^t = (y_1, y_2, y_3)$$

Le produit vectoriel de U et de V est le vecteur W de composantes z_1, z_2 , et z_3 données par les formules :

$$z_1 = x_2 y_3 - x_3 y_2 \quad ,$$

$$z_2 = x_3 y_1 - x_1 y_3 \quad ,$$

$$z_3 = x_1 y_2 - x_2 y_1 \quad .$$

Ces formules permettent de calculer le produit vectoriel de deux vecteurs quelconques de \mathbb{R}^3 en six multiplications.

La matrice $B(Z)$ associée au calcul de ces trois formes bilinéaires est la suivante :

$$B(Z) = \begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix} .$$

On va supposer tout d'abord $K[x_1, x_2, x_3, y_1, y_2, y_3]$ non-commutatif.

Dans ce cas le nombre minimum de multiplications générales nécessaires pour évaluer $U \wedge V$ est égal au rang tensoriel de la matrice $B(Z)$. Nous allons montrer que ce rang tensoriel est exactement égal à cinq.

Théorème 7

/ Le produit vectoriel de deux vecteurs de \mathbb{R}^3 en cinq multiplications est optimal . /

□ Il faut tout d'abord montrer qu'il existe une méthode de calcul de $U \wedge V$ en cinq multiplications. Cela revient à chercher une décomposition de la matrice $B(Z)$ en cinq matrices de rang un. Ceci est possible comme le montre l'expression suivante de $B(Z)$.

$$\begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & z_3 & z_3 \\ 0 & z_3 & z_3 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ -z_1 & -z_1 & 0 \\ -z_1 & -z_1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ z_1 - z_3 & z_1 - z_3 & z_1 - z_3 \\ 0 & 0 & 0 \end{pmatrix} \\ + \begin{pmatrix} 0 & 0 & -z_2 - z_3 \\ 0 & 0 & 0 \\ z_1 + z_2 & 0 & 0 \end{pmatrix} .$$

Cette décomposition de $B(Z)$ en cinq matrices de rang un conduit aux formules suivantes de calcul des composantes z_1, z_2, z_3 de $U \wedge V$:

$$z_1 = -(x_2 + x_3)(y_1 + y_2) + x_2(y_1 + y_2 + y_3) + x_3 y_1,$$

$$z_2 = x_3 y_1 - x_1 y_3$$

$$z_3 = (x_1 + x_2)(y_2 + y_3) - x_2(y_1 + y_2 + y_3) - x_1 y_3 .$$

Pour démontrer l'optimalité de cette formule, il suffit maintenant de prouver que le rang tensoriel de $B(Z)$ (c'est-à-dire aussi celui de $F(X, Y, Z) = X^t B(Z) Y$) est au moins égal à cinq. Pour faire cela, on va tout d'abord se servir des transformations qui laissent invariante $F(X, Y, Z)$.

On a :

$$F(X, Y, Z) = -(-x_3 z_2 + x_2 z_3) y_1 + (x_1 z_2 - x_2 z_1) y_2 + (-x_1 z_2 + x_2 z_1) y_3$$

Soit P une matrice régulière de $\mathcal{M}_{3,3}(\mathbb{R})$.

Un calcul direct montre que l'on a :

$$F(PX', PY', |P| P^{-1} Z') = -(-x'_3 z'_2 + x'_2 z'_3) y'_1 + (x'_1 z'_2 - x'_2 z'_1) y'_2 + (-x'_1 z'_2 + x'_2 z'_1) y'_3 .$$

Par conséquent $T = (P, P, |P| P^{-1})$ est une transformation qui laisse F invariante.

Si q est le rang tensoriel de F on peut écrire :

$$F(X,Y,Z) = \sum_{j=1}^q (U_j^t X)(V_j^t Y)(W_j^t Z) .$$

Il existe forcément trois vecteurs linéairement indépendants parmi les q vecteurs U_j (ou W_j).

On peut toujours supposer que U_1 et U_2 sont **linéairement** indépendants et que W_3 est linéairement indépendant de U_1 et U_2 : choisissons donc pour matrice P , celle déterminée par :

$$\begin{aligned} P^t U_1 &= e_1, \quad P^t U_2 = e_3, \quad P^t W_3 = e_2 \\ (P^t &= (U_1 W_3 V_2)^{-1} \text{ est bien régulière}). \end{aligned}$$

Si on applique la transformation $T = (P, P, |P| P^{-1})$ à $F(X,Y,Z)$ on obtient pour ce choix de la matrice P :

$$\begin{aligned} F(PX', PY', |P| P^{-1}Z') &= \sum_{j=1}^q (U_j^t PX')(W_j^t |P| P^{-1}Z')(V_j^t PY') \\ &= x'_1 (W_1^t Z')(V_1^t Y') + x'_3 (W_2^t Z')(V_2^t Y') + (U_3^t X)(W_3^t Z') y'_2 \\ &\quad + \sum_{j=4}^q (U_j^t X')(W_j^t Z')(V_j^t Y'). \end{aligned}$$

Si on fait maintenant $x'_1 = x'_3 = y'_2 = 0$ on obtient la nouvelle forme trilineaire suivante dont le rang tensoriel est au plus égal à $q-3$:

$$F'(X', Y', Z') = x'_2 z'_3 y'_1 + x'_2 z'_1 y'_3 .$$

Or cette nouvelle forme est de rang tensoriel égal à deux.

On en déduit donc :

$$q-3 \geq 2$$

Soit $q \geq 5$. \square

Théorème 7'

/ Il faut au moins quatre multiplications pour faire le produit vectoriel de deux vecteurs de R^3 quand on utilise la commutativité. /

□ On suppose ici $K[x_1, x_2, x_3, y_1, y_2, y_3]$ commutatif. Alors le nombre minimal de multiplications nécessaires pour évaluer le produit vectoriel est égal au plus petit rang tensoriel des matrices $N(Z)$ (6×6) telles que :

$$N(Z) + N(Z)^t = \begin{pmatrix} 0 & B(Z) \\ B(Z)^t & 0 \end{pmatrix} .$$

Soit q le rang tensoriel d'une telle matrice $N(Z)$, et écrivons $N(Z)$ en fonction de q matrices de rang un :

$$N(Z) = \sum_{j=1}^q (W_j^t Z) U_j V_j^t$$

On peut toujours supposer W_1, W_2, W_3 linéairement indépendants. On peut déterminer Z_0 (avec $Z_1^0 \neq 0$ par exemple) tels que :

$$W_1^t Z_0 = W_2^t Z_0 = 0 .$$

Le rang de la matrice $N(Z^0)$ sera au plus égal à $q-2$.

On aura d'autre part :

$$N(Z^0) + N(Z^0)^t = \begin{pmatrix} 0 & B(Z^0) \\ B(Z^0)^t & 0 \end{pmatrix} .$$

Or $B(Z^0)$ est une matrice de rang deux si $Z_1^0 \neq 0$, par conséquent $N(Z^0) + N(Z^0)^t$ est une matrice de rang 4. Le rang de $N(Z^0)$ est donc au moins égal à 2 et comme il est au plus égal à $q-2$ on aura :

$$q-2 \geq 2$$

$$\text{soit } q \geq 4 . \quad \square$$

Théorème 8

/ Le calcul simultané du produit scalaire et du produit vectoriel de deux vecteurs de R^3 en ~~en~~^{six} ~~en~~ multiplications est optimal quand on n'utilise pas la commutativité. /

Il faut calculer simultanément les quatre formes bilinéaires suivantes :

$$\begin{aligned} f_1 &= x_1 y_1 + x_2 y_2 + x_3 y_3 \quad , \\ f_2 &= x_2 y_3 - x_3 y_2 \quad , \\ f_3 &= x_3 y_1 - x_1 y_3 \quad , \\ f_4 &= x_1 y_2 - x_2 y_1 \quad . \end{aligned}$$

La matrice $B(Z)$ associée à ce calcul est la suivante :

$$B(Z) = \begin{pmatrix} z_1 & z_4 & -z_3 \\ -z_4 & z_1 & z_2 \\ z_3 & -z_2 & z_1 \end{pmatrix} .$$

Dans la démonstration du théorème 5, nous avons montré que le rang tensoriel de cette matrice était au moins égal à six. Il nous suffit donc maintenant de montrer qu'il existe une décomposition de cette matrice en six matrices de rang un : ce type de matrice a déjà été étudié dans le chapitre IV (paragraphe 4).

En utilisant la décomposition mentionnée alors, on obtient :

$$\begin{aligned} \begin{pmatrix} z_1 & z_4 & -z_3 \\ -z_4 & z_1 & z_2 \\ z_3 & -z_2 & z_1 \end{pmatrix} &= \begin{pmatrix} z_4 & z_4 & 0 \\ -z_4 & -z_4 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} -z_3 & 0 & -z_3 \\ 0 & 0 & 0 \\ z_3 & 0 & z_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & z_2 & z_2 \\ 0 & -z_2 & -z_2 \end{pmatrix} \\ &+ \begin{pmatrix} z_1 - z_4 + z_3 & 0 & 0 \\ 0 & z_1 + z_4 - z_2 & 0 \\ 0 & 0 & z_1 - z_3 + z_2 \end{pmatrix} . \end{aligned}$$

Les formules correspondantes, qui sont donc optimum puisqu'elles n'utilisent que six multiplications différentes sont les suivantes :

$$\begin{aligned}
f_1 &= x_1y_1 + x_2y_2 + x_3y_3 \quad , \\
f_2 &= (x_2 - x_3)(y_2 + y_3) - x_2y_2 - x_3y_3 \quad , \\
f_3 &= (x_3 - x_1)(y_1 + y_3) + x_1y_1 - x_3y_3 \quad , \\
f_4 &= (x_1 - x_2)(y_1 + y_2) - x_1y_1 + x_2y_2 \quad . \quad \square
\end{aligned}$$

Théorème 9

/ Le calcul du produit de Lie $AB - BA$ de deux matrices A et B de taille 2 peut se faire en cinq multiplications seulement. /

□ Soit A et B deux matrices quelconques de $\mathcal{M}_{2,2}(K)$.
Posons :

$$A = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} \quad , \quad B = \begin{pmatrix} y_1 & y_2 \\ y_3 & y_4 \end{pmatrix} .$$

On suppose évidemment ici la commutativité des x_i et des y_j .

On a donc :

$$AB - BA = \begin{pmatrix} x_2y_3 - x_3y_2 & x_1y_2 + x_2y_4 - x_2y_1 - x_4y_2 \\ -x_1y_3 + x_3y_1 - x_3y_4 + x_4y_3 & x_3y_2 - x_2y_3 \end{pmatrix}$$

Le calcul de $AB - BA$ revient au calcul simultané de trois formes bilinéaires dont la matrice $B(Z)$ associée est la matrice suivante :

$$B(Z) = \begin{pmatrix} 0 & z_3 & -z_2 & 0 \\ -z_3 & 0 & z_1 & z_3 \\ z_2 & -z_1 & 0 & -z_2 \\ 0 & -z_3 & z_2 & 0 \end{pmatrix} .$$

La dernière ligne et la dernière colonne de cette matrice sont proportionnelles respectivement à la première ligne et à la première colonne.

Le rang tensoriel de $B(Z)$ est donc égal à celui de sa sous-matrice suivante :

$$\begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix} .$$

On vient de voir que le rang tensoriel de cette matrice était égal à cinq et donc le rang tensoriel de $B(Z)$ est aussi égal à cinq. On en déduit donc la décomposition suivante de $B(Z)$:

$$\begin{pmatrix} 0 & z_3 & -z_2 & 0 \\ -z_3 & 0 & z_1 & z_3 \\ z_2 & -z_1 & 0 & -z_2 \\ 0 & -z_3 & z_2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & z_3 & z_3 & 0 \\ 0 & z_3 & z_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -z_3 & -z_3 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ -z_1 & -z_1 & 0 & z_1 \\ -z_1 & -z_1 & 0 & z_1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ + \begin{pmatrix} 0 & 0 & 0 & 0 \\ z_1 - z_3 & z_1 - z_3 & z_1 - z_3 & z_1 - z_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & -z_2 - z_3 & 0 \\ 0 & 0 & 0 & 0 \\ z_1 + z_2 & 0 & 0 & -z_1 - z_2 \\ 0 & 0 & z_2 + z_3 & 0 \end{pmatrix} .$$

On en déduit donc les formules de calculs suivantes :

$$\begin{aligned} x_2 y_3 - x_3 y_2 &= [\text{II}] + [\text{III}] + [\text{IV}] , \\ -x_1 y_3 + x_3 y_1 - x_3 y_4 + x_4 y_3 &= [\text{IV}] + [\text{V}] , \\ x_1 y_2 + x_2 y_4 - x_2 y_1 - x_4 y_2 &= [\text{I}] - [\text{III}] + [\text{V}] , \end{aligned}$$

où [I] , [II] , ..., [V] désignent les cinq multiplications à effectuer, à savoir :

$$\begin{aligned} [\text{I}] &= (x_1 + x_2 - x_4)(y_2 + y_3) , \\ [\text{II}] &= (x_2 + x_3)(-y_1 - y_2 + y_4) , \\ [\text{III}] &= x_2(y_1 + y_2 + y_3 - y_4) , \\ [\text{IV}] &= x_3(y_1 - y_3) , \\ [\text{V}] &= (-x_1 + x_4)y_3 . \end{aligned}$$

L'optimalité de cette formule n'est pas assurée, car comme on a supposé ici la commutativité (pour le calcul des éléments de $AB - BA$) on peut simplement affirmer que le nombre minimal de multiplications nécessaires pour faire ce calcul est au moins égal à quatre (cf. théorème 7'). \square

5 . RANG TENSORIEL DES MATRICES ANTI-SYMETRIQUES.

Dans le chapitre 4, on a montré (dans le paragraphe consacré aux matrices ayant des symétries particulières) que le sous-espace de $\mathcal{M}_{n,n}(K)$ constitué par les matrices anti-symétriques avait un rang tensoriel majoré par $\frac{n(n+1)}{2}$ et minoré par $\frac{n(n-1)}{2}$. Dans le cas particulier $n=3$, on a montré que le rang tensoriel était égal à cinq. Nous pouvons généraliser ce résultat de la manière suivante :

Théorème 10

- / Le rang tensoriel du sous-espace des matrices anti-symétriques de $\mathcal{M}_{n,n}(K)$ est exactement égal à $\frac{n(n+1)}{2} - 1$. /
- \square Le rang tensoriel de l'ensemble des matrices anti-symétriques de $\mathcal{M}_{n,n}(K)$ est égal au rang tensoriel de la matrice $B(Z)$ suivante :

$$B(Z) = \begin{pmatrix} 0 & z_{12} & z_{13} & \cdots & z_{1n} \\ -z_{12} & 0 & z_{23} & \cdots & z_{2n} \\ -z_{13} & -z_{23} & 0 & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ & & & & z_{n-1,n} \\ -z_{1n} & -z_{2n} & \cdots & -z_{n-1,n} & \cdot \end{pmatrix} .$$

On pose comme d'habitude :

$$\begin{aligned} X^t &= (X_1, \dots, X_n) , \\ Y^t &= (Y_1, \dots, Y_n) , \\ Z^t &= (Z_{12}, Z_{13}, \dots, Z_{n-1,n}) . \end{aligned}$$

Le rang tensoriel de $B(Z)$ est égal au rang tensoriel de la forme trilinéaire :

$$F(X,Y,Z) = X^t B(Z) Y .$$

Pour démontrer le résultat énoncé, il faut d'abord montrer que ce rang tensoriel est bien minoré pour $\frac{n(n+1)}{2} - 1$, ensuite il faudra donner une décomposition de $B(Z)$ sous forme d'une combinaison linéaire de $\frac{n(n+1)}{2} - 1$ matrices de rang un.

a/ Minoration du rang tensoriel de $B(Z)$.

On va utiliser les transformations qui laissent invariante la forme $F(X,Y,Z) = X^t B(Z) Y$.

Soit P une matrice régulière quelconque de $\mathcal{M}_{n,n}(K)$, alors la matrice $P^t B(Z) P$ est encore anti-symétrique (si $B(Z)^t = -B(Z)$ on a bien $P^t B(Z)^t P = -P^t B(Z) P$).

On peut donc écrire :

$$B(Z') = P^t B(Z) P$$

avec $Z' = Q Z$ $Q \in \mathcal{M}_{\frac{n(n-1)}{2}, \frac{n(n-1)}{2}}(K)$.

Il est facile de voir comment la matrice Q est construite à partir de la matrice P :

La première ligne de Q est construite à partir de la première et de la deuxième ligne de P et tous ses éléments sont les déterminants des matrices d'ordre deux obtenues en prenant successivement l'intersection de ces deux lignes avec la colonne 1 et 2, 1 et 3, ..., 1 et n , puis 2 et 3, ..., 2 et n , ..., $n-1$ et n .

Les lignes suivantes seront obtenues de la même manière, mais en prenant successivement les lignes 1 et 3, 1 et 4, ..., 1 et n , puis 2 et 3, ..., 2 et n , ..., $n-1$ et n de la matrice P .

La transformation $T = (P,P,Q)$ est donc une transformation qui laisse invariante F . Soit q le rang tensoriel de F et écrivons F sous la forme :

$$F(X,Y,Z) = \sum_{j=1}^q (U_j^t X)(W_j^t Z)(V_j^t Y) ,$$

$$(U_j, V_j \in K^n \quad j=1, \dots, q, \quad W_j \in K^{\frac{n(n-1)}{2}} \quad j=1, \dots, q).$$

Appliquons T à F écrite comme précédemment :

$$F(PX, PY, QZ) = \sum_{j=1}^q (U_j^t PX) (W_j^t QZ) (U_j^t PY)$$

On a toujours

$$F(PX, PY, QZ) = X^t B(Z) Y .$$

Choisissons P telle que :

$$U_1^t P = e_n$$

(cela revient à prendre la dernière colonne de $(P^t)^{-1}$ égale à U_1).

On aura donc :

$$X^t B(Z) Y = X_n^t (W_1^t QZ) (V_1^t PY) + \sum_{j=2}^q (U_j^t PX) (W_j^t QZ) (V_j^t PY) .$$

Si on fait $X_n = 0$ on obtient une nouvelle forme dont le rang tensoriel est au plus égal à $q-1$.

Or la matrice de cette forme $B'(Z)$ est la suivante :

$$B'(Z) = \begin{pmatrix} 0 & z_{12} & z_{13} & \cdots & z_{1n} \\ -z_{12} & 0 & & & z_{2n} \\ \vdots & & & & \vdots \\ -z_{1n-1} & & & & z_{n-1,n} \end{pmatrix} .$$

Comme la sous-matrice de $B'(Z)$ constituée par les $n-1$ premières lignes et colonnes est la matrice $B_{n-1}(Z)$ et que la dernière colonne est constituée par $n-1$ paramètres ne figurant pas dans $B_{n-1}(Z)$, on peut écrire :

$$\text{Rt } B'(Z) \geq \text{Rt } B_{n-1}(Z) + n-1$$

Comme $\text{Rt } B'(Z) \leq q-1$, on en déduit bien :

$$q \geq \text{Rt } B_{n-1}(Z) + n .$$

Donc (Rt $B_1(Z) = 0$) :

$$\text{Rt } B_n(Z) \geq n+n-1+\dots+3+2$$

$$\text{Rt } B_n(Z) \geq \frac{n(n+1)}{2} - 1.$$

b/ Décomposition de $B_n(Z)$ en $\frac{n(n+1)}{2} - 1$ matrices de rang un.

On peut en effet écrire $B_n(Z)$ sous la forme suivante :

$$B_n(Z) = Z_{12} \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 1 \\ 0 & -1 & & & \\ & & 0 & & \\ 0 & -1 & & & \end{pmatrix} + Z_{13} \begin{pmatrix} 0 & 0 & 1 & & \\ 0 & 0 & 1 & 0 & \\ -1 & -1 & 0 & 1 & 1 \\ & & -1 & & \\ 0 & -1 & & 0 & \end{pmatrix} + \dots$$

$$\dots + Z_{1n} \begin{pmatrix} 0 & \dots & 0 & 1 \\ & & 1 & \\ & & \vdots & \\ & & 1 & \\ -1 & \dots & -1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & z'_{23} & \dots & z'_{2n} \\ & -z'_{23} & \ddots & & \\ \vdots & \vdots & & z'_{n-1,n} \\ 0 & -z'_{2n} & \dots & z'_{n-1,n} \end{pmatrix} .$$

(Les z'_{ij} étant des combinaisons linéaires des z_{ij}).

Les $n-1$ premières matrices de cette décomposition sont des matrices anti-symétriques (ce qui assure que la dernière matrice est de la forme indiquée) et de rang deux.

Les $2(n-1)$ matrices de rang deux ainsi mises en évidence sont les suivantes :

$$E_i = i \rightarrow \begin{pmatrix} & & & & \downarrow^i & & \\ & 0 & & & & 0 & \\ -1 & -1 & \dots & -1 & 0 & 1 & \dots & 1 \\ & & & & & & & \\ 0 & & & & & & & 0 \end{pmatrix}, \quad (i=2, \dots, n),$$

$$E'_i = i \rightarrow \begin{pmatrix} & & & & \downarrow^i & & \\ & 0 & & & & & \\ & & 1 & & & & \\ & & 1 & & & & \\ & & 1 & & & & \\ 0 & 0 & & 0 & & & \\ & & -1 & & & & \\ & & -1 & & & & \end{pmatrix}. \quad (i=2, \dots, n).$$

calcul en sept multiplications générales du produit de deux matrices 2×2 quand $K = \mathbb{N}$. (On a donné une démonstration de cette optimalité dans (12''), pour un corps K quelconque, démonstration différente de celle du paragraphe 2).

Toutes les minorations obtenues avec l'aide de ces techniques sont linéaires en le nombre d'indéterminées. Par exemple, pour le produit de deux matrices $n \times n$, on a obtenu une borne inférieure de $2n^2 - n$ multiplications seulement (la méthode de Strassen conduit à utiliser $n^{\log_2 7}$ multiplications) Théoriquement, il est possible, en combinant l'utilisation des transformations invariantes avec celle du théorème 1, d'obtenir des minorations non linéaires. Cependant aucun exemple de tels résultats n'est encore connu pour le calcul de plusieurs formes bilinéaires. Ceci reste donc une question ouverte.

Avant de clôturer cette partie, il semble indispensable d'évoquer d'autres résultats sur ce même sujet.

En ce qui concerne les méthodes générales de minoration du rang tensoriel, on peut penser utiliser les théorèmes de Winograd (6') et Fiduccia (35) valables pour des calculs plus généraux que ceux de plusieurs formes bilinéaires. On va donc examiner rapidement à quels résultats ces théorèmes conduiraient dans ce cas. Ces théorèmes donnent des minorations pour le nombre de multiplications générales nécessaires pour le calcul du produit $M Y$ où M est une matrice $(m \times n)$ à éléments dans $K[x_1, \dots, x_p]$ et Y un vecteur (y_1, \dots, y_n) .

Il y a trois façons possibles de mettre le calcul de plusieurs formes bilinéaires sous cette forme (la matrice M pourra être égale à l'une des matrices $B(Z)$, $B'(X)$, $B''(Y)$). On va supposer dans la suite que M est la matrice $B(Z)$.

Le théorème sur les lignes (ou sur les colonnes) de M s'énonce ainsi :

" Si k lignes (ou colonnes) de la matrice M sont K linéairement indépendantes, alors le nombre de multiplications générales pour faire le produit $M Y$ est minoré par k " .

Ce théorème correspond, pour le calcul des formes bilinéaires, au cas où l'on a :

$$\text{Max}_{B \in \{B_i\}} (\text{Rang}(B)) \geq k .$$

On a alors de façon triviale :

$$\text{Rt } B(Z) \geq k \quad (\text{cas non commutatif})$$

$$\text{Rt } M(Z) \geq k \quad (\text{cas commutatif}) \quad \text{si } M(Z)+M(Z)^t = \begin{pmatrix} 0 & B(Z) \\ B(Z)^t & 0 \end{pmatrix} .$$

Le théorème suivant est dû à Fiduccia (35) :

" Si la matrice M possède une sous-matrice M' ($s \times c$) telle que si $\alpha^t M \beta \in K$, $\alpha \in K^s$, $\beta \in K^c$, on ait soit $\alpha = 0$, soit $\beta = 0$, alors il faut au moins $s+c-1$ multiplications générales pour calculer $M Y$ ".

Il est facile de voir que ce théorème correspond à un cas particulier du théorème 1 dans le cas du calcul de formes bilinéaires (on peut alors dans l'énoncé de ce théorème remplacer la condition $\alpha^t M \beta \in K$ par la condition $\alpha^t M \beta = 0$).

Supposons, pour simplifier, que l'on ait $M' = M$ ($M = B(Z)$) et que le théorème ci-dessus soit satisfait.

On a donc :

$$\sum_{i,j}^{m,n} B_{i,j,k} U_i V_j = 0 \quad (k=1, \dots, p) \Rightarrow U = 0 \quad \text{ou} \quad V = 0$$

On doit donc avoir par exemple :

$$\left[\sum_{i=1}^m \left(\sum_{j=1}^n B_{i,j,k} V_j \right) U_i = 0 \quad (k=1, \dots, p), V \neq 0 \right] \Rightarrow U = 0 \quad (1)$$

Ceci impose donc que l'on ait : $\rho \geq m$.

La condition (1) exprime que la matrice du système doit être de rang égal à m si V est non nul. Ceci revient à dire que la matrice $B''(Y)$ est toujours de rang m si Y est non nul.

D'après le théorème 1, on a donc bien :

$$R^t B''(Y) \geq m+n-1 \quad (\text{cas non commutatif}).$$

$$R^t M(Y) \geq m+n-1 \quad \text{si} \quad M(Y)+M(Y)^t = \begin{pmatrix} 0 & B''(Y) \\ B''(Y)^t & 0 \end{pmatrix},$$

(cas commutatif).

Ceci fait donc le lien entre les théorèmes de Fiduccia et Winograd et ceux que l'on a donné en (1,a). D'autre part, Van Leeuwen - Van Emde Boas (42) ont donné un théorème général de minoration du nombre de multiplications générales nécessaires pour calculer plusieurs formes bilinéaires analogue au théorème 1, ainsi qu'une démonstration de l'optimalité du produit vectoriel de deux vecteurs en cinq multiplications. Cette démonstration est totalement différente de celle donnée dans le paragraphe 4 et ne permet pas d'obtenir le rang tensoriel des matrices anti-symétriques. Fiduccia (35) semble avoir été le premier à donner une formule de calcul du produit vectoriel en cinq multiplications.

En ce qui concerne le produit de deux matrices $n \times n$, une minoration en $2n^2-1$ multiplications a récemment été donné par Brockett and Dobkin (43) et une formule utilisant 23 multiplications a été trouvé par Laderman (121 - page A 33) dans le cas non commutatif pour le produit de deux matrices 3×3 . Cependant l'écart entre la borne inférieure en $O(n^2)$, et la borne supérieure en $O(n^{\log_2 7})$ n'est pas réduit, car il faudrait une formule en 21 multiplications pour le produit de deux matrices 3×3 pour améliorer la méthode de Strassen.

Récemment, Kruskal (39), a montré que le problème de la recherche du rang tensoriel d'une forme trilinéaire se posait aussi dans certains problèmes de statistiques.

Zalcstein et Fiduccia (36) ont d'autre part généralisée la borne inférieure de sept multiplications nécessaires pour effectuer, avec la commutativité, le produit de deux quaternions sur R , au cas du produit de deux éléments d'une algèbre de dimension n sans diviseur de zéro : ils obtiennent dans ce cas une borne inférieure de $2n-1$ multiplications (la matrice $B(Z)$ est toujours régulière si $Z \neq 0$).

Dans le même ordre d'idée, on peut poser la question suivante :

Quelle est la complexité du produit de deux éléments dans une algèbre de Lie de dimension n ?

Les résultats des paragraphes 4 et 5 apportent un début de réponse à cette question.

REFERENCES SUR LE CHAPITRE BV

- (8") BROCKETT, RW ; DOBKIN, D. "On the optimal evaluation of a set of bilinear forms".
Proceedings of 5th ACM symposium on theory of Computing (1973).
- (31) BROCKETT, RW ; DOBKIN, D. "On the number of multiplications required for matrix multiplication". Preprint, (may 1975).
- (32) DOBKIN, D. "On the arithmetic complexity of a class of arithmetic computations".
PH. Thesis , Harward University (1973).
- (33) DE GROOTE, H.F. "On the computational complexity of quaternion multiplication".
I.P.L. vol.3, n°6 (july 1975), 117-119.
- (34) DE GROOTE, H.F. ; FISHER M.J. ; SCHONHAGE, A. "On quaternion multiplication". Preprint (1975).
- (9') FIDUCCIA, C.M. "On obtaining upper bounds on the complexity of matrix multiplication". In complexity of computer computations,
RE MILLER, JW THATCHER, editors, plenum press, New-York (1972).
- (35) FIDUCCIA, CM "On the algebraic complexity of matrix multiplication".
PHD thesis, Brown University (1973).
- (36) FIDUCCIA, CM, ZALCSTEIN, Y " Algebras having linear multiplicative complexities". State University of New-York at Stony Brook,
TR # 46, (august 1975).
- (37) FIDUCCIA C.M., ZALCSTEIN, Y "Linear upper bounds on the multiplicative complexity of certain algebras".
proc. 1975, conference on Information Science and Systems.
John Hopkins University.

- (11') HOPCROFT, JE., KERR LR, "On minimizing the number of multiplications necessary for matrix multiplication".
SIAM J. Appl. Math. 20 (1971), 30-36.
- (38) HOWELL TD., LAFON JC., "The complexity of the quaternion product".
Cornell University TR 75-245 (june 1975).
- (39) KRUSKAL, JB "Trilinear decomposition of three-way Arrays : Rank and Uniqueness in arithmetic complexity and in statistical models". Preprint (1976).
- (12") LAFON JC., "Optimum computation of p bilinear forms".
J. Linear Algebra 10-225-240 (1975).
- (40) LAFON JC., "Sur le produit de deux quaternions".
CRAS, série A t.280 (10 mars 1975), 665-668.
- (41) LAFON JC, "Minoration du rang tensoriel de p matrices et optimalité de certains calculs algébriques".
Séminaire d'Analyse Numérique, Grenoble (février 1976) n° 244.
- (42) LAFON JC, "Evaluation simultanée de plusieurs formes bilinéaires".
Journées Optimisation des algorithmes fondamentaux.
ENS PARIS, (décembre 1975).
- (4") STRASSEN, "Vermeidung Von Divisionen".
Crelle J. für die Reine und Angew. Mathematik. 264
(1973), 184-202.
- (42) VAN LEEUWEN J., VAN EMDE BOAS P. "Some elementary proofs of lower bounds in complexity theory".
Prepublication august 1975. Mathematical centrum. Amsterdam
IW 41/75.

- (6') WINOGRAD, S., "On the number of multiplications necessary to compute certain functions".
Comm. pure Applied Math. 23 (1970), 165-179.
- (15') WINOGRAD, S., "On multiplication of 2×2 matrices".
J. Linear Algebra 4 (1971), 381-388.
- (43) BROCKETT, RW. , DOBKIN, D., "On the optimal evaluation of a set of bilinear forms".
Préprint (1975), Revised version of (8").

PARTIE C

ETUDE DES PRINCIPAUX CALCULS MATRICIELS

POUR DIFFERENTS CRITERES DE COÛTS

PLAN DE LA PARTIE CIntroductionChapitre CI - Rappels sur les calculs matriciels

1. Principales notations.
2. Matrices de formes particulières.
3. Quelques propriétés des matrices.
4. Inversion d'une matrice par des méthodes de partitionnement.

Chapitre CII - Applications des résultats de la partie B

1. Inversion d'une matrice. Résolution d'un système linéaire
2. Calcul du déterminant
3. Décomposition LR et QR
4. Cas des matrices de formes particulières
5. Algorithmes utilisant les matrices de rotation élémentaires.

Chapitre CIII - Optimalité de quelques algorithmes basés sur l'emploi des matrices élémentaires.

1. Résolution d'un système linéaire
 - a/ Algorithmes optimaux
 - b/ Optimalité de Gauss pour l'obtention d'un système triangulaire
 - c/ Algorithmes optimaux de résolution d'un système linéaire
2. Transmutation d'une matrice sous formes Hessenberg, tridiagonale , et de Frobénius.
 - a/ Algorithmes optimaux
 - b/ Transmutations par des matrices élémentaires
 - c/ Méthode optimale d'obtention de la forme d'Hessenberg
 - d/ Méthode optimale de réduction sous forme tridiagonale
 - e/ Méthode optimale de réduction sous forme Frobénius.

Chapitre IV - Utilisation optimale du parallélisme

1. Produit de deux matrices.
2. Inversion d'une matrice. Résolution d'un système linéaire. Calcul d'un déterminant.
3. Décomposition LR et QR.
4. Obtention d'une forme canonique semblable à une matrice.
5. Matrices de formes particulières .

Chapitre CV - Influence de la pagination sur la rapidité d'exécution des algorithmes de calculs matriciels.

- I. Principales définitions.
 - 1/ Mémoires hiérarchisées
 - 2/ Mémoire virtuelle.
 - 3/ Pagination.
 - 4/ Gestion de la mémoire.
 - 5/ Fonctionnement en multi-programmation et en temps partagé.
 - II. Modèles mathématiques.
 - 1/ Programmes.
 - 2/ Implémentation d'un programme.
 - 3/ Stratégie de remplacement.
 - 4/ Stratégie de remplacement optimale et implémentation optimale d'un programme.
 - III. Etude de différents algorithmes.
 - 1/ Stockage d'une matrice.
 - 2/ Opérations élémentaires sur les matrices.
 - 3/ Résolution d'un système linéaire. Inversion d'une matrice.
Calcul d'un déterminant.
Décomposition LR et QR.
 - 4/ Obtention des formes canoniques.
 - 5/ Tridiagonalisation d'une matrice symétrique.
 - a/ Méthode de Givens.
 - b/ Méthode d'Householder.
 - 6/ Calcul des valeurs propres et des vecteurs propres par la méthodes de Jacobi.
- Conclusions.

INTRODUCTION

Dans cette partie, on se propose d'étudier la complexité des principaux calculs matriciels pour des critères de coûts variés.

Les calculs matriciels ainsi examinés sont les suivants : résolution d'un système linéaire, inversion d'une matrice, calcul d'un déterminant, décomposition LR ou QR d'une matrice. Obtention des formes canoniques semblables à une matrice (forme d'Hessenberg, de Frobenius et forme tridiagonale). Calcul du polynôme caractéristique, calcul des valeurs propres par les méthodes LR, QR et par la méthode de Jacobi.

Dans le chapitre I sont regroupées toutes les notations utilisées dans cette partie, ainsi que les principales définitions relatives aux algorithmes étudiés. Le chapitre se termine par l'étude de l'inversion d'une matrice par des méthodes de partitionnement.

Dans le chapitre II, on montre comment on peut appliquer les résultats de la partie B pour obtenir des algorithmes en $O(n^{\log_2^7})$ pour inverser une matrice régulière, pour calculer son déterminant et pour déterminer ses décompositions du type LR ou QR.

On montre que l'inverse et le déterminant d'une matrice cyclique peuvent se calculer en $O(n \log n)$ opérations si la FFT peut être utilisée, en $O(n^2)$ sinon. On montre enfin, comment réduire du quart le nombre de multiplications utilisées par les algorithmes basés sur l'emploi des matrices de rotations.

Dans le chapitre suivant, on étudie la classe des algorithmes qui n'utilisent que des produits par des matrices élémentaires et des calculs rationnels. On peut alors démontrer l'optimalité de certains algorithmes qui permettent d'obtenir les formes d'Hessenberg, de Frobenius, ou tridiagonale d'une matrice par des transmutations à l'aide de matrices élémentaires, ainsi que l'optimalité de la méthode de Gauss pour la résolution d'un système linéaire. Pour ce dernier problème, on montre cependant qu'il existe des algorithmes optimaux qui ne triangularisent pas la matrice du système.

Le chapitre IV est consacré à l'étude du cas où différentes opérations arithmétiques peuvent être effectuées en même temps (c'est-à-dire "en parallèle"). Il faut alors concevoir les algorithmes de telle façon qu'ils utilisent au mieux cette possibilité. On étudie les performances des principaux algorithmes de calculs matriciels quand on admet un nombre arbitraire de calculateurs "en parallèle".

On peut aussi estimer le gain maximal que peut apporter l'utilisation du parallélisme dans ces calculs.

L'intérêt de cette étude réside aussi dans le fait qu'elle montre qu'un algorithme particulier peut être très performant pour un critère de coût et être très mauvais pour un autre critère de coût.

Dans le chapitre V on étudie l'influence de la pagination sur la conception (et sur la performance) des principaux algorithmes de calculs matriciels). On suppose dans ce chapitre, que les données relatives à un calcul sont groupées en blocs appelés pages dont certaines se trouvent dans la mémoire centrale et d'autres dans une mémoire secondaire. Pour qu'un algorithme puisse effectuer une opération il faut que toutes les données nécessaires soient dans la mémoire principale de l'ordinateur et donc que leurs pages y soient. Si toutes les pages contenant les données ne peuvent être en mémoire centrale simultanément, il faudra, à certains instants, aller en chercher une nouvelle dans la mémoire secondaire. Ceci étant une opération très coûteuse, il s'agit d'organiser les calculs de façon à minimiser ces cas. On montre comment procéder de manière optimale dans le cas des principaux calculs matriciels.

CHAPITRE 1

RAPPELS SUR LES CALCULS MATRICIELS

PLAN

1. Principales notations.
2. Matrices de formes particulières.
3. Quelques propriétés sur les matrices.
4. Inversion d'une matrice par des méthodes de partitionnement.
 - a/ Partitionnement d'une matrice carrée stable pour la multiplication.
 - b/ Construction de tels partitionnements.
 - c/ Calcul de l'inverse d'une matrice :
 - 1) Cas d'un groupe cyclique
 - 2) Cas d'un groupe abélien.

Références

1 . PRINCIPALES NOTATIONS .

K désignera toujours un corps de caractéristique différente de deux. L'ensemble des matrices à m lignes et n colonnes sur K sera noté $\mathcal{M}_{m,n}(K)$. La i ème colonne (resp. la i ème ligne) de A sera notée $A_{\cdot,i}$ (resp. $A_{i,\cdot}$). L'élément de A se trouvant à l'intersection de la i ème ligne et de la j ème colonne sera noté $a_{i,j}$. La transposée de la matrice A de $\mathcal{M}_{m,n}(K)$ est la matrice $A^t = (a'_{ij})$ de $\mathcal{M}_{n,m}(K)$ avec $a'_{ij} = a_{ji}$. Le déterminant de la matrice A de $\mathcal{M}_{m,m}(K)$ sera noté : $\det A$.

Le polynome caractéristique de la matrice A est le polynome $\det (A - \lambda I)$.

On note par $\Delta_{m,k}$ l'ensemble des k uplets dont les k éléments appartiennent à $\{1, 2, \dots, m\}$ et tels que si $\alpha = (\alpha_1, \dots, \alpha_k) \in \Delta_{m,k}$ on ait : $1 \leq \alpha_1 < \alpha_2 < \dots < \alpha_k \leq m$.

$(\Delta_{m,k})$ contient $\binom{m}{k}$ éléments et il est vide si $m < k$. On désigne par

$\alpha' = (\alpha'_1, \dots, \alpha'_{m-k}) \in \Delta_{m,m-k}$ le complémentaire de α par rapport à $\{1, 2, \dots, m\}$.

Si $A \in \mathcal{M}_{m,n}(K)$, $\alpha \in \Delta_{m,k}$, et $\beta \in \Delta_{n,\ell}$, alors :

$$A(\alpha, \beta) \equiv A \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_k \\ \beta_1 & \beta_2 & \dots & \beta_\ell \end{pmatrix} \equiv (a_{\alpha_i \beta_j}) \in \mathcal{M}_{k,\ell}.$$

L'adjointe de $A \in \mathcal{M}_{n,n}(K)$ est la matrice $A^A = (b_{ij})$ de $\mathcal{M}_{n,n}(K)$ telle que :

$$b_{ij} = (-1)^{i+j} \det A \begin{pmatrix} 1, \dots, j-1, j+1, \dots, n \\ 1, \dots, i-1, i+1, \dots, n \end{pmatrix}.$$

On a évidemment :

$$A A^A = A^A A = (\det A) I_n.$$

Donc, si A est régulière son inverse est donnée par :

$$A^{-1} = (\det A)^{-1} A^A .$$

On désigne par E_{ij} ($i=1, \dots, n$, $j=1, \dots, n$) les n^2 matrices de la base canonique de $\mathcal{M}_{n,n}(K)$.

Matrices élémentaires

On appelle matrice élémentaire soit une matrice qui s'écrit sous la forme $I + a E_{ij}$ avec $a \in K$, soit une matrice de transposition V_{ij} , c'est-à-dire une matrice égale à l'identité sauf dans les lignes i et j qui ont la forme suivante :

$$\begin{array}{cc} \text{col. } i & \text{col } j \\ \left(\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right) & \begin{array}{l} \text{ligne } i \\ \text{ligne } j \end{array} \end{array} .$$

On note par $E_{ij}(a)$ la matrice $I + a E_{ij}$ si $i \neq j$.

Si $i=j$, on note par $E_i(a)$ la matrice $I + (a-1) E_{ii}$ ($a \neq 0$).

Toutes ces matrices sont régulières et on a :

$$E_i(a)^{-1} = E_i\left(\frac{1}{a}\right), E_{ij}(a)^{-1} = E_{ij}(-a), V_{ij}^{-1} = V_{ij} .$$

2 . MATRICES DE FORMES PARTICULIERES

Soit $A \in \mathcal{M}_{m,m}(K)$.

a/ Matrice triangulaire inférieure.

A sera dite triangulaire inférieure si $a_{ij} = 0 \quad \forall j > i$.

$$A = \begin{pmatrix} x & & & \\ & \cdot & & 0 \\ & \cdot & \cdot & \\ & x & \dots & x \end{pmatrix} .$$

Si de plus on a : $a_{ii} = 1$, $i=1, \dots, m$, alors A est dite triangulaire inférieure unitaire (ou unité).

b/ Matrice triangulaire supérieure.

A est triangulaire supérieure si A^t est triangulaire inférieure.

c/ Matrice Hessenberg inférieure (supérieure).

Une matrice A sera dite Hessenberg inférieure (supérieure) si

$$a_{ij} = 0 \quad \forall j > i+1 \quad (\forall j < i-1) .$$

$$A = \begin{pmatrix} x & x & & 0 \\ & \ddots & \ddots & \\ & & \ddots & x \\ x & & & x \end{pmatrix} .$$

d/ Matrice tridiagonale.

A est tridiagonale si elle est à la fois Hessenberg supérieure et inférieure :

$$A = \begin{pmatrix} x & x & & 0 \\ x & \ddots & \ddots & \\ & \ddots & \ddots & x \\ 0 & & x & x \end{pmatrix} .$$

e/ Matrice de Frobenius simple.

A sera dite de forme de Frobenius simple si elle peut s'écrire sous la forme :

$$A = \begin{pmatrix} 0 & & & p_0 \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \\ 0 & & 0 & p_{n-2} \\ & & 1 & p_{n-1} \end{pmatrix} .$$

$$a_{i+1,i} = 1 \quad , \quad i=1, \dots, n-1 \quad \text{et} \quad a_{i,n} = p_{n-i+1} .$$

On a alors :

$$\det(A-\lambda I) = (-1)^n (\lambda^n - p_{n-1} \lambda^{n-1} - p_{n-2} \lambda^{n-2} + \dots + p_0) .$$

f/ Matrice de Frobenius générale.

Une matrice A sera de forme de Frobenius générale si elle peut s'écrire sous la forme suivante :

$$A = \begin{pmatrix} F_1 & & & 0 \\ & F_2 & & \\ & & \ddots & \\ 0 & & & F_k \end{pmatrix} \text{ avec } F_i \text{ matrice de Frobenius simple.}$$

g/ Matrice de Jordan.

Une matrice A sera dite de Jordan si elle s'écrit sous la forme suivante :

$$A = \begin{pmatrix} a & 1 & & & 0 \\ & a & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 0 & & & & a \end{pmatrix}.$$

h/ Matrice de Jordan générale.

Une matrice de forme de Jordan générale peut s'écrire :

$$A = \begin{pmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_k \end{pmatrix}, \text{ les } k \text{ matrices } J_1, \dots, J_k \text{ étant de Jordan.}$$

i/ Matrices de rotations élémentaires. Matrices unitaires.

On note par $V_{ij}(\theta)$ la matrice de rotation élémentaire suivante :

$$V_{ij}(\theta) = \cos\theta E_{ii} + \sin\theta E_{ij} - \sin\theta E_{ji} + \cos\theta E_{jj}.$$

Une matrice $V_{ij}(\theta)$ est unitaire (une matrice A est dite unitaire si $A A^t = I$).

Si $K = \mathbb{C}$, on note par $A^* = (a_{ij}^*)$ la matrice transposée conjuguée de A ($a_{ij}^* = \bar{a}_{ji}$). Une matrice A de $\mathcal{M}_{n,n}(\mathbb{C})$ sera dite unitaire si :
 $AA^* = I$. Sur \mathbb{C} , on peut utiliser les matrices de rotation élémentaire suivantes :

$$R_{ij} = \bar{c} E_{ii} + \bar{s} E_{ij} - s E_{ji} + c E_{jj} \quad \text{avec} \quad |c|^2 + |s|^2 = 1 .$$

j/ Matrices hermitiennes unitaires élémentaires.

Une matrice $A \in \mathcal{M}_{m,n}(\mathbb{C})$ est dite hermitienne si elle vérifie $A = A^*$.

On dira qu'elle est hermitienne unitaire élémentaire si elle s'écrit :

$$A = I - 2 U U^t \quad U^t U = I .$$

3 . QUELQUES PROPRIETES DES MATRICES.

On rappelle ici quelques résultats concernant les matrices qui seront utilisés par la suite. Les livres qui nous servent de références dans ce paragraphe sont ceux de GANTMACHER (1), GASTINEL (2), HOUSEHOLDER (4) et WILKINSON (7).

Propriété 1

/ Pour toute matrice A régulière de $\mathcal{M}_{n,n}(K)$, il existe une matrice de permutation P telle que la matrice PA puisse s'écrire sous la forme LR avec L triangulaire inférieure unitaire et R triangulaire supérieure. /

Propriété 2

/ On peut obtenir la matrice R de la décomposition LR liée à A en prémultipliant A par des matrices élémentaires (algorithme de Gauss). /

Propriété 3

/ Pour toute matrice A régulière, il existe une matrice triangulaire inférieure unitaire T telle que $TA = A'$ avec A' orthogonale en lignes. On peut obtenir T comme un produit de matrices élémentaires. (procédé d'orthogonalisation de Schmidt). /

Propriété 4

/ Pour toute matrice A régulière de $\mathcal{M}_{n,n}(\mathbb{K})$ il existe une décomposition de la forme :

$$A = LQ$$

avec L triangulaire inférieure, et Q unitaire, ainsi qu'une décomposition de la forme :

$$A = Q_1 R$$

avec Q_1 unitaire et R triangulaire supérieure. /

Propriété 4'

/ Pour toute matrice A de $\mathcal{M}_{n,n}(\mathbb{K})$ il existe une matrice Q unitaire produit de $n(n-1)/2$ matrices de rotations élémentaires telles que :

$$QA = R . /$$

Propriété 5

/ Toute matrice A symétrique définie positive de $\mathcal{M}_{n,n}(\mathbb{R})$ peut s'écrire sous la forme :

$$A = RR^T$$

triangulaire inférieure. /

Propriété 6

/ Soit $F(\lambda) = (-1)^n (\lambda^n - \sum_{k=1}^n p_k \lambda^{n-k})$ le polynôme caractéristique de la matrice A

Alors on a : (théorème de Cayley-Hamilton) : $F(A) = 0$.

D'autre part : $\det A = (-1)^{n-1} p_n$.

Si A est inversible, on a :

$$A^{-1} = p_n^{-1} (A^{n-1} - p_1 A^{n-2} - \dots - p_{n-1} I) . /$$

Propriété 7

/ Toute matrice A de $\mathcal{M}_{n,n}(K)$ peut être transmuée en une matrice de forme Frobénius générale par des transmutations de caractère rationnel sur K :

$$\mathcal{M}^{-1} A \mathcal{M} = F.$$

\mathcal{M} peut s'écrire comme un produit de matrices élémentaires, ou comme un produit de matrices du second degré (Danilevski modifié).

De plus si, sur K , le polynôme caractéristique de A est irréductible, on obtient ainsi une forme de Frobénius simple. /

Propriété 8

/ Toute matrice A de $\mathcal{M}_{n,n}(K)$ peut être transmuée en une matrice de forme tri-diagonale par des transmutations de caractère rationnel sur K , soit en utilisant des transmutations par des matrices élémentaires, soit en utilisant des transmutations par des matrices du second degré. /

Propriété 9

/ Toute matrice A de $\mathcal{M}_{n,n}(K)$ peut être transmuée en une matrice d'Hessenberg, supérieure ou inférieure par des transmutations de caractère rationnel sur K et n'utilisant que des matrices élémentaires. /

Propriété 10

/ Si A est une matrice hermitienne de $\mathcal{M}_{n,n}(C)$, on peut la transmuier en une tridiagonale hermitienne en utilisant, soit des transmutations par des matrices de rotation élémentaires (méthode de Givens) soit des transmutations par des matrices hermitiennes unitaires élémentaires (méthode de Householder). /

Propriété 11

/ Toute matrice A de $\mathcal{M}_{n,n}(C)$ peut être transmuée en une matrice d'Hessenberg supérieure (ou inférieure) par une suite de transmutations, par des matrices de rotation élémentaires, ou par une suite de transmutations par des matrices hermitiennes unitaires élémentaires. /

Remarque

Les propriétés 1 à 6 sont liées aux principales méthodes de résolution directe d'un système linéaire $AX = B$ et au calcul de l'inverse d'une matrice. Les propriétés 6 à 11 sont liées au problème du calcul des valeurs propres d'une matrice : l'obtention des formes tridiagonales ou de Frobenius semblables à une matrice permet de calculer le polynôme caractéristique de ces matrices.

L'obtention de la forme d'Hessenberg supérieure est très importante car dans la pratique on applique l'algorithme LR (Rutishauser) ou l'algorithme QR (Francis) sur cette forme qui reste invariante par rapport à la méthode LR ou QR).

4 . INVERSION D'UNE MATRICE PAR DES METHODES DE PARTITIONNEMENT.

Soit M une matrice inversible de $\mathcal{M}_{n,n}(K)$.

Ecrivons la matrice M sous la forme suivante :

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

avec $A \in \mathcal{M}_{p,p}(K)$, $D \in \mathcal{M}_{n-p,n-p}(K)$, $C \in \mathcal{M}_{n-p,p}(K)$, $B \in \mathcal{M}_{p,n-p}(K)$.

Ceci correspond à un partitionnement de M en quatre sous-matrices. Pour un tel partitionnement, on peut calculer l'inverse de M par la formule classique suivante (si A est régulière) :

$$M^{-1} = \begin{pmatrix} A^{-1} + A^{-1} B Q^{-1} C A^{-1} & -A^{-1} B Q^{-1} \\ -Q^{-1} C A^{-1} & Q^{-1} \end{pmatrix}$$

avec $Q = D - C A^{-1} B$.

Si A est régulière, alors on peut montrer que Q l'est aussi (M étant supposée inversible). On peut donc énoncer le théorème suivant :

Théorème 1

/ Pour toute matrice inversible M de $\mathcal{M}_{n,n}(K)$ et pour tout entier p compris entre 1 et n , il existe une matrice de permutation P telle que PM puisse s'inverser par la formule 1. /

□ Montrons que pour p donné, on peut déterminer P telle que la matrice A de la matrice PM soit inversible. La matrice M étant régulière, ses p premières colonnes sont linéairement indépendantes. Il existe donc p indices de lignes i_1, \dots, i_p tels que la sous-matrice $p \times p$ de M formée par les éléments communs à ses p lignes i_1, \dots, i_p et aux p premières colonnes soit régulière. On prendra donc une permutation P des lignes de M telle que les lignes i_1, \dots, i_p deviennent les p premières lignes de M .

Démontrons maintenant que si A est inversible, alors Q l'est aussi.

Prémultiplions M par la matrice $\begin{pmatrix} A^{-1} & 0 \\ -CA^{-1} & I_{n-p} \end{pmatrix}$.

On obtient :

$$\begin{pmatrix} A^{-1} & 0 \\ -CA^{-1} & I_{n-p} \end{pmatrix} M = \begin{pmatrix} I_p & A^{-1} B \\ 0 & D - CA^{-1} B \end{pmatrix} = \begin{pmatrix} I_p & A^{-1} B \\ 0 & Q \end{pmatrix}.$$

On a donc :

$$(\det A^{-1}) \det M = \det Q.$$

Par conséquent, si M et A sont régulières on a bien Q régulière. □

On va maintenant étudier dans quelle mesure on peut concevoir des partitionnements plus généraux et comment on pourrait les utiliser pour inverser une matrice.

a/ Partitionnement d'une matrice carrée stable pour la multiplication

On désigne par E l'ensemble des N matrices E_{ij} qui forment la base canonique de $\mathcal{M}_{N,N}(K) = M_N$.

$$P = (P_1, P_2, \dots, P_m)$$

désigne une partition de E en m sous-ensembles P_i .

Remarque 1

Une partition P correspond en fait à une décomposition de $\mathcal{M}_{N,N}(K) = M_N$ en somme directe de m sous-espaces $(M_N)_i$ $i=1, \dots, m$.
 $(M_N)_i$ est le sous-espace engendré par les éléments de P_i .

$$M_N = \sum_{i=1}^m \oplus (M_N)_i$$

$$\forall M \in M_N, \quad M = \sum_{i=1}^m M_i, \quad M_i \in (M_N)_i.$$

Définition 2

/ Les sous-matrices M_1, \dots, M_m de M constituent un partitionnement de la matrice M . Ce partitionnement sera dit de type P . /

On s'intéresse aux diverses opérations matricielles pour un partitionnement P des matrices de M_N .

Il est clair que la seule difficulté provient du produit matriciel. On est donc amenés à considérer les partitionnements P que l'on qualifiera de stables vis-à-vis de la multiplication matricielle.

Définition 3

/ P est un partitionnement stable si :

$$\forall M_i \in (M_N)_i \quad \forall M_j \in (M_N)_j \quad \exists k \mid M_i M_j \in (M_N)_k . /$$

Désignons par P_i la matrice carrée d'ordre N n'ayant que des zéros et des uns et qui est la somme des matrices E_{ij} de l'ensemble P_i .

Si on considère ces matrices comme construites sur une algèbre de Boole à deux éléments (0) et (1), on a immédiatement la proposition suivante :

Proposition 1

/ Le partitionnement P sera stable pour la multiplication matricielle si et seulement si l'ensemble des matrices P_i forme un semi-groupe (pour la multiplication des matrices booléennes). /

$$\text{Exemple : } N = 2 \quad P_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} .$$

Dans toute la suite nous considérerons les partitionnements P tels que les matrices P_i forment un groupe.

b/ Construction de tels partitionnements.

(cf. Hall, M. [3]).

Soit G un groupe. A tout élément $a \in G$ on associe l'application γ_a de G en lui-même défini par :

$$\forall x \in G \quad \gamma_a(x) = a.x \quad (\text{translation à gauche})$$

(L'opération du groupe est notée multiplicativement).

Il est clair que γ_a est une bijection, que G est un domaine d'opérateur réduit et que le groupe Γ de ces translations à gauche est isomorphe à G . De ceci on déduit pour un groupe fini d'ordre m le résultat suivant :

Théorème

/ Tout groupe fini G d'ordre m est isomorphe à un sous-groupe du groupe symétrique S_m . /

Proposition 2

/ A tout groupe G d'ordre $m \leq N$ on peut faire correspondre un partitionnement P dont les m matrices P_i forment un groupe isomorphe à G . /

En effet, G est isomorphe à un sous-groupe Γ du groupe S_m .

Un élément γ de Γ peut être représenté par une matrice de permutation d'ordre m . Une telle matrice n'a que des zéros et des uns.

Choisissons m nombres $\lambda_k > 0$ tels que : $\sum_1^m \lambda_k = N$.

A toute matrice $\gamma_i \in \Gamma$ d'ordre m , faisons correspondre la matrice P_i carrée d'ordre N de la façon suivante :

si $\gamma_i(j,k) = 0$ on le remplace par une matrice nulle d'ordre λ_j, λ_k .
 si $\gamma_i(j,k) = 1$ on le remplace par une matrice n'ayant que des 1 et d'ordre λ_j, λ_k .

Exemple :

Au groupe G d'ordre 2 isomorphe au groupe des deux matrices

$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ et $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ on peut faire correspondre le partitionnement

suisant pour $N = 5$:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} .$$

Remarque 2

/ On n'obtient pas ainsi évidemment tous les partitionnements stables pour la multiplication. Pour s'en convaincre il suffit de considérer par exemple le cas $n=2$ et $m=2$ avec le partitionnement suivant :

$$P_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} , \quad P_2 = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} .$$

c/ Calcul de l'inverse d'une matrice.

Soit $M \in \mathcal{M}_{n,n}(K)$. Supposons M partitionnée en m sous-matrices M_1, \dots, M_m , le partitionnement considéré provenant d'un groupe G . Soit X' l'inverse de M . La matrice X' est aussi partitionnée en m sous-matrices X_1, \dots, X_m .

Le groupe G est isomorphe à un sous-groupe du groupe symétrique S_m . On désigne par g_1, \dots, g_m les générateurs du groupe G .

L'inverse X' de la matrice M doit vérifier l'équation suivante :

$$\left(\sum_{i=1}^m M_i \times g_i \right) X = K$$

$$K^t = (I, 0, \dots, 0), \quad X^t = (x_1, \dots, x_m).$$

(\times désigne le produit de Kronecker de deux matrices).

On peut résoudre cette équation en se servant uniquement des propriétés du groupe G dans le cas d'un groupe G cyclique (cf. PEASE (6)), et dans le cas d'un groupe abélien (cf. LAFON (5)).

1/ G est un groupe cyclique

A un isomorphisme près, G est le groupe $(I, S, S^2, \dots, S^{m-1})$.

$$S = \begin{pmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 1 & & & & 0 \end{pmatrix} .$$

Il faut résoudre l'équation :

$$\left(\sum_{i=0}^{m-1} M_i \times S^i \right) X = K .$$

L'algorithme suivant a été donné par PEASE (6).

Description de l'algorithme

1er pas : on multiplie à gauche (2) par $M_0^{-1} \times I$ et on élimine le terme en $(x \cdot S)$, en multipliant ensuite à gauche par $(I \times I - M_{m-1} \times S)$.

kème pas : Les termes en $(x \cdot S)$, $(x \cdot S^2)$, ..., $(x \cdot S^{k-1})$ sont supposés éliminés. Nous avons donc une équation de la forme suivante :

$$(I \times I + P_{m-k} \times S^k + P_{m-k-1} \times S^{k+1} + \dots + P_1 \times S^{m-1}) \times = H .$$

Il s'agit de montrer que l'on peut éliminer le terme en $(x \cdot S^k)$.

Multiplions par $(I \times I + (\sum_{i=1}^{k-1} Q_{k-i} \times S^i) - (P_{m-k} \times S^k))$ à gauche.

Le terme en $(x \cdot S^k)$ disparaît. Mais il faut empêcher la réapparition des termes déjà éliminés. Pour cela, il faut choisir les Q_i de telle manière que :

$$Q_h = P_{m-k} \cdot P_h - \sum_{i=1}^{h-1} Q_i \cdot P_{h-i} .$$

Ceci peut être résolu en calculant successivement Q_1, Q_2, \dots, Q_{k-1}

$$Q_1 = P_{m-k} \cdot P_1$$

$$Q_2 = P_{m-k} (P_2 - P_1^2) \text{ etc...}$$

Remarque 1

A chaque étape on obtient une équation du type suivant :

$$(A_0 \times e + \sum A_i \times g_i) \times = H .$$

Si A_0 est singulière le calcul ne peut être continué. Mais si l'une des matrices A_k parmi les restantes, n'est pas singulière, on peut multiplier par $I \times g_k^{-1}$, ce qui amène A_k à la place de A_0 , ce qui permet de continuer le processus.

Dans son article, PEASE donne ensuite deux autres cas :

- Cas d'un partitionnement correspondant au groupe $C_2 \times C_2 \times \dots \times C_2$
 $(C_2 \text{ groupe d'ordre } 2 \text{ engendré par la matrice } \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$
- Cas d'un partitionnement correspondant à un groupe G défini par la donnée de deux générateurs a et b vérifiant $a^k = b^s = e$ et $ab = ba^t$.

On va traiter le cas plus général ou le groupe G définissant le partitionnement P s'écrit sous la forme d'un produit direct de n groupes cycliques.

2/ G est un groupe abélien

On utilise le théorème de structure suivant (3).

Théorème

/ Tout groupe abélien fini d'ordre $n = P_1^{e_1} \cdot P_2^{e_2} \dots P_k^{e_k}$ est le produit direct des sous-groupes de Sylow $S(P_1), S(P_2), \dots, S(P_k)$. L'ordre du sous-groupe $S(P_i)$ est $P_i^{e_i}$ et $S(P_i)$ peut se mettre sous la forme d'un produit direct de groupes cycliques d'ordre $P_i^{e_{i1}}, P_i^{e_{i2}}, \dots, P_i^{e_{is}}$ avec $e_{i1} + e_{i2} + \dots + e_{is} = e_i$.

On va donc uniquement résoudre le problème pour un groupe abélien G produit direct de n groupes cycliques C_i d'ordre m_i .

$$G = C_1 \times C_2 \times \dots \times C_n$$

$$|C_i| = m_i \qquad |G| = m = \prod_{i=1}^n m_i$$

S_i : matrice de permutation d'ordre m_i , engendrant un groupe cyclique isomorphe à C_i .

g_i : matrice d'ordre m $g_i = (I \times I \times \dots \times S_i \times I \times \dots \times I)$
 (produit de Kronecker de n matrices, la jème étant d'ordre m_j).

G : est isomorphe au groupe engendré par les n matrices g_i .

Un élément quelconque de G peut donc être écrit sous la forme :

$$g_1^{i_1} g_2^{i_2} \dots g_n^{i_n} . \text{ Posons :}$$

$$e = g_1^{m_1} g_2^{m_2} \dots g_n^{m_n} .$$

On doit résoudre l'équation suivante :

$$(3) \quad (M_0 \times e + \sum_{i_1, \dots, i_n} M_{i_1, \dots, i_n} \times g_1^{i_1} g_2^{i_2} \dots g_n^{i_n}) X = K$$

(M_{i_1, \dots, i_n} : matrice carrée d'ordre N .)

X et K matrice rectangulaire d'ordre mN et N .

Relation d'ordre sur G .

Soient deux éléments de G : $g_1^{i_1} g_2^{i_2} \dots g_n^{i_n}$ et $g_1^{i'_1} g_2^{i'_2} \dots g_n^{i'_n}$ s'il existe

$p < n$ tel que :

$$i_1 = i'_1, \dots, i_p = i'_p \quad \text{et} \quad i_{p+1} < i'_{p+1}$$

on dira que le premier élément précède le second et on écrira :

$$g_1^{i_1} g_2^{i_2} \dots g_n^{i_n} < g_1^{i'_1} g_2^{i'_2} \dots g_n^{i'_n}$$

En permettant à tous les i_j, i'_j d'être égaux, on définit la relation \leq .

L'équation (3) devient, en écrivant dans l'ordre croissant les éléments de G :

$$(4) \quad (M_0 \times e + \sum_{i=0}^{m_n-1} M_{0, \dots, 0, i} \times g_n^i + \sum_{k, i} M_{0, \dots, 0, k, i} \times (g_{n-1}^k g_n^i) + \dots) X = K.$$

Algorithme d'élimination

Principe :

Désignons par G_n le groupe engendré par g_n

G_{n-1} le groupe engendré par g_n et g_{n-1}

G_{n-k} le groupe engendré par $g_n, g_{n-1}, \dots, g_{n-k}$.

Les sous-groupes G_i forment la suite normale suivante :

$$G = G_1 \supset G_2 \supset G_3 \dots \supset G_n$$

et

$$G_i + G_{i+1} = C_i .$$


L'algorithme que l'on va décrire est basé sur ces deux propriétés :

Dans l'équation (4), on élimine successivement tous les termes de G_n , puis de G_{n-1} , G_{n-2} , etc...

G_{n-k+1} étant éliminé, on effectue l'élimination de G_{n-k} en utilisant la relation $G_{n-k} / G_{n-k+1} = C_{n-k}$.

On élimine donc les différentes classes $g_{n-k} G_{n-k+1}$, $g_{n-k}^2 G_{n-k+1}$, ...
 ..., $g_{n-k}^{m_{n-k}-1} G_{n-k+1}$.

La seule difficulté est de montrer que si toutes les classes correspondantes aux g_{n-k} , g_{n-k}^2 , ..., g_{n-k}^{p-1} sont éliminées on pourra éliminer la classe $g_{n-k}^p G_{n-k+1}$, sans réintroduire les précédentes.

C'est ce que l'on montre dans la suite. 

b/ Description

G_n étant un groupe cyclique, on peut éliminer de l'équation (4) les termes correspondants en appliquant la procédure donnée au paragraphe II.

b1) Elimination des termes correspondants à G_{n-1}

$$G_{n-1} / G_n = C_{n-1} .$$

Montrons que l'on peut éliminer successivement les termes correspondants aux différentes classes g_{n-1} , g_{n-1}^2 , ..., $g_{n-1}^{m_{n-1}-1}$ sans que les classes déjà éliminées ne réapparaissent.

On raisonne par induction :

b2) Elimination correspondante à g_{n-1}

On normalise l'équation obtenue après b1) et on multiplie à gauche par :

$$(I \times e - \sum_{i=0}^{m_{n-1}-1} M_{0\dots 0,1,i} \times g_{n-1} g_n^i)$$

Puis on réapplique b1).

Il faut ensuite éliminer la classe correspondante à g_{n-1}^2 .

b3)

C23

b3) Supposons avoir éliminé les classes $g_{n-1}, g_{n-1}^2, \dots, g_{n-1}^{p-1}$.

Montrons que l'on peut éliminer celle de g_{n-1}^p sans réintroduire aucun terme des précédentes classes.

L'équation obtenue après normalisation s'écrit :

$$\left((I \times e + \sum_{i=0}^{m_{n-1}} M_{0 \dots 0, p, i} \times (g_{n-1}^p g_n^i) + \sum_{k=p+1}^{m_{n-1}-1} \sum_{i=0}^{m_n-1} M_{0 \dots 0, k, i} \times (g_{n-1}^k g_n^i) + \dots) \right) x = K.$$

On la multiplie à gauche par la quantité A :

$$A = \left(I \times e - \sum_{i=0}^{m_{n-1}} M_{0 \dots 0, p, i} \times (g_{n-1}^p g_n^i) + \sum_{k=1}^{p-1} \sum_{i=0}^{m_{n-1}} U_{k, i} \times g_{n-1}^k g_n^i \right)$$

Il faut choisir les $U_{k, i}$ de telle manière que les termes correspondants aux éléments $g_{n-1}^k g_n^i$ avec $k \leq p-1$ ne réapparaissent pas.

Montrons que ce calcul est possible

- Calcul de $U_{p-1, i}$ ($i=0, \dots, m_n-1$).

On doit avoir :

$$\begin{aligned} & \left(- \sum_{i=0}^{m_{n-1}} M_{0 \dots 0, p, i} \times (g_{n-1}^p g_n^i) \right) \left(\sum_{j=0}^{m_n-1} M_{0 \dots 0, m_{n-1}-1, i} \times (g_{n-1}^{m_{n-1}-1} g_n^j) \right) \\ & + \sum_{i=0}^{m_n-1} U_{p-1, i} \times (g_{n-1}^{p-1} g_n^i) = 0 \end{aligned}$$

$$\Rightarrow U_{p-1, q} = \sum_{i_1+i_2=q} M_{0 \dots 0, p, i_1} \cdot M_{0 \dots 0, m_{n-1}-1, i_2} \pmod{m_n}$$

$$q = 0, \dots, m_n-1.$$

De même on aura :

$$\begin{aligned} U_{p-2, q} &= \sum_{i_1+i_2=q} M_{0 \dots 0, p, i_1} \cdot M_{0 \dots 0, m_{n-1}-2, i_2} \\ &- \sum_{i_1+i_2=q} U_{p-1, i_1} M_{0 \dots 0, m_{n-1}-1, i_2}. \end{aligned}$$

De façon générale supposons avoir calculé les $U_{p-1,i}, U_{p-2,i}, \dots$ jusqu'à $U_{p-k+1,i}$.

Les $U_{p-k,i}$ $i=0, \dots, m_{n-1}$ devront vérifier :

$$\begin{aligned} & \left(\sum_{i=0}^{m_{n-1}} M_{0 \dots 0 p, i} \times (g_{n-1}^p g_n^i) \right) \left(\sum_{j=0}^{m_{n-1}} M_{0 \dots 0 m_{n-1} - k, i} \times (g_{n-1}^{m_{n-1} k} g_n^j) \right) \\ & + \sum_{i=0}^{m_{n-1}} U_{p-k, i} \times (g_{n-1}^{p-k} g_n^i) + \left(\sum_{i=0}^{m_{n-1}} U_{p-1, i} \times g_{n-1}^{p-1} g_n^i \right) \left(\sum_{j=0}^{m_{n-1}} M_{0 \dots 0 m_{n-1} - k + 1} \times g_{n-1}^{m_{n-1} - k + 1} g_n^j \right) \\ & + \dots + \left(\sum_{i=0}^{m_{n-1}} U_{p-k+1, i} \times g_{n-1}^{p-k+1} g_n^i \right) \left(\sum_{j=0}^{m_{n-1}} M_{0 \dots 0 m_{n-1} - 1} \times g_{n-1}^{m_{n-1} - 1} g_n^j \right) = 0 . \end{aligned}$$

on aura donc $U_{p-k,i}$ en fonction des $U_{p-k+1,j}, \dots, U_{p-1,j}$ déjà calculés. Par conséquent, l'élimination de G_{n-1} pour cette méthode est possible.

b4) Supposons avoir éliminé G_{n-k+1} et éliminons G_{n-k} . Comme on a $G_{n-k}/G_{n-k+1} = C_{n-k}$ et que C_{n-k} est un groupe cyclique on peut adopter le même procédé que pour l'élimination de G_{n-1} pour l'élimination des termes correspondant aux différentes classes :

$$g_{n-k} G_{n-k+1}, g_{n-k}^2 G_{n-k+1}, \dots, g_{n-k}^{m_{n-k}-1} G_{n-k+1} .$$

Après l'élimination de $G_1 = G$ on a donc l'inverse de M . Ceci résoud, du moins théoriquement la question pour un groupe abélien. Il reste à savoir quand la normalisation est toujours possible. Dans le cas où elle ne serait pas possible, on pourrait procéder comme pour le paragraphe 2. Mais ceci n'exclut pas la faillite du procédé pour certaines matrices.

REFERENCES SUR LE CHAPITRE C I

- (1) GANTMACHER, FR., "The theory of matrices, vol I and II".
CHELSEA, New-York, (1959).
- (2) GASTINEL, N., "Analyse Numérique Linéaire".
HERMANN, Paris (1966).
- (3) HALL, M.JR., "The theory of groups".
MacMillan Company, New-York (1959).
- (4) HOUSEHOLDER, AS. "The theory of matrices in Numerical Analysis".
Blaisdell publishing Co, New-York, (1965).
- (5) LAFON, J.C. "Calcul de l'inverse d'une matrice par des méthodes de
partitionnement." Séminaire d'Analyse Numérique, Grenoble,
N° 114, (décembre 1970).
- (6) PEASE, MC. "Inversion of matrices by partitioning".
J. ACM vol 16, N°2, (April 1969) 302-314.
- (7) WILKINSON J.H. "The algebraic eigenvalue problem".
Clarendon Press, Oxford (1965).

CHAPITRE CII

APPLICATIONS DES RESULTATS DE LA PARTIE B

PLAN

1. Inversion d'une matrice. Résolution d'un système linéaire.
 2. Calcul du déterminant.
 3. Décomposition LR et QR.
 4. Cas des matrices de formes particulières.
 5. Algorithmes utilisant les matrices de rotations élémentaires.
- Références.

1 . INVERSION D'UNE MATRICE REGULIERE.

Soit M une matrice régulière de $\mathcal{M}_{n,n}(K)$. La méthode de Gauss permet le calcul de l'inverse de M avec un nombre de multiplications de l'ordre de n^3 .

Le résultat suivant montre que cette méthode n'est pas la meilleure (cf. STRASSEN (1')).

Théorème 1

/ On peut calculer l'inverse de toute matrice régulière de $\mathcal{M}_{n,n}(K)$ en utilisant $O(n^{\log_2^7})$ multiplications. /

□ On va raisonner dans le cas où la dimension n de la matrice M considérée est une puissance de deux. S'il n'en est pas ainsi on pourrait considérer la matrice suivante :

$$\begin{pmatrix} M & 0 \\ 0 & I \end{pmatrix},$$

de taille 2^k avec $k = \lceil \log_2 n \rceil$, dont l'inverse est la matrice :

$$\begin{pmatrix} M^{-1} & 0 \\ 0 & I \end{pmatrix}.$$

On suppose donc $n = 2^k$. D'après le théorème I.1, on sait qu'il existe une matrice de permutation P_k telle que l'on puisse écrire :

$$P_k M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

avec A matrice régulière (A, B, C, D sont quatre matrices carrées de taille 2^{k-1}).

On peut donc inverser la matrice $P_k M$ en utilisant la formule classique suivante :

$$(1) \quad (P_k M)^{-1} = \begin{pmatrix} A^{-1} + A^{-1} B Q^{-1} C A^{-1} & -A^{-1} B Q^{-1} \\ -Q^{-1} C A^{-1} & Q^{-1} \end{pmatrix},$$

avec $Q = D - C A^{-1} B$.

Comme on a : $(P_k M)^{-1} = M^{-1} P_k$, on aura l'inverse de M sans aucun autre calcul.

Désignons par $I(n)$ le nombre de multiplications nécessaires pour inverser une matrice de taille n , et par $M(n)$ le nombre de multiplications nécessaires pour effectuer le produit de deux matrices de taille n .

Par application de la formule 1, on voit que pour inverser la matrice M de taille 2^k , on est ramené à inverser deux matrices de taille 2^{k-1} et à effectuer six produits de matrices de taille 2^{k-1} , on a donc :

$$(2) \quad I(2^k) = 2 I(2^{k-1}) + 6 M(2^{k-1}).$$

Mais, avec l'application récursive des formules de STRASSEN pour effectuer le produit de deux matrices on a :

$$M(n) = n^{\log_2^7},$$

c'est-à-dire :

$$M(2^k) = 7^k.$$

La relation (2) donne (comme $I(1) = 1$ et $M(1) = 1$) :

$$I(2^k) = 2^k + 6 \cdot 7^{k-1} \left(\sum_{q=0}^{k-1} \left(\frac{2}{7}\right)^q \right) - 2.$$

$$I(2^k) = 2^k + 6 \cdot 7^{k-1} + \frac{\left(\frac{2}{7}\right)^k - 1}{\left(\frac{2}{7}\right) - 1} - 2.$$

$$\text{Soit} \quad I(2^k) = \frac{6}{5} 7^k - \frac{2^k}{5} - 2.$$

On a donc bien dans le cas général :

$$I(n) = O(n^{\log_2^7}). \quad \square$$

Remarque 1

La relation (2) montre en fait que l'on a :

$$I(n) = O(M(n)).$$

En particulier, si l'on savait faire le produit de deux matrices $n \times n$ en n^a multiplications (avec $a < \log_2^7$) on aurait également :

$$I(n) = O(n^a).$$

Remarque 2

Dans le cas où la dimension n de la matrice M considérée n'était pas une puissance de deux, on l'a considérée comme une sous-matrice d'une matrice de taille 2^k avec $k = \lceil \log_2 n \rceil$.

Cela montre bien que dans tous les cas

$I(n) = O(n^{\log_2^7})$, mais a l'inconvénient de donner une constante grande (multipliée par sept).

En fait, on aurait simplement besoin d'augmenter la taille d'une matrice de 1 au plus (dans le cas n impair) et partitionner cette matrice en quatre blocs de taille $\frac{n}{2}$. Le résultat asymptotique reste le même, mais la constante devient celle du cas $n = 2^k \left(\frac{6}{5}\right)$.

Remarque 3

Le résultat reste valable si on compte le nombre total de multiplications et d'additions utilisées pour calculer l'inverse. En effet, on sait que si $M(n)$ désigne le nombre total d'additions et de multiplications nécessaires pour effectuer le produit de deux matrices $n \times n$ par STRASSEN, on a :

$$M(n) = O(n^{\log_2^7}).$$

La relation (2) deviendra :

$$(2') \quad I(2^k) = 2 I(2^{k-1}) + 6 M(2^{k-1}) + 2^{2k-1}.$$

On en déduit donc :

$$I(n) = O(M(n)) \quad (\text{si } M(n) > n^2).$$

Corollaire 1

/ La résolution d'un système linéaire $MX = B$ avec M régulière de taille n peut se faire en $O(n^{\log_2^7})$ opérations arithmétiques. /

□ On peut calculer l'inverse de M en $O(n^{\log_2^7})$ opérations arithmétiques. Il reste donc ensuite à effectuer le produit de M^{-1} par le vecteur X ce qui ne nécessite que n multiplications et $n(n-1)$ additions au plus.

Par conséquent la résolution de $MX = B$ peut se faire en $O(n^{\log_2^7})$ comme l'inversion de M . □

Remarque

Les résultats précédents ont été établis à l'aide de la formule classique d'inversion d'une matrice par partitionnement. BUNCH et HOPCROFT (10) ont montré que la décomposition d'une matrice M sous forme d'un produit d'une matrice triangulaire inférieure unitaire L et d'une matrice triangulaire supérieure R pouvait se faire en $O(n^{\log_2^7})$ opérations arithmétiques, ce qui donne donc aussi le résultat pour la résolution d'un système linéaire, et l'inversion d'une matrice ainsi que pour le calcul d'un déterminant. D'autre part, quel que soit le nombre d'opérations arithmétiques finalement nécessaires pour inverser une matrice ($I(n)$) ou multiplier deux matrices ($M(n)$) on peut montrer que ces deux nombres seront toujours du même ordre, plus précisément, on a le théorème suivant :

Théorème 2

/ Il existe deux constantes k_1 et k_2 telles que l'on ait :

$$I(n) \leq k_1 M(n) \quad \text{et} \quad M(n) \leq k_2 I(n) \quad \text{où} \quad M(n) \quad \text{et} \quad I(n)$$

désignent le nombre d'opérations arithmétiques nécessaires pour multiplier deux matrices et pour inverser une matrice de taille n . /

□ L'existence de k_1 a été démontrée dans le théorème 1. Pour démontrer que l'on peut écrire $M(n) \leq k_2 I(n)$, il suffit de ramener le produit de deux matrices à un calcul d'inverse. Ceci peut se faire par exemple, de la manière suivante :

Soit deux matrices A et B de $\mathcal{M}_{n,n}(K)$. On désire calculer leur produit AB en se ramenant au calcul de l'inverse d'une matrice.

On peut noter que l'on a par exemple :

$$\begin{pmatrix} I & B \\ A & I \end{pmatrix}^{-1} = \begin{pmatrix} I + B(I-AB)^{-1}A & -B(I-AB)^{-1} \\ -(I-AB)^{-1}A & (I-AB)^{-1} \end{pmatrix} .$$

Pour avoir le produit AB on se ramène ainsi à l'inversion d'une matrice $n \times n$ puis à l'inversion d'une matrice $\frac{n}{2}, \frac{n}{2}$.

On a donc par ce procédé :

$$M(n) \leq I(n) + I\left(\frac{n}{2}\right) + n ,$$

et donc :

$$M(n) \leq 2 I(n) . \quad \square$$

2 . CALCUL DU DETERMINANT D'UNE MATRICE

Théorème 3

/ Le calcul du déterminant d'une matrice M de $\mathcal{M}_{n,n}(K)$ peut être fait avec $O(n^{\log_2 7})$ opérations arithmétiques. /

□ Ce résultat découle évidemment de celui de BUNCH et HOPCROFT (10) sur le calcul de la décomposition LR d'une matrice. La démonstration suivante permet d'obtenir une estimation meilleure du nombre de multiplications (divisions) et d'additions (soustractions) nécessaires pour un calcul.

Soit $d(n)$ le nombre de multiplications -divisions nécessaires pour calculer le déterminant d'une matrice M de $\mathcal{M}_{n,n}(K)$.

Si on partitionne M en quatre sous-matrices A, B, C, D le résultat du théorème I.1 s'applique :

$$\begin{aligned} \det M &= \det A \det Q , \\ &= \det AQ \end{aligned}$$

avec $Q = D - C A^{-1} B .$

Ceci ramène donc le calcul du déterminant d'une matrice d'ordre n à celui d'une matrice d'ordre $\frac{n}{2}$ au moyen de trois produits et d'une inversion de matrices d'ordre $\frac{n}{2}$.

On a donc :

$$d(n) = d\left(\frac{n}{2}\right) + 3 M\left(\frac{n}{2}\right) + I\left(\frac{n}{2}\right)$$

Si $n = 2^k$ on a exactement :

$$\begin{aligned} d(2^k) &= d(2^{k-1}) + 3 \cdot 7^{k-1} + \frac{6}{5} 7^{k-1} - \frac{2^{k-1}}{5} - 2, \\ &= d(2^{k-1}) + \frac{2 \cdot 1}{5} 7^{k-1} - \frac{2^{k-1}}{5} - 2. \end{aligned}$$

On obtient :

$$d(2^k) = \frac{7^{k+1}}{10} - \frac{1}{5} 2^k - 2k - \frac{1}{2} \dots$$

et dans le cas général :

$$d(n) = O\left(n^{\log_2 7}\right).$$

Le même résultat est valable si l'on compte aussi les additions et les soustractions. \square

3 . DECOMPOSITIONS LR ET QR

Théorème 4

/ On peut mettre une matrice M sous la forme d'un produit par une matrice triangulaire inférieure unité L et une matrice triangulaire supérieure R en $O\left(n^{\log_2 7}\right)$ multiplications. /

□ Comme on ne compte ici que le nombre de multiplications générales utilisées par les calculs on suppose que la matrice A peut se décomposer sous la forme LR (la recherche d'une permutation sur ses lignes telle que ceci ait lieu n'est pas prise en compte).

On suppose $n = 2^k$. Décomposons M en quatre sous-matrices de taille 2^{k-1} :

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} .$$

et écrivons :

$$L = \begin{pmatrix} L_1 & 0 \\ C_1 & L_2 \end{pmatrix} , \quad R = \begin{pmatrix} R_1 & B_1 \\ 0 & R_2 \end{pmatrix} .$$

L_1, L_2 sont deux matrices triangulaires inférieures unités de taille 2^{k-1} et R_1, R_2 sont deux matrices triangulaires supérieures de taille 2^{k-1} .

On doit donc avoir :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} L_1 R_1 & L_1 B_1 \\ C_1 R_1 & C_1 B_1 + L_2 R_2 \end{pmatrix} .$$

Désignons par LR(n) le coût en nombre de multiplications-divisions de l'obtention de L et de R. Le calcul de L_1 et de R_1 nécessite le coût LR $\left(\frac{n}{2}\right)$. Ensuite le calcul de $B_1 = (L_1)^{-1} B$ et de $C_1 = C R_1^{-1}$ nécessite $2 I\left(\frac{n}{2}\right)$ et $2 M\left(\frac{n}{2}\right)$.

On obtient ensuite L_2 et R_2 en décomposant la matrice $D - C_1 B_1$ ce qui nécessite le coût LR $\left(\frac{n}{2}\right)$ et $M\left(\frac{n}{2}\right)$.

On a donc en tout :

$$LR(n) = 2 LR\left(\frac{n}{2}\right) + 2 I\left(\frac{n}{2}\right) + 3 M\left(\frac{n}{2}\right) .$$

Comme $I(n) \leq \frac{6}{5} M(n)$,

on obtient :

$$LR(n) \leq 2 LR\left(\frac{n}{2}\right) + \frac{27}{5} M\left(\frac{n}{2}\right) .$$

Ceci entraîne immédiatement que l'on a :

$$LR(n) = O(M(n)) .$$

soit

$$LR(n) = O\left(n^{\log_2 7}\right) . \quad \square$$

On va maintenant étudier le problème de la décomposition QR d'une matrice M de $\mathcal{M}_{m,n}(\mathbb{C})$ ($m \geq n$).

On sait (propriété I.3) que l'on peut déterminer une matrice Q de $\mathcal{M}_{m,n}(\mathbb{C})$ et une matrice triangulaire supérieure R de $\mathcal{M}_{n,n}(\mathbb{C})$ telles que l'on ait :

$$M = Q R \quad \text{avec} \quad Q^* Q = I_n$$

La méthode de Schmidt ou celle d'Householder permet d'obtenir la décomposition de M en $O(mn^2)$ opérations arithmétiques. On va montrer que cette décomposition peut être calculée en

$$O(mn (\log_2^7 - 1)) \text{ opérations arithmétiques.}$$

Théorème 5

/ La décomposition QR d'une matrice M de $\mathcal{M}_{m,n}(\mathbb{C})$ de rang m peut être calculée avec

$$O(mn (\log_2 7 - 1)) \text{ opérations arithmétiques. /}$$

□ Supposons pour simplifier le raisonnement que l'on ait $n = 2^k$, $m = 2^{k'}$. On peut alors partitionner les matrices M , Q et R en quatre sous-matrices de la manière suivante :

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \quad Q = \begin{pmatrix} Q_1 & Q_3 \\ Q_2 & Q_4 \end{pmatrix}, \quad R = \begin{pmatrix} R_1 & I \\ 0 & R_2 \end{pmatrix},$$

avec R_1, R_2 triangulaire supérieure.

$$A, B, C, D, Q_1, Q_2, Q_3, Q_4 \in \mathcal{M}_{\frac{m}{2}, \frac{n}{2}}(\mathbb{C}), \quad R_1, R_2 \in \mathcal{M}_{\frac{n}{2}, \frac{n}{2}}(\mathbb{C}), \quad T \in \mathcal{M}_{\frac{n}{2}, \frac{n}{2}}(\mathbb{C}).$$

On désigne par $Q R(m,n)$ le coût du calcul de la décomposition QR d'une matrice de $\mathcal{M}_{m,n}(\mathbb{C})$.

La condition pour que l'on ait $Q^* Q = I_n$ devient :

$$\begin{pmatrix} Q_1^* & Q_2^* \\ Q_3^* & Q_4^* \end{pmatrix} \begin{pmatrix} Q_1 & Q_3 \\ Q_2 & Q_4 \end{pmatrix} = \begin{pmatrix} I_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{pmatrix}.$$

c'est-à-dire :

$$(1) \quad \begin{aligned} Q_1^* Q_1 + Q_2^* Q_2 &= I_{\frac{n}{2}} , & Q_1^* Q_3 + Q_2^* Q_4 &= 0 , \\ Q_3^* Q_3 + Q_4^* Q_4 &= I_{\frac{n}{2}} , & Q_3^* Q_1 + Q_4^* Q_2 &= 0 . \end{aligned}$$

La condition $A = QR$ devient :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} Q_1 R_1 & Q_1^T + Q_3 R_2 \\ Q_2 R_1 & Q_2^T + Q_4 R_2 \end{pmatrix} .$$

On doit donc avoir en particulier :

$$\begin{pmatrix} A \\ C \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} R_1 , \quad \text{avec} \quad Q_1^* Q_1 + Q_2^* Q_2 = I_{\frac{n}{2}}$$

Ceci représente donc le calcul de la décomposition QR d'une matrice de $\mathcal{M}_{m, \frac{n}{2}}(C)$. Le coût de ce calcul est donc $QR(m, \frac{n}{2})$.

Ayant calculé Q_1, R_1 et Q_2 il faut déterminer Q_3, Q_4, T et R_2 telles que l'on ait :

$$\begin{aligned} B &= Q_1^T + Q_3 R_2 , \\ D &= Q_2^T + Q_4 R_2 , \end{aligned}$$

et que les conditions (1) soient satisfaites.

Les conditions (1) permettent d'obtenir :

$$\begin{aligned} Q_1^* B + Q_2^* D &= (Q_1^* Q_1 + Q_2^* Q_2) T + (Q_1^* Q_3 + Q_2^* Q_4) R_2 \\ &= T . \end{aligned}$$

Donc $T = Q_1^* B + Q_2^* D$.

Le calcul de T nécessite $2 M(\frac{n}{2}, \frac{m}{2}, \frac{n}{2})$ opérations arithmétiques

($M(m, n, p)$ désigne le coût du calcul du produit de deux matrices de $\mathcal{M}_{m, n}(C)$ et de $\mathcal{M}_{n, p}(C)$).

On aura ensuite :

$$\begin{matrix} B - Q_1 T \\ D - Q_2 T \end{matrix} = \begin{pmatrix} Q_3 \\ Q_4 \end{pmatrix} R_2 ,$$

avec $Q_3^* Q_3 + Q_4^* Q_4 = I_{\frac{n}{2}}$.

Cette dernière étape du calcul nécessite donc, d'une part le calcul de $Q_1 T$ et de $Q_2 T$ ($2M(\frac{m}{2}, \frac{n}{2}, \frac{n}{2})$), puis d'autre part, le calcul de la décom-

sition QR de la matrice $\begin{pmatrix} B - Q_1 T \\ D - Q_2 T \end{pmatrix}$ ce qui nécessite $QR(m, \frac{n}{2})$ et $\frac{m n}{2}$

additions-soustractions.

On a donc en tout la relation :

$$QR(m,n) = 2 QR(m, \frac{m}{2}) + 2M(\frac{n}{2}, \frac{m}{2}, \frac{n}{2}) + 2M(\frac{m}{2}, \frac{n}{2}, \frac{n}{2}) + \frac{m n}{2} + \frac{n^2}{4}$$

Cette relation donne immédiatement :

$$QR(m,n) = O(m n^{\log_2 7 - 1})$$

(ou $QR(m,n) = O(m n^{a-1})$ si le produit de deux matrices $n \times n$ peut se faire en n opérations arithmétiques) \square

Ce résultat a été montré pour la première fois par SHONHAGE (13), (14) par une méthode analogue, ainsi que par l'utilisation de matrices de rotation "généralisées".

4 . CAS DES MATRICES DE FORMES PARTICULIERES.

Tout ce qui précède s'applique pour des matrices de formes tout à fait quelconques. Par contre dès que l'on considère des sous-espaces particuliers de matrices les résultats précédents peuvent ne pas donner d'améliorations sur les algorithmes déjà utilisés. On va examiner dans cette optique quelques sous-espaces de $M_{n,n}(K)$.

a/ Matrices triangulaires (inférieures ou supérieures).

Le déterminant d'une telle matrice peut se calculer de manière évidente en n multiplications seulement. Les décompositions LR et QR sont immédiates. Par contre en ce qui concerne le produit de deux matrices triangulaires inférieures entre elles, et l'inversion d'une telle matrice, on ne connaît pas de méthodes meilleures que les méthodes précédemment exposées en $O(n^{\log_2 7})$. Ceci semble montrer que si dans le cas général tous ces calculs sont de la même difficulté, il ne semble pas en être de même du tout quand on regarde des sous-ensembles particuliers de $M_{n,n}(K)$.

b/ Matrices cycliques.

On sait faire le produit de deux matrices cycliques de $M_{n,n}(C)$ en n multiplications générales et $O(n \log n)$ opérations arithmétiques. On sait également inverser une telle matrice régulière en n divisions et $O(n \log n)$ opérations arithmétiques totales (cf. B.IV).

Si on prend comme corps de base R , alors on ne peut appliquer la transformée de Fourier et le produit de deux matrices cycliques de $M_{n,n}(R)$ nécessite n^2 multiplications (et $2n^2 - n$ opérations arithmétiques). Peut-on alors faire moins que $O(n^{\log_2 7})$ opérations arithmétiques pour inverser une matrice cyclique de $M_{n,n}(R)$?

Combien coûte le calcul du déterminant d'une matrice cyclique de $M_{n,n}(C)$ ou de $M_{n,n}(R)$?

Soit donc C une matrice cyclique de $M_{n,n}(C)$ ou $M_{n,n}(R)$. On peut écrire C sous la forme suivante :

$$\begin{pmatrix} C_1 & C_2 & \dots & C_n \\ C_n & C_1 & \dots & C_{n-1} \\ \vdots & & & \\ C_2 & C_3 & \dots & C_n & C_1 \end{pmatrix} \quad \begin{array}{l} C_i \in C \text{ ou } C_i \in R \\ (i=1, \dots, n) . \end{array}$$

Sous cette forme, on ne peut utiliser les méthodes de partitionnement précédemment utilisées car les sous-matrices obtenues ne seraient pas cycliques.

On peut cependant utiliser le résultat suivant :

Lemme 1

Il existe deux matrices de permutation P et Q telles que l'on ait :

$$P C P = \begin{pmatrix} C_1 & C_2 \\ C_2 Q & C_1 \end{pmatrix}, \quad \text{avec } C_1, C_2 \text{ cycliques.}$$

□ Montrons ce résultat dans les cas $n = 2k$.

Il suffit de constater que l'on peut prendre :

$$C_1 = \begin{pmatrix} C_1 & C_3 & \dots & C_{2k-1} \\ C_{2k-1} & C_1 & C_3 & \dots & C_{2k-3} \\ \vdots & & & & \\ C_3 & C_5 & \dots & C_1 \end{pmatrix}, \quad C_2 = \begin{pmatrix} C_2 & C_4 & C_6 & \dots & C_{2k} \\ C_{2k} & C_2 & C_4 & \dots & C_{2k-2} \\ \vdots & & & & \\ C_4 & C_6 & \dots & C_2 \end{pmatrix}$$

et

$$C_2 Q = \begin{pmatrix} C_{2k} & C_2 & C_4 & \dots & C_{2k-2} \\ C_{2k-2} & C_{2k} & & & \\ & & \dots & & \\ & & & \dots & \\ C_2 & & & & C_{2k} \end{pmatrix}. \quad \square$$

On peut alors énoncer le théorème suivant :

Théorème 6

Le calcul de l'inverse d'une matrice cyclique à éléments réels (de $\mathcal{M}_{n,n}(\mathbb{R})$) peut se faire avec $O(n^2)$ opérations arithmétiques. /

□ Soit $C \in \mathcal{M}_{n,n}(\mathbb{R})$. D'après le lemme 1 on peut écrire :

$$P C P = \begin{pmatrix} C_1 & C_2 \\ C_2 Q & C_1 \end{pmatrix}.$$

(P et Q étant deux matrices de permutation et $C_2 Q$ est aussi cyclique de même que C_1 et C_2).

On peut donc écrire :

$$P C^{-1} P = \begin{pmatrix} C_1^{-1} + C_1^{-1} C_2 R^{-1} C_2 Q C_1^{-1} & -C_1^{-1} C_2 R^{-1} \\ -R^{-1} C_2 Q C_1^{-1} & R^{-1} \end{pmatrix}$$

avec $R = C_1 - C_2 Q C_1^{-1} C_2$.

Soit $I_c(n)$ le nombre de multiplications nécessaires pour inverser une matrice cyclique de $\mathcal{M}_{n,n}(R)$. La formule précédente montre que pour calculer C^{-1} il faut faire cinq produits de matrices cycliques $\frac{n}{2}$, $\frac{n}{2}$ et deux inversions de matrices cycliques $\frac{n}{2}$, $\frac{n}{2}$.

On a donc :

$$I_c(n) = 2 I_c\left(\frac{n}{2}\right) + 5 M_c\left(\frac{n}{2}\right)$$

Si $M_c(n) = n^2$ on obtient :

$$I_c(n) = O(n^2) .$$

(si $n = 2^k$ on a : $I_c(2^k) = \frac{5}{2} (2^k)^2 - \frac{3}{2} (2^k)$.).

Le résultat reste le même si on considère toutes les opérations arithmétiques ($M_c(n) = 2n^2 - n$ et le calcul de R nécessite n additions-soustractions en plus). \square

Corollaire 1

Le calcul du déterminant d'une matrice cyclique de $\mathcal{M}_{n,n}(R)$ peut se calculer avec $O(n^2)$ opérations arithmétiques.

\square On peut en effet raisonner comme pour le théorème 3. On a :

$$\det C = \det(P C P) = \det(C_1 R)$$

avec $R = C_1 - C_2 Q C_1^{-1} C_2$.

Le calcul de R nécessite le calcul de C_1^{-1} ($O(\frac{n}{2})^2$) et le calcul des deux produits de matrices cycliques $C_1^{-1} C_2$ et $C_2 Q$. Il faut ensuite faire encore le produit $C_1 R$. On a donc :

$$(2) \quad D_c(n) = D_c\left(\frac{n}{2}\right) + 3 M_c\left(\frac{n}{2}\right) + I_c\left(\frac{n}{2}\right)$$

$$\Rightarrow D_c(n) = O(n^2) . \quad \square$$

Corollaire 2

Le calcul du déterminant d'une matrice cyclique de $\mathcal{M}_{n,n}(C)$ peut se calculer en $O(n \log n)$ opérations arithmétiques.

□ Dans ce cas on a :

$$M_c(n) = O(n \log n) \quad \text{et} \quad I_c(n) = O(n \log n).$$

La relation (2) donne donc :

$$D_c(n) = O(n \log n) . \quad \square$$

c/ Matrices de Toeplitz et de Hankel.

D'après les résultats de la partie B (chapitre IV), le produit d'une matrice de Toeplitz (ou de Hankel) de $\mathcal{M}_{n,n}(C)$ par un vecteur peut se faire avec $2n-1$ multiplications générales et $O(n \log n)$ opérations arithmétiques. Par conséquent, le produit de deux matrices de Toeplitz (ou de Hankel) peut se faire avec $2n^2 - n$ multiplications générales et $O(n^2 \log n)$ opérations arithmétiques totales. Sur R ou sur Q on ne sait pas faire le produit de deux matrices de Toeplitz (ou de Hankel) avec moins de

$O(n^{\log_2 7})$ opérations arithmétiques totales.

Par contre, on sait résoudre un système linéaire du type $A X = b$ avec $A \in \mathcal{M}_{n,n}(K)$, en $O(n^2)$ opérations arithmétiques totales si la matrice A est de Toeplitz (ou de Hankel). cf. (8), (9), (12), (15), (16).

5 . ALGORITHMES UTILISANT LES MATRICES DE ROTATION ELEMENTAIRES.

La méthode de Givens permet de triangulariser une matrice en la prémultipliant à gauche par des matrices de rotation élémentaires. (On obtient ainsi la matrice Q de la décomposition QR sous forme d'un produit de matrices de rotation). On peut également obtenir une matrice d'Hessenberg semblable à une matrice donnée en utilisant des transmutations par des matrices de rotation. On utilise également des transmutations par des matrices de rotation dans la méthode de Jacobi.

On va montrer comment on peut diminuer du quart le nombre de multiplications-divisions utilisées par toutes ces méthodes (cf. LAFON (11)).

Soit en effet $V_{ij}(\theta)$ une matrice de rotation élémentaire.

$$V_{ij}(\theta) = \cos \theta E_{ii} + \cos \theta E_{jj} - \sin \theta E_{ij} + \sin \theta E_{ji} .$$

Quand on multiplie une matrice A à gauche (resp. à droite) par une matrice telle que $V_{ij}(\theta)$, on ne modifie que les deux lignes (resp. les deux colonnes) i et j de cette matrice.

Tout revient à effectuer le produit :

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} A_{i1} & \dots & A_{in} \\ A_{j1} & \dots & A_{jn} \end{pmatrix} = \begin{pmatrix} A'_{i1} & \dots & A'_{in} \\ A'_{j1} & \dots & A'_{jn} \end{pmatrix} .$$

On peut écrire :

$$A'_{ik} = \cos \theta A_{ik} - \sin \theta A_{jk} ,$$

$$A'_{jk} = \sin \theta A_{ik} + \cos \theta A_{jk} .$$

Ce calcul de A'_{ik} et A'_{jk} nécessite quatre multiplications et deux additions . Mais on peut aussi écrire :

$$A'_{ik} = \cos \theta (A_{ik} + A_{jk}) - (\sin \theta + \cos \theta)A_{jk} . ,$$

$$A'_{jk} = \cos \theta (A_{ik} + A_{jk}) + (\sin \theta - \cos \theta)A_{ik} .$$

Sous cette forme, on voit que le calcul de A'_{ik} et de A'_{jk} peut se faire avec trois multiplications seulement.

On écrira donc :

$$S_1 = \sin \theta + \cos \theta$$

$$S_2 = \sin \theta - \cos \theta$$

$$P_1 = \cos \theta (A_{ik} + A_{jk})$$

$$A'_{ik} = P_1 - S_1 A_{jk}$$

$$A'_{jk} = P_1 + S_2 A_{ik}$$

Pour calculer les $2n$ quantités A'_{ik} , A'_{jk} ($k=1, \dots, n$) on voit qu'il faudra avec ces formules :

$3n$ multiplications et $3n+2$ additions-soustractions.

Cette modification est intéressante dès lors que le temps d'exécution de la multiplication est supérieur à celui d'une addition. Elle entraîne une diminution d'un quart du nombre des multiplications employées par les algorithmes précédemment cités (le nombre total d'opérations arithmétiques augmentant du double du nombre de produits effectués par des matrices de rotation).

REFERENCES SUR LE CHAPITRE C II

- (8) AKAIKE, H., "Block Toeplitz matrix inversion".
SIAM J. Appl. Math. Vol. 24, N° 2, (march 1973).
- (9) BAREISS, E.H., "Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices".
Numer. Math. 13 (1969), 404-424.
- (10) BUNCH J., HOPCROFT JE., "Triangular factorization and inversion by fast matrix multiplication".
Math. Comp., 28 : 125 (1974), 231-236.
- (11) LAFON, J.C., "Quelques applications de l'étude du rang tensoriel d'un ensemble de matrices".
Séminaire d'Analyse Numérique, Grenoble, N° 166, (janvier 1973).
- (12) RISSANEN, J. "Solution of linear equations with Hankel and Toeplitz matrices". Numer. Math. 22, (1974) 361-366.
- (13) SHONHAGE, A., "Unitäre transformationen großer Matrizen".
Numer. Math. 20 (1973).
- (14) SHONHAGE, A., "Fast schmidt orthogonalization and unitary transformations of large matrices".
Proc. symposium Carnegie Mellon University (May 16-18 1973)
ed. by TRAUB "Complexity of sequential and parallel numerical algorithms"; 283-291.
- (14') STRASSEN, V., "Gaussien elimination is not optimal".
Numerisch Math. 13 (1969), 354-356.
- (15) ZOHAR, S., "The solution of a Toeplitz set of linear equations".
J. ACM vol 21, N° 2, (april 1974), 272-276.
- (16) WATSON, G.A., " An algorithm for the inversion of block matrices of Toeplitz form". J. ACM vol 20, N° 3, (july 1973), 409-415.

CHAPITRE CIII

OPTIMALITE DE QUELQUES

ALGORITHMES BASES SUR L'EMPLOI

DES MATRICES ELEMENTAIRES

PLAN

- 1 . Résolution d'un système linéaire
 - a/ Algorithmes optimaux
 - b/ Optimalité de Gauss pour l'obtention d'un système triangulaire
 - c/ Algorithmes optimaux de résolution d'un système linéaire
 - 2 . Transmutation d'une matrice sous forme d'Hessenberg, tridiagonale, et de Frobénius.
 - a/ Algorithmes optimaux
 - b/ Transmutations par des matrices élémentaires
 - c/ Méthode optimale d'obtention de la forme d'Hessenberg
 - d/ Méthode optimale de réduction sous forme tridiagonale
 - e/ Méthode optimale de réduction sous forme Frobénius.
- Références.

INTRODUCTION

Dans tout ce chapitre, on ne considère qu'une classe restreinte d'algorithmes de calculs sur les matrices : celle constituée des algorithmes qui n'utilisent que les opérations $+$, $-$, \times et $/$ du corps K et qui ne modifient une matrice qu'en la multipliant par des matrices élémentaires.

Dans cette classe restreinte d'algorithmes, on étudie la complexité, au point de vue du nombre de multiplications-divisions (ou du nombre d'additions-soustractions) utilisées, des calculs suivants :

triangularisation d'une matrice, résolution d'un système linéaire, obtention d'une matrice de forme d'Hessenberg, (ou tridiagonale, ou de Frobenius) semblable à la matrice initiale.

On sait que l'on peut effectivement résoudre ces calculs en n'utilisant que les opérations $+$, $-$, \times et $/$ du corps K et que des produits par des matrices élémentaires. Les méthodes basées sur l'emploi des matrices de rotation et celles qui nécessitent le calcul d'une racine carrée n'entrent pas dans cette classe d'algorithmes.

On montre que la méthode de Gauss usuelle est la méthode optimale d'obtention de la forme triangulaire d'un système linéaire.

En supposant que tous les algorithmes considérés conservent les zéros obtenus au cours des calculs, on démontre l'optimalité de Gauss pour la résolution complète du système. On décrit tous les algorithmes qui utilisent le même nombre d'opérations arithmétiques que Gauss (et qui sont donc aussi optimaux). En particulier, on montre qu'il n'est pas nécessaire de passer par l'étape intermédiaire de la triangularisation du système pour le résoudre avec le nombre minimal d'opérations arithmétiques - ce qui contredit le résultat obtenu par Klyuyev et Kokovkin-Shcherback sur ce problème. Notons que ces derniers ont été les premiers à poser dans un cadre analogue, le problème de l'optimalité de Gauss pour la résolution d'un système linéaire

On donne ensuite les algorithmes optimaux d'obtention des formes d'Hessenberg, tridiagonale et de Frobénius semblables à une matrice donnée. Pour l'obtention des formes tridiagonales et de Frobénius, les algorithmes obtenus diffèrent des algorithmes classiques. Le gain obtenu sur ces algorithmes classiques est de l'ordre de n^2 opérations arithmétiques. (Le nombre total d'opérations arithmétiques reste de l'ordre de n^3 opérations). Tous ces résultats ont déjà été exposés dans deux séminaires (18), (19).

1. RESOLUTION D'UN SYSTEME LINEAIRE

Soit un système de n équations linéaires à n inconnues :

$$(1) \quad A X = B ,$$

$A \in \mathcal{M}_{n,n}(K)$ (K corps infini), $X, B \in K^n$.

On note par a_{ij} l'élément i,j de la matrice A .

On associe à ce système la matrice A_1 suivante :

$$A_1 = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{pmatrix} \quad A_1 \in \mathcal{M}_{n,n+1}(K).$$

Définition 1

On obtient la forme triangulaire supérieure équivalente au système (1) si, par des combinaisons linéaires de ses lignes et des opérations rationnelles sur K on a mis la matrice A_1 sous la forme suivante :

$$R_1 = \begin{pmatrix} a'_{11} & \dots & a'_{1n} & b'_1 \\ 0 & & & \\ 0 & \dots & 0 & a'_{nn} & b'_n \end{pmatrix} .$$

La solution du système (1) vérifie :

$$\begin{pmatrix} a'_{11} & \dots & a'_{1n} \\ & & \\ 0 & & a'_{nn} \end{pmatrix} X = B' \quad B'^t = (b'_1, \dots, b'_n) .$$

Définition 2

On obtient la forme diagonale (ou résolue) équivalente au système 1 si, par des combinaisons linéaires de ses lignes et des opérations rationnelles sur K , on a mis la matrice A_1 sous la forme :

$$D_1 = \begin{pmatrix} 1 & & & b_1'' \\ & \ddots & & \\ & & \ddots & 0 \\ 0 & & & 1 & b_n'' \end{pmatrix} .$$

La solution du système (1) sera alors donnée par la dernière colonne de la matrice D_1 .

a/ Algorithmes optimaux

On s'intéresse ici seulement au cas général, c'est-à-dire au cas où les éléments de la matrice A_1 donnée sont absolument quelconques (sans relations algébriques entre eux). Cela veut dire en particulier que la matrice A_1 est toujours supposée régulière.

On peut donc considérer les éléments a_{11}, \dots, a_{nn} , b_1, \dots, b_n comme des indéterminées. Les calculs effectués auront donc lieu dans

$K(a_{11}, \dots, a_{nn}, b_1, \dots, b_n)$ (on garde les mêmes symboles pour les opérations de $K(a_{11}, \dots, b_n)$ que ceux des opérations de K).

On désigne par \mathcal{A}_R (resp. par \mathcal{A}_D) l'ensemble des algorithmes permettant d'obtenir R_1 (resp. D_1) à partir de A_1 en effectuant une suite finie de produits à gauche par des matrices élémentaires et une suite finie de calculs dans $K(a_{11}, \dots, a_{nn}, b_1, \dots, b_n)$.

A tout algorithme de \mathcal{A}_R (resp. \mathcal{A}_D), \mathcal{A} , on associe un coût mesuré par le nombre de multiplications et de divisions qu'il utilise. On note par $C_R^X(n, \mathcal{A})$ ce coût (resp. par $C_D^X(n, \mathcal{A})$). Dans le cas où le coût serait mesuré par le nombre d'additions-soustractions utilisées par l'algorithme, le coût sera noté $C_R^+(n, \mathcal{A})$ ou $C_D^+(n, \mathcal{A})$.

On définit $C_R^X(n)$ et $C_D^X(n)$ par :

$$C_R^X(n) = \text{Min}_{\mathcal{A} \in \mathcal{A}_R} C^X(n, \mathcal{A}) ,$$

$$C_D^X(n) = \text{Min}_{\mathcal{A} \in \mathcal{A}_D} C^X(n, \mathcal{A}) .$$

Un algorithme \mathcal{A}^* de \mathcal{A}_R (resp. de \mathcal{A}_D) qui est tel que l'on ait $C_R^X(n) = C_R^X(n, \mathcal{A}^*)$ (resp. $C_D^X(n) = C_D^X(n, \mathcal{A}^*)$) est un algorithme optimal dont le coût est le coût minimal.

b/ Optimalité de Gauss pour l'obtention du système triangulaire

On va démontrer que la méthode de Gauss est optimale à la fois pour le nombre de multiplications-divisions et pour le nombre d'additions-soustractions utilisées par tous les algorithmes de \mathcal{A}_R .

On va en effet montrer que l'on a :

$$C_R^X(n) = \frac{1}{6} n(n+1)(2n+1) - n$$

$$C_R^+(n) = \frac{n(n^2-1)}{3} .$$

Lemme 1

/ Dans le cas général, pour obtenir la matrice R_1 à partir de la matrice A_1 il faut utiliser au moins $\frac{n(n-1)}{2}$ matrices du type $E_{ij}(a)$ avec $a \in K(a_{11}, \dots, a_{nn}, b_1, \dots, b_n)$. /

□ En effet, pour obtenir k zéros en utilisant k matrices $E_{i_k j_k}(a_k)$, il faut résoudre un système linéaire de k équations à k inconnues (les a_k ou des produits de certains d'entre eux).

L'obtention de plus de k zéros n'est possible que si le déterminant de cette matrice est nul, ce qui est impossible par hypothèse. Le résultat découle du fait qu'il faut obtenir $\frac{n(n-1)}{2}$ zéros pour avoir la forme R_1 . □

Définition

On désigne par \mathcal{A}_\circ la classe des algorithmes de \mathcal{A}_R qui permettent de calculer R_1 à partir de A_1 en utilisant le nombre minimum possible de matrices élémentaires et en obtenant un zéro après chaque produit.

Théorème 1

/ Tous les algorithmes de la classe \mathcal{A}_\circ sont optimaux pour le nombre de multiplications-divisions (resp. additions-soustractions) employées. /

- Ce résultat prouve que la méthode de Gauss est optimale puisqu'elle fait partie de la classe \mathcal{A}_\oplus .

On va montrer que, pour un algorithme de \mathcal{A}_\oplus , l'ordre d'obtention des zéros est très particulier.

Soient $A_1^1, \dots, A_1^K = R_1$ les K matrices obtenues à partir de A_1 après chaque produit par une matrice élémentaires ($K = \frac{n(n-1)}{2}$).

Proposition 1

Si $k > j$, un algorithme de \mathcal{A}_\oplus ne peut placer un zéro en (i, k) avant d'en avoir obtenu un en (i, j) .

Il faut remarquer tout d'abord que tous les zéros créés par un algorithme \mathcal{A} de \mathcal{A}_\oplus sont conservés lors des multiplications restantes car un tel algorithme ne peut obtenir au plus que $\frac{n(n-1)}{2}$ zéros.

Supposons que A_1^q diffère de A_1^{q-1} par un zéro supplémentaire en position (i, k) et que l'élément (i, j) ($j < k$) ne soit pas encore nul. ($k < i$ sinon $\mathcal{A} \notin \mathcal{A}_\oplus$). A une étape ultérieure, il nous faudra bien mettre un zéro en (i, j) . Comme il faudra conserver le zéro en (i, k) pour que $\mathcal{A} \in \mathcal{A}_\oplus$, il faudra combiner la ligne i avec une autre ligne ayant aussi un zéro en (i, k) et l'élément (i, j) non nul. Il est clair que l'on peut alors refaire le même raisonnement pour cette ligne. D'autre part, on doit avoir $i > k$, car sinon on utiliserait un produit élémentaire pour placer un zéro au dessus ou sur la diagonale et l'algorithme \mathcal{A} utiliserait plus de $\frac{n(n-1)}{2}$ produits par des matrices élémentaires.

En réitérant le raisonnement, on voit que l'on aboutira forcément à une impossibilité.

Corollaire 1

Un algorithme de \mathcal{A}_\oplus ne pourra mettre un zéro en (i, k) . ($i > k$) que si les zéros en $(i, 1), (i, 2), \dots, (i, k-1)$ ont déjà été obtenus.

Corollaire 2

Le nombre minimal pour le cas général, d'opérations m/d (resp. d'opérations \dagger) pour les algorithmes de \mathcal{A}_\oplus est :

$$\frac{1}{6} n(n+1)(2n+1) - n \quad (\text{resp. } \frac{n(n^2-1)}{3}) .$$

En effet, pour obtenir un zéro en (i,k) il faut combiner la ligne i avec la ligne j ayant un élément non nul en (j,k) et les éléments $(j,1), (j,2), \dots, (j,k-1)$ nuls.

On calculera $\frac{a_{i,k}}{a_{j,k}}$ et $a_{i,q} - \frac{a_{i,k}}{a_{j,k}} a_{j,q}$ $q=k+1, \dots, n+1$

Le nombre d'opérations est donc :

1 division + $n+1-k$ multiplications et $n+1-k$ opérations $+$.

Le nombre total de multi-divisions sera donc :

$$\sum_{k=1}^{n-1} (n-k)(n+2-k) = \frac{1}{6} n(n+1)(2n+1) - n.$$

Remarque

L'algorithme de Gauss pour l'obtention du système triangulaire donne ce nombre d'opérations. \square

Le résultat précédent est valable seulement pour les algorithmes de \mathcal{A}_{\oplus} .

C'est-à-dire pour ceux qui satisfont aux trois conditions suivantes :

- L'algorithme ne place pas de zéros en (i,j) pour $j \geq i$.
- L'algorithme conserve les zéros déjà obtenus.
- Il n'obtient qu'un zéro au plus pour un produit élémentaire.

On va montrer que le résultat précédent est encore valable si l'on supprime ces trois restrictions.

Théorème 1'

- / L'algorithme optimal d'obtention de R_1 à partir de A_1 ne se trouve pas parmi les algorithmes de \mathcal{A}_R qui obtiennent au moins un zéro supplémentaire au-dessus de la diagonale de A_1 (tout en conservant les zéros créés). /
- \square Si un tel algorithme crée des zéros dans la partie supérieure de A_1 , tout en respectant l'ordre de création des zéros de la partie inférieure que suivent les algorithmes de \mathcal{A}_{\oplus} , alors son coût sera supérieur à celui d'un algorithme de \mathcal{A}_{\oplus} : il suffit de voir que pour chaque zéro ainsi créé, dans la partie supérieure, le coût de sa création est supérieur aux économies d'opérations qu'il permet de réaliser.

Soit (i,j) ($j > i$) la position du premier zéro créé au-dessus de la diagonale de A_1 par l'algorithme considéré. On va calculer la variation δ_{ij} entraînée par cette création sur le coût de la méthode de Gauss. On a :

$$\delta_{ij} = C_{ij} - G_{ij} ,$$

où C_{ij} est le coût de création minimal de ce zéro, et G_{ij} est la diminution maximale du nombre de multiplications-divisions entraînée par cette création (par rapport à Gauss).

Soit k_{ij} le nombre maximal de zéros déjà créés dans une ligne q ($q \neq i$) de la matrice A_1 dont l'élément (q,j) est non nul ($k_{ij} < j$). En combinant cette ligne q avec la ligne i pour créer le zéro en position (i,j) on aura le coût suivant (en nombre de multiplications-divisions) :

$$C_{ij} = n+1-k_{ij} .$$

Evaluons maintenant le gain escompté par rapport à Gauss. A chaque utilisation ultérieure de la ligne i pour créer un zéro dans la colonne i , on utilisera une multiplication de moins que pour Gauss. On aura donc :

$$G_{ij} = n-i \quad \text{si} \quad k_{ij} < i$$

$$G_{ij} = n-k_{ij} \quad \text{si} \quad k_{ij} \geq i$$

On a finalement :

$$\delta_{ij} = 1 + (i-k_{ij}) \quad \text{si} \quad k < i$$

$$\delta_{ij} = +2 \quad \text{si} \quad k \geq i$$

Dans les deux cas la variation de coût entraîné est positive et le coût total va donc augmenter.

Qu'en est-il pour les zéros ultérieurement créés dans la partie supérieure ? Le raisonnement est le même que celui précédemment fait sauf dans les deux cas suivants :

Cas 1 : On crée un zéro en (ℓ, i) ($\ell < i$) en se servant de la ligne i qui possède un zéro en (i, j) ($j > i$).

Cas 2 : On crée un zéro en (ℓ, j) ($j > \ell$) après en avoir créé un en (i, j) .

Cas 1 :

En effet, dans le cas où l'on crée un zéro en (ℓ, i) avec $\ell < i$, on peut diminuer de un le coût de création de ce zéro en se servant de la ligne i où l'on a déjà créé le zéro (i, j) . Mais pour que ce zéro (ℓ, i) puisse servir à quelque chose il faut évidemment que la colonne ℓ possède des éléments non nuls en-dessous de la diagonale. En particulier, pour que l'on puisse avoir $\delta_i = 1$ on doit avoir $k_{ij} < \ell$ (et donc $k_{ij} < i$) si bien que l'on aura toujours :

$$\delta_{\ell i} + \delta_{ij} \geq 4.$$

Cas 2

Dans le cas où l'on crée un zéro en (ℓ, j) ($j > \ell$) après en avoir créé un en (i, j) , il faut remarquer que l'on aurait eu intérêt à mettre un 1 en (j, j) si $k_{ij} = j-1$. En effet, le coût de cette opération aurait été de :

$$n+1-j \text{ divisions}$$

et le gain apporté de : $n-j$ multiplications pour la création des zéros de la colonne j en-dessous de la diagonale et de deux pour la création des zéros (i, j) et (i, k) .

Seulement, dans ce cas on aura $k_{ij} = j-1$ et $k_{\ell j} = j-1$ et donc :

$$\delta_{ij} = 2 \quad \text{et} \quad \delta_{\ell j} = 1$$

Le gain escompté reste donc toujours positif.

Examinons maintenant le cas où un tel algorithme ne suivrait pas l'ordre d'obtention des zéros de la partie inférieure de A_1 que suivent les algorithmes de \mathcal{A}_\oplus . Autrement dit peut-on avoir intérêt à créer un zéro en (i, k) ($k < i$) avec l'élément (i, j) ($j < k$) non nul ? Cela est possible si on se permet de créer des zéros au-dessus de la diagonale de A_1 . En effet, on pourra ultérieurement mettre un zéro en (i, j) et conserver le zéro en (i, k) si auparavant on a créé un zéro en (j, k) .

Pour montrer qu'on ne peut ainsi espérer gagner sur la méthode de Gauss, on peut simplement remarquer que le coût de création du zéro (i,k) sera le même que celui d'un zéro (j,k) dans la partie supérieure, et que le gain apporté par le zéro (i,k) sera le même que celui apporté par le zéro créé dans la partie supérieure. Le raisonnement précédemment fait montre donc que la variation de coût total sera toujours positive, c'est à-dire qu'on augmentera le coût de la méthode par rapport à celui d'un algorithme de \mathcal{A}_\oplus . \square

On peut montrer par un raisonnement analogue que l'on n'a pas intérêt à faire disparaître un zéro déjà créé, ou à faire k produits successifs par des matrices élémentaires avant d'obtenir q ($q \leq k$) zéros en même temps.

c/ Algorithmes optimaux de résolution d'un système linéaire.

Avec les mêmes définitions que précédemment, on va étudier l'obtention de D_1 à partir de A_1 dans le cas général.

On va tout d'abord montrer que l'algorithme d'élimination de Gauss, s'il est optimal, n'est pas le seul, et montrer en même temps que l'algorithme optimal peut ne pas passer par la forme R_1 .

On va décrire un algorithme qui vérifie ces deux propriétés.

c1) Description de l'algorithme

Pour $k = 1, 2, \dots, n$ on fera :

$$a_{k,j} = 0 \text{ pour } j=1, \dots, k-1$$

$$a_{k,j} = a_{k,j} - \sum_{i=1}^{k-1} a_{k,i} \cdot a_{i,j} \quad j=k, \dots, n+1$$

puis

$$a_{k,j} = a_{k,j} / a_{k,k} \quad j=k+1, \dots, n+1$$

enfin

$$a_{j,k} = 0 \quad j=1, \dots, k-1$$

$$a_{j,q} = a_{j,q} - a_{j,k} \times a_{k,q} \quad \begin{array}{l} j=1, \dots, k-1 \\ q=k+1, \dots, n+1 \end{array}$$

Après le pas k , on obtiendra une matrice de la forme suivante :

$$k \rightarrow \begin{pmatrix} 1 & 0 & \dots & 0 & \overset{k}{0} & x & x \\ & 0 & & & & & \\ & & & & 0 & & \\ \hline & 0 & 0 & & 1 & & \\ & x & x & & & & \\ & & & & & & \\ & x & & & & & x \end{pmatrix}$$

c2) Nombre d'opérations m/d

$$N = \sum_{k=1}^n (n+1-k) + 2(k-1)(n+1-k) + k-1$$

$$N = \frac{n^3}{3} + n^2 - \frac{n}{3}.$$

Ce qui est exactement le nombre de m/d utilisées par la méthode de Gauss (Il en est de même du nombre d'additions et de soustractions).

c3) Algorithmes optimaux

Dans ce paragraphe, on va montrer l'optimalité de la méthode de Gauss et celle de l'algorithme précédemment décrit parmi la classe des algorithmes \mathcal{A}_\oplus , c'est-à-dire parmi les algorithmes qui transforment A_1 en D_1 en obtenant un nouveau zéro après chaque produit par une matrice élémentaire et qui conservent ensuite le zéro obtenu.

On va tout d'abord montrer que si l'algorithme de \mathcal{A}_\oplus considéré conduit d'abord à une matrice R_1 alors il ne peut être meilleur que la méthode de Gauss.

Lemme 1

/ La méthode de Gauss est la méthode optimale pour résoudre un système linéaire si on doit d'abord triangulariser le système. /

- Comme on sait que cette méthode est optimale pour transformer le système sous la forme triangulaire (passage de A_1 à R_1), il suffit de montrer qu'ensuite on ne peut obtenir D_1 à partir de R_1 en moins de $\frac{n(n+1)}{2}$ multiplications-divisions. Ceci est évident dans le cas général car alors pour obtenir les $\frac{n(n-1)}{2}$ zéros dans la partie supérieure de la matrice il faut au moins $\frac{n(n-1)}{2}$ multiplications-divisions, et il en faut aussi au moins n supplémentaires pour faire apparaître une diagonale unité. □

Comparons maintenant la méthode de Gauss avec les algorithmes de \mathcal{A}_\oplus qui transforment A_1 en D_1 sans passer par l'étape intermédiaire R_1 . Cela veut dire qu'un tel algorithme va placer des zéros dans la partie supérieure à la diagonale avant d'avoir rempli de zéros la partie inférieure à cette diagonale. Dans quel cas peut-on espérer gagner ainsi par rapport à la méthode de Gauss ?

Théorème 2

- / Les seuls algorithmes de \mathcal{A}_\oplus qui utilisent le même nombre d'opérations arithmétiques des deux types que Gauss sont ceux qui pour obtenir un zéro en (i,j) avec $(j > i)$ l'obtiennent en combinant la ligne i avec la ligne j quand les éléments (j,k) ($k < j$) de celle-ci sont nuls et l'élément (j,j) égal à un. /

- Soit (i,j) la position du premier zéro mis au-dessus de la diagonale par un algorithme de \mathcal{A}_\oplus .

Soit $C(i,j)$ le coût d'obtention de ce zéro.

Si la ligne j a $j-1-x$ éléments nuls, alors le coût minimal d'obtention de ce zéro sera :

$$C(i,j) = n+1 - (j-1-x) \quad \text{si } x > 0$$

$$\text{Soit } C(i,j) = n+2-j+x \quad \text{pour } x > 0.$$

Dans le cas où $x=0$, on a intérêt à mettre d'abord un en position (j,j) . En effet cette opération va coûter $n-j$ divisions, mais elle va permettre ensuite d'économiser $n-j$ divisions pour obtenir les zéros en-dessous de la diagonales et 1 supplémentaire pour l'obtention du zéro (i,j) .

$$\text{Donc } C(i,j) = n+1-j \quad \text{si } x = 0.$$

Le coût par l'algorithme de Gauss aurait été :

$$C(i,j) = 1.$$

L'introduction de ce zéro au-dessus de la diagonale peut évidemment permettre de diminuer le coût d'obtention des éléments non nuls de la colonne i .

Le gain possible est :

$$G_1 = n-i \quad \text{si} \quad x > j-i$$

$$G_1 = n-j+x \quad \text{si} \quad x \geq 0$$

Le coût supplémentaire (par rapport à Gauss) de l'obtention du zéro en (i,j) est $C(i,j)-1$. Le gain possible est G_1 pour que cette méthode soit meilleure que Gauss il faut donc que l'on ait :

$$G_1 - C(i,j) + 1 \geq 0$$

pour $x > j-i$ on a :

$$\begin{aligned} G_1 - C(i,j) + 1 &= n-i-(n+1-j+x) \\ &= j-i-x-1 \end{aligned}$$

soit $G_1 - C(i,j) + 1 < 0$

pour $x < j-i$ mais $x \neq 0$ on a :

$$\begin{aligned} G_1 - C(i,j) + 1 &= n-j+x-(n+1-j+x) \\ &= -1 \end{aligned}$$

pour $x = 0$ on obtient :

$$G_1 - C(i,j) + 1 = 0 .$$

Ceci signifie qu'on obtient pour $x = 0$, le même nombre d'opérations que par Gauss. \square

2 . TRANSMUTATION D'UNE MATRICE SOUS FORME D'HESSENBERG, TRIDIAGONALE ET DE FROBENIUS

Dans ce paragraphe, on va étudier la complexité de l'obtention de l'une des trois formes canoniques semblables à une matrice A quelconque de $M_{n,n}(K)$ quand on se restreint à n'utiliser que des transmutations par des matrices élémentaires et les opérations $+$, $-$, \times et $/$ de K .

a/ Algorithmes optimaux

On considère les éléments a_{11}, \dots, a_{nn} de la matrice A comme des indéterminées. Tous les calculs sont donc effectués dans $K(a_{11}, \dots, a_{nn})$. (On note également par $+$, $-$, \times et $/$ les opérations de $K(a_{11}, \dots, a_{nn})$. Comme dans le paragraphe 1, on ne compte pas les opérations $a-a$, $a \neq 0$, et a/b quand $a = b$, $a = 0$ ou $b = 1$.

On désigne par \mathcal{A}_H (resp. par \mathcal{A}_T , et \mathcal{A}_F) la classe des algorithmes qui transforme une matrice A quelconque de $M_{n,n}(K)$ en une matrice de forme Hessenberg supérieure (resp. tridiagonale, et de Frobenius) à l'aide de transmutation par des matrices élémentaires et en n'utilisant que les opérations $+$, $-$, \times , $/$ de $K(a_{11}, \dots, a_{nn})$.

On note par $C_H^X(n)$ (resp. par $C_T^X(n)$ et $C_F^X(n)$) le nombre minimal de multiplications-divisions nécessaires pour obtenir la forme d'Hessenberg supérieure (resp. tridiagonale et de Frobenius).

$$C_H^X(n) = \min_{\mathcal{A} \in \mathcal{A}_H} C_H^X(\mathcal{A}, n),$$

$C_H^X(\mathcal{A}, n)$ désigne le nombre de multiplications-divisions employées par l'algorithme \mathcal{A} .

Un algorithme \mathcal{A} de \mathcal{A}_H dont le coût est égal à $C_H^X(n)$ sera dit optimal. Dans la suite, on va déterminer les algorithmes optimaux et les coûts minimaux $C_H^X(n)$, $C_T^X(n)$, $C_F^X(n)$.

On verra que ces algorithmes sont aussi ceux qui utilisent le moins d'opérations du type additions-soustractions. Dans la suite, les opérations multiplications-divisions seront désignées par opérations $*$.

b/ Transmutations par des matrices élémentaires

Soit $A' = E_i(a) A E_i(\frac{1}{a})$ A' ne diffère de la matrice A que par les éléments de la i ème ligne qui sont multipliés par a et ceux de la i ème colonne qui sont multipliés par $\frac{1}{a}$.

De même la matrice $E_{ij}(a) A E_{ij}(-a)$ ne diffère de la matrice A que par les éléments de la i ème ligne et ceux de la j ème colonne :

$$a'_{ik} = a_{ik} + a a_{jk} \quad k \neq j ,$$

$$a'_{qj} = a_{qj} - a a_{qi} \quad q \neq i ,$$

$$a'_{ij} = a_{ij} + a a_{jj} - a a_{ii} - a^2 a_{ji} .$$

Par une suite de transmutations par de telles matrices, on peut obtenir les formes d'Hessenberg, tridiagonale, et de Frobenius semblables à une matrice. On peut en effet à l'aide d'une transmutation par une matrice $E_{ij}(a)$, obtenir une matrice semblable à la matrice initiale mais avec un zéro dans une nouvelle position.

b1) Obtention d'un zéro par une transmutation élémentaire

Soit $E_{ij}(a)$ une matrice élémentaire telle que la matrice $E_{ij}(a) A E_{ij}(-a)$ ait un zéro en (p,q) ($A(p,q) \neq 0$).

On notera une telle matrice $E_{ij}(p,q)$.

On doit avoir, soit $i = p$, soit $j = q$.

1) Pour $i = p$ il faut choisir j tel que : $a_{jq} \neq 0$,

$$\text{et } a = - \frac{a_{pq}}{a_{jq}} .$$

2) Pour $j = q$ il faut choisir i tel que : $a_{pi} \neq 0$,

$$\text{et } a = \frac{a_{pq}}{a_{pi}} .$$

Cas particulier

Si $i = p$ et $j = q$ dans ce cas il faudrait déterminer a , solution de l'équation du second degré (si $a_{ji} \neq 0$) :

$$a^2 a_{ji} + a(a_{ii} - a_{jj}) + a_{ij} = 0$$

Ce qui est exclu pour les algorithmes de \mathcal{A}_H (de \mathcal{A}_F et de \mathcal{A}_T).

Dans le cas où l'on a $a_{ji} = 0$, on peut obtenir un zéro en (i,j) en utilisant

la transmutation par la matrice $E_{ij} \left(\frac{a_{ij}}{a_{jj} - a_{ii}} \right)$.

b2) Méthode optimale d'obtention d'un zéro

On désigne par $n_{i.}$ le nombre de zéros se trouvant déjà dans la ligne i et par $n_{.i}$ le nombre de zéros se trouvant déjà dans la colonne i .

$n_{(ij)}$: nombre de zéros des lignes i et j ayant même indice de colonne.

Soit j' l'indice de ligne tel que : $n_{j'} = \max_j n_{j.}$ avec $a_{j'q} \neq 0$

j'' l'indice de colonne tel que : $n_{.j''} = \max_j n_{.j}$ avec $a_{pj''} \neq 0$.

Au point de vue du nombre d'opérations $*$, la politique optimale pour l'obtention du zéro (p,q) sera de faire la transmutation pour $E_{p,j'}$ si $n_{j'} > n_{j''}$, par $E_{j''q}$ si $n_{.j''} > n_{j'}$.

Pour le nombre d'opérations \pm il faut considérer les indices j' et j'' réalisant les maximums de $n_{j.} + n_{.j} - (n_{(j,p)} + n_{.(j,p)})$ (pour j') et de $n_{j.} + n_{.j} - (n_{(j,q)} + n_{.(j,q)})$ pour j'' , les indices j étant ceux $a_{jq} \neq 0$ pour j' , $a_{pj} \neq 0$ pour j'' .

c/ Méthode optimale d'obtention de la forme d'Hessenberg

Dans toute la suite on ne parlera que de la réduction sous la forme Hessenberg supérieure (raisonnement similaire pour le cas inférieure). Il s'agit d'obtenir à partir de A , par une suite de transmutations élémentaires, une matrice avec $\frac{(n-1)(n-2)}{2}$ zéros dans les positions (i,j) $i \geq 2$ $j < i-1$.

Hypothèse fondamentale

On suppose que la suite de transmutation est telle que tout zéro obtenu dans une des positions désirées est conservé ensuite.

On peut alors démontrer les trois propositions suivantes :

Proposition 1

Si la j ème colonne est remplie de zéro dans les positions voulues, alors en général, il faut qu'il en soit de même pour toutes les colonnes précédentes $(1, \dots, j-1)$.

Supposons en effet, que l'on obtienne au bout d'un certain nombre de transmutations, une matrice A' semblable à A et telle que :

$$a'_{kj} = 0 \quad \text{pour} \quad k \geq j+2 .$$

On va tout d'abord montrer que tous les éléments a'_{ik} ($k=1, \dots, j-1$ et $i \geq j+2$) doivent être nuls.

Supposons en effet qu'il en existe un non nul :

$$a'_{pq} \neq 0 \quad \text{pour} \quad p \geq j+2, \quad q \leq j-1 .$$

Alors il sera nécessaire de produire un zéro à la place (p,q) pour réduire A sous la forme désirée.

Soit $E_{ki}(p,q)$ la matrice élémentaire produisant ce zéro. Si $k = p$ il faut choisir $i \geq j+2$ pour, dans le cas général, être sûr de conserver les zéros de la colonne j .

Si $i = q$, il faut choisir $k \leq j+1$ pour la même raison.

Ceci veut dire qu'on ne pourra éliminer cet élément a'_{pq} qu'en utilisant un autre élément non nul $a'_{p,q}$, situé aussi dans le rectangle $p \geq j+2$, $q \in \{1, 2, \dots, j-1\}$.

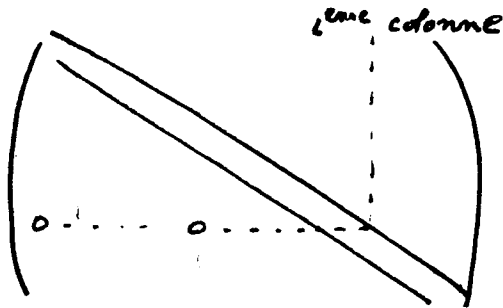
Comme on ne peut produire qu'un zéro à la fois pour une matrice A absolument quelconque, on voit qu'alors le dernier élément restant ne pourra pas être éliminé sans supprimer un zéro déjà produit.

Démontrons maintenant que tous les éléments a_{pq} avec $p \in \{3, j+1\}$ et $q \leq p-2$ doivent aussi être nuls.

Le raisonnement consiste à montrer comme précédemment, que le dernier élément de cette zone que l'on devrait remplacer par un zéro s'il n'en n'était pas ainsi, ne pourrait l'être qu'en supprimant un zéro déjà introduit. Ceci est immédiat à vérifier.

Proposition 2

Si la ligne i est remplie de zéro dans les positions voulues, alors en général, il faut qu'il en soit de même pour les lignes précédentes $(1, \dots, i-1)$.



On raisonne de la même façon que précédemment : s'il n'en n'était pas ainsi, le dernier élément de la zone

$$p \in \{2, \dots, i-1\}$$

$q \leq p-2$ ne pourrait être remplacé par un zéro sans supprimer un zéro déjà existant.

Proposition 3

On ne peut introduire un zéro dans la $i+1$ ème colonne avant d'avoir mis tous les zéros voulus dans les i premières.

Supposons que l'élément a_{pi} ($p \leq i+2$) soit différent de zéro. D'après la proposition 1 la i ème colonne ne peut être terminée avant les $i-1$ premières. Au cours de la réduction sous forme d'Hessenberg on se trouvera donc dans la situation suivante : un seul élément non nul dans la i ème colonne et que l'on doit remplacer par un zéro les $i-1$ èmes premières colonnes étant convenablement remplies.

Pour éliminer cet élément a_{pi} on doit combiner la p ème ligne (ou i ème colonne) avec une ligne (ou colonne) ayant la même distribution de zéros. Le seul cas possible est d'utiliser la $i+1$ ème ligne. Mais alors le zéro existant dans la $i+1$ ème colonne ne sera sauvegardé que si la p ème colonne a un zéro pareillement placé. Le même raisonnement pourrait alors s'appliquer à la p ème colonne ($p > i+1$), et l'on voit qu'on aboutira soit à une impossibilité (si $p < n$), soit au cas où la $n-2$ ème colonne serait terminée sans qu'une des précédentes le soit, ce qui est en contradiction avec la proposition (1).

En conclusion, il sera en général nécessaire, si l'on veut garder tous les zéros déjà introduits, de transmuter la matrice A de façon à obtenir d'abord les zéros voulus de la première colonne, puis ceux de la seconde..., jusqu'à ceux de la $n-2$ ème colonne.

Considérons alors l'algorithme suivant de réduction d'une matrice sous forme d'Hessenberg supérieure :

ième étape : obtention d'une matrice semblable à la matrice initiale avec les zéros voulus dans la ième colonne.

On procède ainsi :

si $a_{i+1,i} = 0$, alors on cherche un élément $a_{j,i}$ ($j \geq i+2$) non nul. S'il n'en existe pas, alors la colonne i possède déjà les zéros voulus, on passe donc à la $i+1$ ème étape. Sinon, on transmue par $E_{i+1,p}(1)$ si $a_{p,i} \neq 0$. Ensuite on effectuera dans un ordre quelconque les transmutations par

$$E_{p,i+1}\left(-\frac{a_{p,i}}{a_{i+1,i}}\right) \text{ pour } p = i+2, \dots, n.$$

On s'arrête après la $n-2$ ème colonne.

Calculons le nombre maximal d'opérations * nécessaires pour transmuier une matrice A sous forme d'Hessenberg supérieure par cette méthode :

$$\begin{aligned} C_H^X(n) &= \sum_{p=1}^{n-2} \left(\sum_{k=p+2}^n (n-n+1)+n \right) \\ &= \sum_{p=1}^{n-2} (n-p-1)(2n+1-p) \\ &= \sum_{p=1}^{n-2} 2n^2 - n - 3np + p^2 - 1. \end{aligned}$$

$$C_H^X(n) = \frac{5n^3}{6} - 2n^2 + \frac{n}{6} + 1.$$

D'autre part, les propositions 1, 2 et 3 montrent, que pour une matrice initiale absolument quelconque, le procédé est optimal.

Du point de vue du nombre d'opérations \dagger , il est clair qu'à chaque étape, il y a une opération \dagger de moins qu'il y a d'opérations * (celle correspon-

dant au calcul de $\frac{a_{p,i}}{a_{i+1,i}}$).

Donc :

$$C_H^+(n) = f(n) - \frac{(n-2)(n-1)}{2}$$

$$C_H^+(n) = \frac{5n^3}{6} - \frac{5n^2}{2} + \frac{5n}{3}$$

Le raisonnement précédemment fait montre que la méthode décrite en (I) est aussi optimale pour le nombre d'opérations \pm .

REMARQUE

La démonstration précédente n'est valable qu'en raisonnant sur les méthodes telles qu'à chaque transmutation élémentaire tous les zéros déjà obtenus aux places désirées soient conservés et qu'il y ait au plus un zéro supplémentaire.

On va maintenant examiner plus rapidement les réductions sous forme tridiagonale et de Frobenius.

d/ Méthode optimale de réduction sous forme tridiagonale

On conserve les hypothèses qui ont permis de conclure pour la forme d'Hessenberg.

1/ Obtention de la forme tridiagonale en passant par la forme d'Hessenberg :

On utilise d'abord la méthode optimale d'obtention de la forme d'Hessenberg (supérieure par exemple).

Cette étape nécessite donc :

$$\frac{5n^3}{6} - 2n^2 + \frac{n}{6} + 1 \text{ opérations } * .$$

On va maintenant chercher quelle est la meilleure méthode pour obtenir la forme tridiagonale à partir de la forme d'Hessenberg supérieure ainsi obtenue. Pour cela, on va tout d'abord caractériser les différentes politiques d'obtention des zéros dans la partie supérieure par les propositions suivantes

PROPOSITION 4

/ On ne peut obtenir une matrice semblable à la matrice initiale avec la colonne j convenablement remplie de zéros (resp. avec la ligne j remplie de zéros dans les positions désirées) sans que toutes les colonnes $3, 4, \dots, j-1$ (resp. les lignes $1, 2, \dots, j-1$) aient également des zéros dans toutes les positions voulues. /

Pour démontrer cette proposition, il suffit de constater que dans le cas contraire, on ne pourrait obtenir la forme tridiagonale tout en conservant les zéros déjà obtenus.

PROPOSITION 5

/ Le dernier zéro (i,k) de la ligne i ($k > i+1$) ne peut être obtenu avant les zéros en les positions $(i+1,k-1)$, $(i+1,k)$ et $(i+1,k+1)$ (si $k < n$). /

Cela provient du fait que pour obtenir ce zéro on doit utiliser la transmutation par la matrice :

$$E_{i+1,k} \left(\frac{a_{i,p}}{a_{i,i+1}} \right) .$$

On va maintenant comparer la méthode classique avec une deuxième méthode qui respecte aussi ces conditions.

PREMIER PROCEDE (méthode classique)

Pour obtenir une matrice avec des zéros convenablement placés dans la i ème ligne, on fera les transmutations par :

$$E_{i+1,p} \left(\frac{a_{i,p}}{a_{i,i+1}} \right) \quad p = i+2, \dots, n . \text{ On}$$

On s'arrête à la $n-2$ ème ligne.

Comptons le nombre maximal d'opérations * nécessaires :

$$T_1 = \sum_{p=1}^{n-2} \sum_{k=1}^{n-p-1} (3 + k + 1) .$$

$$T_1 = \sum_{p=1}^{n-2} 4(n-p-1) + \frac{(n-p-1)(n-p)}{2}$$

on trouve :

$$T_1 = \frac{n^3}{6} + \frac{3n^2}{2} - \frac{17n}{3} + 4 .$$

DEUXIEME PROCEDE

Pour obtenir un zéro en (i,k) dans la colonne i ($i=3,4,\dots,n-1$), on fera la transmutation par la matrice :

$$E_{k,k+1} \left(- \frac{a_{k,i}}{a_{i+1,i}} \right) \quad \text{pour } k > i-1 \quad \text{et } i+k \leq n .$$

On économise ainsi deux opérations * pour l'obtention des zéros de la première ligne et une opération * pour l'obtention des autres zéros (par rapport à la première méthode).

Les zéros restant à obtenir seront obtenus comme dans la première méthode :

Pour obtenir le zéro (i,k) de la ligne i ($i=1,\dots,n-2$) pour $k > i+1$ et $i+k > n$ on fera la transmutation par la matrice :

$$E_{i+1,k} \left(\frac{a_{i,k}}{a_{i,i+1}} \right).$$

Le gain obtenu par rapport à la première méthode est égal à :

$$2(2p-2) + \sum_{j=0}^{p-3} (2j+1) \quad \text{pour } n=2p, \quad \text{soit } p^2 - 2$$

$$2(2p-1) + \sum_{j=1}^{p-2} 2j \quad \text{pour } n=2p+1, \quad \text{soit } p^2 + p - 2$$

Le coût total est donc : e

$$T_2 = \frac{n^3}{6} + \frac{3}{2}n^2 - \frac{17n}{3} + 4 - \begin{cases} p^2 - 2 & \text{si } n=2p \\ p^2 + p - 2 & \text{si } n=2p+1 \\ +2 & \text{si } n=3 \end{cases}$$

REMARQUE

Pour ces deux méthodes on a supposé implicitement que l'on avait toujours $a_{i,i+1} \neq 0$ ($i=1,\dots,n-2$) et $a_{i+1,i} \neq 0$ ($i=3,\dots,n$).

Si $a_{i,i+1} = 0$ et si l'un des deux éléments $a_{i+1,i+1}$ et $a_{i+2,i+1}$ est différent de zéro il suffit de faire la transmutation par $E_{i,i+1}^{(1)}$ (ou par $E_{i,i+2}^{(1)}$) pour se ramener au cas $a_{i,i+1} \neq 0$.

Si $a_{i+1,i} = 0$ et si aucun des $a_{j,j+1}$ n'est nul pour $j \leq i-2$ alors on pourra appliquer la méthode 1 au lieu de la méthode 2 pour les éléments sur la ième colonne $((1,i), (2,i), \dots, (i-2,i))$.

Comme le fait d'avoir $a_{i+1,i} = 0$ signifie que dans la réduction sous forme d'Hessenberg supérieure on a obtenu d'un seul coup la ième colonne avec tous ses zéros, on n'a jamais plus de T_2 opérations *.

On peut d'ailleurs pour éviter ces difficultés, s'imposer d'obtenir les éléments $(i+1,i)$ différents de zéro au cours de la réduction sous forme Hessenberg supérieure.

Toutes les autres méthodes donneraient un nombre plus élevé d'opérations \star . En effet, on peut aisément constater que si on obtient un zéro en (i,k) $k > i+1$ et $i+k \leq n$ sans que tous les éléments (q,j) soient nuls ($\forall j < i$ et $\forall q > j+1$), alors le coût supplémentaire d'obtention de ce zéro (par rapport à la méthode 2) est supérieur au gain d'opérations \star apporté pour la création des zéros restants à obtenir dans la partie inférieure.

La méthode 2 (et celle de l'exemple 1) sont donc optimales et le coût minimal d'obtention de la forme tridiagonale est :

$$C_T^X(n) = n^3 - \frac{n^2}{2} - \frac{11n}{3} + 5 - \begin{cases} p^2-2 & \text{si } n=2p \\ p^2+p-2 & \text{si } n=2p+1 \\ -2 & \text{si } n=3 \end{cases}$$

Comparaison avec d'autres méthodes

- La méthode d'Householder de tridiagonalisation par des transmutations à l'aide de matrices du second degré nécessite un nombre d'opérations \star de l'ordre de $\frac{5n^3}{3}$.
- On peut obtenir les zéros de la forme tridiagonale dans le même ordre que dans la méthode d'Householder, en effectuant des transmutations par des matrices élémentaires.

Une telle méthode donne un nombre d'opérations de l'ordre de $\frac{4n^3}{3}$.

e/ Méthode optimale de réduction sous forme de Frobénius

On ne refait pas en détail des raisonnements similaires à ceux déjà fait précédemment.

La méthode optimale (à la fois pour le nombre d'opérations multiplications-divisions et des opérations additions-soustractions), consiste d'abord à effectuer la réduction sous la forme d'Hessenberg supérieure, puis à faire les transmutations par les matrices suivantes :

$$\text{matrices } E_{i,p+1} \left(-\frac{a_{i,p}}{a_{p+1,p}} \right) \quad \text{pour } i=1,\dots,n-1 \text{ et } p=i,i+1,\dots,n-1$$

$$\text{matrices } E_i (a_{i,i-1}^{-1}) \quad \text{pour } i=2,\dots,n.$$

Le coût d'obtention de la forme de Frobénius à partir de la forme d'Hessenberg supérieure est égal à :

$$\sum_{j=2}^n (1+j)(n-j) + 2(n-2) + n-1 \text{ opérations } * ,$$

$$\frac{n^3}{6} + n^2 + \frac{11n}{6} - 5 \text{ opérations } * .$$

Le nombre total d'opérations multiplications-divisions nécessaires pour obtenir la forme de Frobénius semblable à une matrice quelconque est donc égal à :

$$n^3 - n^2 + 2n - 4 \quad (C_F^x(n)) .$$

Par rapport à la méthode de Danilewsky, on a un gain de l'ordre de $\frac{3}{2} n^2$ opérations * (multiplications-divisions).

REMARQUE

Si, pour une matrice particulière, on a :

$a_{p+1,p} = 0$, alors on ne fera pas les transmutations correspondantes. On obtiendra alors une forme de Frobénius générale. Cela n'est évidemment pas le cas quand les éléments de la matrice sont, comme on l'a supposé, sans relation algébrique entre eux.

REFERENCES SUR LE CHAPITRE CIII

- (17) KLYUYEV, VV., KOKOVKIN-SHCERBACK, N.I.
"On the minimization of the number of arithmetic operations
for the solution of linear algebraic systems of equations".
Shurnal Vychislitel'noi Matematiki, Matematicheskoi Fiziki,
5, n° 1, 21-33 , (1965).
also in Technical report cs 24, (june 14, 1965),
Stanford University.
- (18) LAFON, J.C. "Nombre minimum d'opérations nécessaires à l'obtention
des formes d'Hessenberg et tridiagonale d'une matrice".
Séminaire d'Analyse Numérique, Grenoble, (mai 1971) (n° 130).
- (19) LAFON, J.C. "Quelques résultats sur l'optimisation des algorithmes".
Séminaire d'Analyse Numérique, Grenoble, (19 janvier 1972), n° 145.

CHAPITRE CIV

UTILISATION OPTIMALE DU PARALLELISME

PLAN

- 1 . Produit de deux matrices
 - 2 . Inversion d'une matrice. Résolution
d'un système linéaire.
Calcul d'un déterminant.
 - 3 . Décomposition LR et QR
 - 4 . Obtention d'une forme canonique semblable
à une matrice.
 - 5 . Matrices de formes particulières.
- Références.

INTRODUCTION

Ce chapitre est consacré à l'étude des coûts des principaux algorithmes de calculs matriciels quand on suppose l'utilisation de plusieurs unités de calculs capables d'effectuer "en parallèle" n'importe quel type d'opérations arithmétiques.

Le coût d'un algorithme est alors mesuré par le maximum du nombre d'opérations exécutées par chaque unité de calcul. On suppose que toutes les données, ainsi que tous les résultats intermédiaires sont accessibles au même instant par chaque unité de calcul. On admet aussi que l'exécution de l'une des opérations $+$, $-$, \times , $/$, prend une unité de temps et qu'aucun problème de synchronisation des unités de calculs ne se pose. Dans ces conditions, le coût d'un algorithme dépendra du nombre k d'unités de calculs qui sont supposées travailler "en parallèle".

Soit \mathcal{A} un algorithme qui appartient à l'ensemble \mathcal{A}_P des algorithmes de calculs algébriques résolvant le problème P (cf. partie B).

\mathcal{A} est défini par l'ensemble ordonné I de ses instructions.

Une "implémentation k parallèle" de l'algorithme \mathcal{A} est par définition, une partition de l'ensemble I possédant les deux propriétés suivantes :

- Cette partition est constituée par des sous-ensembles de I d'au plus k instructions.
- Si q est le nombre de ces sous-ensembles, on peut les numéroter de 1 à q de telle manière que toutes les instructions du même groupe puissent être exécutées simultanément dès lors que les instructions des groupes 1 à $k-1$ sont exécutées.
- Le coût d'une telle implémentation sera alors égal à q . L'implémentation k parallèle optimale de l'algorithme \mathcal{A} sera celle de coût minimal.
- On note ce coût par :

$$T_k(\mathcal{A}).$$

Le coût de l'algorithme \mathcal{A} exécuté avec k unités de calculs fonctionnant en

parallèle sera pris égal à $T_k(\mathcal{A})$, coût de son implémentation k parallèle optimale. Si le nombre d'unités de calculs est arbitraire on notera le coût de \mathcal{A} par $T_\infty(\mathcal{A})$.

Il est en général plus difficile d'évaluer $T_k(\mathcal{A})$ que $T_\infty(\mathcal{A})$. Le coût $T_\infty(\mathcal{A})$ donne la limite de ce que l'on peut attendre de l'utilisation du parallélisme pour l'algorithme \mathcal{A} . La complexité du problème P , quand on utilise k unités de calculs est mesurée par :

$$C_k(P) = \min_{\mathcal{A} \in \mathcal{A}_P} T_k(\mathcal{A}) .$$

La détermination de $C_k(P)$, ou d'une minoration non triviale de ce nombre est un problème très difficile. Le cas particulier $k=1$, correspond au problème de la détermination du nombre total d'opérations arithmétiques nécessaires pour résoudre P . Les minorations des $C_k(P)$ proviennent en général d'une minoration de ce nombre d'opérations arithmétiques.

Dans la suite, on va étudier comment implémenter de façon optimale les principaux algorithmes de calculs matriciels, et évaluer leur coût quand le nombre d'unités de calculs est arbitraire. Les problèmes étudiés dans cette optique sont les suivants : produit de deux matrices. Inversion d'une matrice. Résolution d'un système linéaire. Calcul d'un déterminant. Décomposition LR et QR d'une matrice. Transmutation d'une matrice sous une forme canonique .

1 . PRODUIT DE DEUX MATRICES

Théorème 1

/ Le produit de deux matrices de $M_{n,n}(K)$ peut se faire, de manière optimales , en $\lceil \log_2 n \rceil + 1$ unités de temps si on utilise au moins n^3 unités de calcul . /

□ Soient A, B, C trois matrices de $M_{n,n}(K)$.

Si $C = AB$ on a :

$$C_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

On peut calculer les n éléments C_{ij} en parallèle en utilisant pour le calcul de chacun d'eux n unités de calcul. Il faudra donc une unité de temps pour calculer les produits $a_{ik} b_{kj}$ ($k=1, \dots, n$) puis $\lceil \log_2 n \rceil$ unités de temps pour en faire la somme.

L'optimalité du résultat provient du fait que WINOGRAD (27) a démontré que le produit scalaire de deux vecteurs de K^n nécessitait exactement n produits et $n-1$ additions, (et il faut $\lceil \log_2 n \rceil$ unités de temps pour faire la somme de n nombres en parallèle). □

Remarque

Le résultat précédent montre que si l'on a au moins n^3 unités de calcul la méthode usuelle de calcul du produit de deux matrices donne le coût minimal $\lceil \log_2 n \rceil + 1$.

Dans le cas de l'utilisation d'un nombre plus restreint d'unités de calcul la méthode usuelle peut ne pas correspondre au coût minimal. En particulier dans le cas d'une seule unité de calcul, la méthode de STRASSEN donne le résultat en $O(n^{\log_2 7})$ opérations arithmétiques seulement. On peut donc se poser la question suivante :

" Quel est le nombre k d'unités de calcul qu'il faut employer pour que la méthode classique du produit de deux matrices exécutée à l'aide de ces k unités soit plus rapide que la méthode de STRASSEN exécutée avec une seule unité de calcul ? "

Pour répondre à cette question, il faut estimer le coût de la méthode classique exécutée avec k unités de calcul en parallèle.

Théorème 1'

/ Si on exécute le produit de deux matrices à l'aide de k unités de calcul en parallèle, le coût de la méthode classique est :

$$n^2 \left(\lceil (2n - 2^{\lceil \log_2 k \rceil}) / k \rceil + \lceil \log k \rceil \right) \quad \text{si } k \leq n \quad /$$

- Le coût de la méthode classique sera égal à n^2 fois le coût de l'évaluation avec k unités de calcul du produit scalaire de deux éléments de K^n .

Soit q le coût du produit scalaire de deux vecteurs de K^n , quand on utilise k unités de calcul. De façon optimale on pourra faire :

$$2^{\lceil \log_2 k \rceil} - 1 + (q - \lceil \log_2 k \rceil) k \text{ opérations binaires}$$

Mais, d'après le résultat déjà cité de WINOGRAD, on doit faire n multiplications et $n-1$ additions. On a donc :

$$2^{\lceil \log_2 k \rceil} - 1 + (q - \lceil \log_2 k \rceil) k \geq 2n-1$$

et donc :

$$q \geq \lceil (2n - 2^{\lceil \log_2 k \rceil}) / k \rceil + \lceil \log_2 k \rceil . \quad \square$$

Ce résultat montre qu'il faudra employer $n^{1/5}$ unités de calcul au moins pour que la méthode classique exécutée en parallèle devienne plus rapide que la méthode de STRASSEN exécutée séquentiellement.

Les résultats précédents montrent que le produit de deux matrices est un calcul pour lequel l'introduction du parallélisme est très bénéfique. En est-il de même pour les autres calculs matriciels ? On va voir que la réponse à cette question est généralement affirmative.

2 . INVERSION D'UNE MATRICE. RESOLUTION D'UN SYSTEME LINEAIRE. CALCUL D'UN DETERMINANT.

Soit A une matrice quelconque de $M_{n,n}(K)$. On note par $I_\infty(n)$, $R_\infty(n)$, $D_\infty(n)$ le coût de l'inversion de A , de la résolution du système $AX = b$ et du calcul du déterminant de A quand on utilise un nombre arbitraire d'unités de calcul .

On a les relations évidentes suivantes :

$$(1) \quad R_{\infty}(n) \leq D_{\infty}(n) + 1 \quad (\text{on résoud le système par la règle de CRAMER})$$

$$I_{\infty}(n) \leq D_{\infty}(n-1) \quad (\text{on calcule l'adjointe de la matrice A}).$$

On peut démontrer le résultat suivant :

Théorème 2

/ Le calcul de l'inverse d'une matrice A de $M_{n,n}(K)$, la résolution d'un système linéaire $AX = b$ et le calcul du déterminant de A peuvent se faire en $O((\log_2 n)^2)$ unités de temps quand on utilise $O(n^4)$ unités de calcul . /

□ Supposons tout d'abord la matrice A triangulaire inférieure, On a alors évidemment :

$$D_{\infty}(A) = \lceil \log_2 n \rceil .$$

Si $n = 2^q$ on peut partitionner A en quatre sous-matrices de tailles 2^{q-1} :

$$A = \begin{pmatrix} A_1 & 0 \\ A_2 & A_3 \end{pmatrix} \quad A_1 \text{ et } A_3 \text{ sont aussi triangulaires inférieures de tailles } 2^{q-1}.$$

On a :

$$A^{-1} = \begin{pmatrix} A_1^{-1} & 0 \\ -A_3^{-1}A_2A_1^{-1} & A_3^{-1} \end{pmatrix} .$$

On a donc :

$$I_{\infty}(2^q) = I_{\infty}(2^{q-1}) + 2 M_{\infty}(2^q) .$$

D'après le lemme 1 on a :

$$M_{\infty}(2^q) = q + 1 .$$

On obtient ainsi

$$I_{\infty}(2^q) = q^2 + 3q + 1$$

Pour A triangulaire inférieure de taille n on a donc bien :

$$I(n) = O(\text{Log}^2 n).$$

Pour prouver le même résultat dans le cas d'une matrice quelconque on ne peut utiliser la même technique d'inversion d'une matrice par partitionnement, car dans ce cas on aurait une relation du type :

$$I_{\infty}(n) = 2 I_{\infty}\left(\frac{n}{2}\right) + 4 M_{\infty}\left(\frac{n}{2}\right),$$

et on obtiendrait donc seulement $I_{\infty}(n) = O(n)$.

On va ici utiliser le fait que la matrice A annule son polynome caractéristique (cf. propriété 6, chapitre I).

Soit $P(\lambda) = (-1)^n (\lambda^{n-P_1} \lambda^{n-1} \dots -P_n)$ le polynome caractéristique de A.

On a :

- 1) $\det A = (-1)^{n-1} P_n$
- 2) $A^{-1} = P_n^{-1} (A^{n-1} - P_1 A^{n-2} \dots - P_{n-1} I)$.

Si $\lambda_1, \dots, \lambda_n$ sont les n valeurs propres de A, on sait que l'on a :

$$\sum_{i=1}^n \lambda_i^k = \text{tr}(A^k) \quad (S_i = \sum_{i=1}^n \lambda_i^k).$$

Le calcul de S_1, \dots, S_n peut se faire en $\lceil \text{Log}_2 n \rceil \times (2 \lceil \text{Log}_2 n \rceil + 1)$ unités de temps.

Par utilisation des relations de Newton, on calcule ensuite les coefficients P_1, \dots, P_n en inversant une matrice triangulaire inférieure. D'après ce qui précède on peut donc déterminer $P(\lambda)$ en $O(\text{Log}^2 n)$ unités de temps. Les formules (1) et (2) montrent donc que le calcul du déterminant et de l'inverse peut aussi se faire en $O(\text{Log}^2 n)$.

Ce résultat a semble-t-il été observé pour la première fois par CSANSKY (21) (dans le contexte du calcul en parallèle). \square

Remarque

La méthode de Gauss en parallèle nécessite $O(n)$ unités de temps. Le calcul d'un déterminant par développement suivant une colonne (ou par toute autre application de la formule de LAPLACE) nécessite au moins $O(n)$ unités de temps. Enfin, on a vu que la méthode d'inversion par partitionnement nécessite aussi $O(n)$ unités de temps quand on l'exécute en parallèle.

3 . DECOMPOSITION LR ET QR EN PARALLELEThéorème 3

/ La décomposition LR d'une matrice de $M_{n,n}(K)$ peut se calculer en parallèle en $O(\log_2^3 n)$ unités de temps. /

Supposons $n = 2^k$. Décomposons la matrice M de $M_{n,n}(K)$ en quatre sous-matrices de taille 2^{k-1} :

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

et écrivons :

$$L = \begin{pmatrix} L_1 & 0 \\ C_1 & L_2 \end{pmatrix}, \quad R = \begin{pmatrix} R_1 & B_1 \\ 0 & R_2 \end{pmatrix}$$

L_1, L_2 sont deux matrices triangulaires inférieures à diagonales unités de taille 2^{k-1} et R_1, R_2 sont deux matrices triangulaires supérieures de taille 2^{k-1} .

On a :

$$\begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A & B \\ 0 & D-CA^{-1}B \end{pmatrix}.$$

Le coût de ce calcul est :

$$I_{\infty}(2^{k-1}) + 2 M_{\infty}(2^{k-1}) + 1.$$

On peut décomposer simultanément A et $D-CA^{-1}B$ sous la forme LR :

$$\begin{aligned} A &= L_1 R_1 \\ D-CA^{-1}B &= L_2 R_2 \end{aligned}$$

Le coût de ce calcul est $LR_{\infty}(2^{k-1})$.

On pourra écrire :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} L_1 & 0 \\ CA^{-1}L_1 & L_2 \end{pmatrix} \begin{pmatrix} R_1 & X \\ 0 & R_2 \end{pmatrix},$$

avec $X = L_1^{-1} B$.

Le coût total de cette décomposition sera donc :

$$LR_{\infty}(2^k) = LR_{\infty}(2^{k-1}) + 2 I_{\infty}(2^{k-1}) + 3 M_{\infty}(2^{k-1}) + 1$$

$$\Rightarrow LR_{\infty}(n) = O((\log_2 n)^3) \quad \square$$

Théorème 4

/ La décomposition QR d'une matrice de $\mathcal{M}_{n,n}(K)$ peut se calculer en parallèle en $O(n)$ unités de temps. /

□ On va montrer que l'on peut réorganiser la méthode de Givens de façon à obtenir cette décomposition en $O(n)$ unités de temps.

Supposons qu'en k unités de temps on puisse déterminer la matrice de rotation $V_{ij}(\theta)$ qui permet d'obtenir un zéro en (i,j) . (Si le calcul d'une racine carrée nécessite une unité de temps, on peut alors prendre $k = 4$).

Le produit de la matrice A de $\mathcal{M}_{n,n}(K)$ par la matrice $V_{ij}(\theta)$ peut se faire en deux unités de temps en parallèle si on dispose d'au moins $4n$ unités de calcul. En effet, on peut alors calculer simultanément toutes les quantités :

$$\begin{aligned} \cos \theta A_{ik} + \sin \theta A_{jk} & \quad (k=1, \dots, n) \\ \cos \theta A_{jk} - \sin \theta A_{ik} & \quad (k=1, \dots, n) \end{aligned}$$

En $k+2$ unités de temps on peut obtenir les zéros en position $(i,1)$,

$(i-1,2), \dots, \left(\left\lceil \frac{i}{2} \right\rceil, \left\lfloor \frac{i}{2} \right\rfloor\right)$:

En effet, on peut faire en même temps les multiplications de la matrice A par les matrices $V_{i1}(\theta_1), V_{i-1,2}(\theta_2), \dots, V_{\left\lceil \frac{i}{2} \right\rceil, \left\lfloor \frac{i}{2} \right\rfloor}(\theta_{\left\lfloor \frac{i}{2} \right\rfloor})$, puisque les lignes modifiées sont toutes différentes.

On obtiendra ainsi la matrice R en $(2n-3) \times (k+2)$ unités de temps si on a au moins n unités de calcul. La matrice Q peut ensuite être obtenue en faisant le produit des $2n-3$ matrices précédemment calculées, ce qui prend un temps inférieur à $\lceil \log_2 2n-3 \rceil \times \lceil \log_2 n \rceil$.

La décomposition QR de A est donc bien déterminée en $O(n)$ unités de temps. \square

Remarque

La méthode décrite dans la démonstration du théorème précédent n'est pas la méthode optimale d'obtention de la décomposition QR de la matrice A en se servant des matrices de rotation. Par exemple pour $n = 4$, on peut obtenir les zéros en $(2,1)$ et $(3,1)$ en même temps en multipliant A par les matrices $V_{21}(\theta_1)$ et $V_{43}(\theta'_1)$. Cependant le temps de calcul reste en $O(n)$.

4. OBTENTION D'UNE FORME CANONIQUE SEMBLABLE A UNE MATRICE

On ne traitera ici que du calcul de la forme d'Hessenberg inférieure semblable à une matrice A de $M_{n,n}(K)$. (Le raisonnement et le résultat restent les mêmes pour le cas du calcul de la forme tridiagonale ou de la forme de Frobenius).

On va donc ici simplement démontrer le théorème suivant :

Théorème 5

/ Le calcul en parallèle de la forme d'Hessenberg inférieure semblable à la matrice A de $M_{n,n}(K)$ peut se faire en $O(n \log n)$ unités de temps. /

□ Remarquons tout d'abord que la transmutation de A par une matrice élémentaire $E_{ij}(a)$ peut se faire en quatre unités de temps seulement :

$$\begin{aligned} A'_{ik} &= A_{ik} + a A_{jk} & k \neq j & \\ & & & \text{deux unités de temps} \\ A'_{kj} &= A_{kj} - a^2 A_{ki} & k \neq i & \\ A'_{ij} &= A_{ij} + a A_{ji} - a A_{ii} - a^2 A_{ji} & & \text{quatre unités de temps.} \end{aligned}$$

Pour obtenir les zéros en position $(k+2, k)$, $(k+3, k)$, ..., (n, k) on effectue les transmutations pour les matrices $E_{i, k+1}(a_i)$ avec

$$a_i = - \frac{A_{i,k}}{A_{k+1,k}} \quad (i=k+2, \dots, n).$$

On peut effectuer toutes ces transmutations en même temps. Si on désigne par A'_{ij} l'élément i, j de la matrice A obtenu après avoir fait toutes ces transmutations, on aura les formules de calcul suivantes :

$$\begin{aligned} A'_{j, k+1} &= A_{j, k+1} - \sum_{i=k+2}^n a_i A_{ji} & (j=1, \dots, k+1) \\ A'_{i, j} &= A_{i, j} + a A_{k+1, j} & (j \neq k+1) \\ A'_{i, k+1} &= A_{i, k+1} + a_i (A_{k+1, k+1} - \sum_{j=k+2}^{i-1} a_j A_{k+1, j}) - a_i A_{i, i} \\ &\quad - a_i^2 A_{k+1, i} & (i=k+2, \dots, n). \end{aligned}$$

Le calcul le plus long à faire sera celui de $A'_{n, k+1}$.

Le calcul de $\sum_{j=k+2}^{n-1} a_j A_{k+1, j}$ prendra $\lceil \log_2 n - k - 2 \rceil + 1$ unités de temps.

Pour obtenir $A'_{n, k+1}$ il faudra donc en tout :

$$\lceil \log_2(n-k-1) \rceil + 3 \text{ unités de temps.}$$

La forme d'Hessenberg sera donc obtenue de cette manière en un temps égal à :

$$\sum_{k=1}^{n-2} (\lceil \log_2(n-k-1) \rceil + 3)$$

On aura donc bien en coût en $O(n \log n)$. □

Dans le cas du calcul de la forme tridiagonale, ou de celle de Frobenius, on calculera d'abord la forme d'Hessenberg par la méthode précédente. On regroupe ensuite toutes les transmutations qui permettent d'obtenir les zéros dans les colonnes $3, \dots, n$ (pour la forme tridiagonale), $1, \dots, n-1$ pour la forme de Frobenius. On aura toujours $O(n \log n)$ unités de temps.

5 . MATRICES DE FORMES PARTICULIERES

a/ Matrices bandes

Soit A une matrice de $M_{n,n}(K)$. On dira que A est une matrice bande de largeur de bande $2k+1$ si on a :

$$A_{i,i+q} = 0 \quad \text{si} \quad q > k$$

$$A_{i-q,i} = 0 \quad \text{si} \quad q > k.$$

Exemple

Une matrice tridiagonale est une matrice bande de largeur de bande 3 ($k=1$). On désigne par B_k l'ensemble des matrices bandes de $M_{n,n}(K)$, de largeur de bande égale à $2k+1$.

Il est immédiat de constater que le produit en parallèle de deux matrices de B_k peut se faire en

$$\lceil \log_2 2k+1 \rceil + 1 \text{ unités de temps.}$$

Si la largeur de bande est petite (par rapport à n) alors le produit en parallèle de deux matrices bandes est plus rapide que celui de deux matrices quelconques. (Il ne dépend plus que de la largeur de la bande).

En est-il de même pour le calcul de l'inverse et pour la résolution d'un système linéaire ?

On va montrer le résultat suivant :

Théorème 6

/ Le calcul de l'inverse d'une matrice bande de largeur de bande $2k+1$ peut se faire en parallèle avec

$$O(\text{Log}_2 \left(\frac{2k+1!}{k! k+1!} \right) \cdot \text{Log}_2 n) \text{ unités de temps.}$$

Il en est de même pour le calcul du déterminant, ou pour la résolution d'un système linéaire. /

□ On va montrer le résultat pour le calcul en parallèle du déterminant d'une telle matrice. Comme on a :

$$R_{\infty}(n) \leq D_{\infty}(n) + 1 \quad \text{et} \quad I_{\infty}(n) \leq D_{\infty}(n-1) ,$$

les résultats sur le calcul de l'inverse et sur la résolution du système linéaire en découleront aussitôt.

On va calculer le déterminant d'une matrice de B_k par la règle de Laplace. Rappelons que si $\sigma_1^r(i_k) = (i_1, i_2, \dots, i_r)$, ($1 \leq i_1 < i_2 < \dots < i_r \leq n$) est un r uplets d'indices alors on a :

$$\det A = (-1)^{I_r} \sum (-1)^{I_r} \det A [\sigma_1^r(i_k) | \sigma_1^r(j_k)] \det A [\overline{\sigma_1^r(i_k)} | \overline{\sigma_1^r(j_k)}]$$

où la sommation est étendue à tous les ensembles $\sigma_1^r(j_k)$ possibles et où :

$$I_r = \sum_{k=1}^r i_k \quad , \quad I_r = \sum_{k=1}^r j_k .$$

$A [\sigma_1^r(i_k) | \sigma_1^r(j_k)]$ désigne la sous-matrice de A formée avec les lignes i_1, \dots, i_r et les colonnes j_1, \dots, j_r .

$A[\overline{\sigma_1^r(i_k)} / \overline{\sigma_1^r(j_k)}]$ désigne la sous-matrice de A formée avec les lignes et les colonnes restantes.

Supposons $n = 2p$ et appliquons la formule de Laplace dans le cas

$$\sigma_1^p(i_k) = (1, 2, \dots, p).$$

Les déterminants à calculer sont tous des déterminants de matrices bandes de largeur de bande au plus égal à $2k+1$, et de dimension $p \times p$. On voit d'autre part qu'il n'existe que $\binom{k}{2k+1}$ termes non nuls dans la somme provenant de la formule de Laplace.

Théorème 7

/ Le calcul en parallèle par la méthode de Givens de la forme tridiagonale semblable à une matrice symétrique nécessite $O(n^2)$ unités de temps. /

□ Examinons les calculs à faire. Posons $c = \cos \theta$ et $s = \sin \theta$.

$$V_{ki}(\theta) = I + (c-1)(E_{kk} + E_{ii}) + s E_{ik} - s E_{ki},$$

$$V_{ki}^{-1}(\theta) = V_{ki}^t(\theta).$$

• La matrice $V_{ki}^t(\theta) A V_{ki}(\theta)$ ne diffère de la matrice A que par les éléments de ses lignes k et i et de ses colonnes k et i .

Si on pose $A' = V_{ki}^t(\theta) A V_{ki}(\theta)$, on aura :

$$A'_{kj} = c A_{kj} + s A_{ij} \quad \text{pour } j \neq k \text{ et } j \neq i$$

$$A'_{ij} = s A_{kj} + c A_{ij} \quad \text{pour } j \neq k \text{ et } j \neq i$$

$$A'_{kk} = c^2 A_{kk} + 2 sc A_{ik} + s^2 A_{ii}$$

$$A'_{ik} = (c^2 - s^2) A_{ik} + sc(A_{ii} - A_{kk}).$$

Pour mettre un zéro en $(1,3)$ par exemple (et en $(3,1)$), on transmue la matrice A par la matrice $V_{23}(\theta)$, où θ est choisie de telle manière que l'on ait :

$$-s A_{12} + c A_{13} = 0$$

On doit donc avoir :

$$c = \frac{A_{12}}{\sqrt{A_{12}^2 + A_{13}^2}}, \quad s = \frac{A_{13}}{\sqrt{A_{13}^2 + A_{12}^2}}$$

On peut calculer, en parallèle, les quantités s , c , s^2 , c^2 et sc en quatre unités de temps (si on suppose que le calcul d'une racine carrée prend une unité de temps).

Le calcul de la matrice $V_{ki}^t(\theta) A V_{ki}(\theta)$ prendra en parallèle trois unités de temps si on connaît sc , c^2 et s^2 .

Ceci montre que pour obtenir un zéro en (i,j) et en (j,i) , il faudra sept unités de temps.

En huit unités de temps seulement on peut faire plusieurs transmutations par des matrices $V_{i_1 j_1}(\theta), V_{i_2 j_2}(\theta), \dots, V_{i_k j_k}(\theta)$ si tous les indices $i_1, j_1, \dots, i_k, j_k$ sont distincts et si $V_{i_p j_p}(\theta)$ est choisie de façon à créer un zéro qui ne se trouve pas sur les lignes et les colonnes modifiées par les autres matrices (lignes et colonnes d'indices $i_1, \dots, i_{p-1}, i_{p+1}, \dots, i_k$ et $j_1, \dots, j_{p-1}, j_{p+1}, \dots, j_k$).

Mais il n'est pas possible de changer l'ordre d'obtention des zéros de façon à profiter de cette possibilité :

On devra mettre tous les zéros en (k, i) ($k > i+1$) avant d'en mettre dans les positions (k', j) $k' > j+1$ et $j > i$, (de façon à conserver les zéros créés).

Il n'est donc pas possible de faire plusieurs transmutations en parallèle. Par conséquent, le temps de calcul sera proportionnel au nombre de transmutations nécessaires. Il sera donc en $O(n^2)$. \square

Remarque 1

Le résultat est le même dans le cas où on transmue une matrice A sous forme d'Hessenberg à l'aide de matrices de rotation .

Remarque 2

Dans la méthode de Jacobi pour l'obtention des valeurs propres d'une matrice symétrique (ou hermitique), on utilise une suite de transmutations de la matrice par des matrices de rotation . Après chaque transmutation on peut annuler un élément (p, q) et (q, p) en dehors de la diagonale. Pour que cette méthode converge il faut simplement que l'on annule une infinité de fois tous les éléments hors diagonaux. Comme on n'a pas besoin de conserver les zéros créés, on peut donc penser, pour cette méthode, à faire plusieurs transmutations en parallèle par des matrices $V_{i_1, j_1}(\theta_1), \dots, V_{i_k, j_k}(\theta_n)$ (les indices devant être tous distincts). On ne peut donc faire au plus que $\lfloor \frac{n}{2} \rfloor$ transmutations en parallèle.

Or un cycle de la méthode de Jacobi utilise $\frac{n(n-1)}{2}$ transmutations. La question qui se pose est donc la suivante :

Peut-on regrouper ces $\frac{n(n-1)}{2}$ transmutations de façon à les exécuter en $\frac{\lfloor \frac{n(n-1)}{2} \rfloor}{\lfloor \frac{n}{2} \rfloor}$ étapes en parallèle regroupant au plus $\lfloor \frac{n}{2} \rfloor$ de ces transmutations ?

Cette stratégie sera donc optimale en ce qui concerne l'utilisation du parallélisme.

SAMEH (24) a donné une réponse affirmative à cette question en décrivant deux telles stratégies optimales.

Les résultats précédents montrent donc que la plupart des calculs matriciels se prêtent bien à un calcul en parallèle.

Dans le chapitre suivant, on aborde l'étude de l'organisation des données d'un programme.

REFERENCES SUR LE CHAPITRE CIV

-
- (20) BRENT, RP., "The parallel evaluation of arithmetic expressions in logarithmic time".
in "Complexity of sequential and parallel numerical algorithms"
ed. by TRAUB, (1973).
- (21) CSANSKY, L., "Fast parallel matrix inversion algorithms".
Preprint (may 1974).
- (22) LAFON, JC., "Calculs algébriques en parallèle".
Colloque Analyse Numérique, Super-Besse (mai 1970).
- (23) MIRANKER, WL., "A survey of parallelism in Numerical Analysis"
SIAM Review vol. 13, N° 4, (oct. 1971).
- (24) SAMEH, A., "On Jacobi and Jacobi like algorithms for a parallel computer."
Mathematics of Computation vol.25, number 115, (july 1971).
- (25) STONE, HS., "An efficient parallel algorithm for the solution of a tridiagonal linear systems of equations".
J. ACM, 120 (1973) 27-38.
- (26) TRAUB, JF., "Iterative solution of tridiagonal systems on parallel or vector computers".
in Complexity of sequential and parallel numerical algorithms
ed. by TRAUB (1973).
- (27) WINOGRAD, S. " On the Algebraic Complexity of inner product".
Linear Algebra and its Applications 4 (1971), 377-379.

CHAPITRE CV

INFLUENCE DE LA PAGINATION SUR LA RAPIDITE
D'EXECUTION DES ALGORITHMES DE CALCULS MATRICIELS

PLAN

- I . Principales définitions.
- II. Modèles mathématiques.
- III. Etude de différents algorithmes.
 - 1/ Stockage d'une matrice.
 - 2/ Opérations élémentaires sur les matrices.
 - 3/ Résolution d'un système linéaire.
 - 4/ Obtention des formes canoniques.
 - 5/ Tridiagonalisation d'une matrice symétrique.
 - a/ Méthode de Givens.
 - b/ Méthode d'Householder.
 - 6/ Calcul des valeurs propres et des vecteurs propres par la méthode de Jacobi.

INTRODUCTION

Le but de ce chapitre est de mettre en évidence l'influence de l'organisation en pages de la mémoire centrale d'un ordinateur, sur la rapidité d'exécution d'un algorithme.

Dans une première partie, on donne les définitions des principaux concepts liés à celui de pagination.

On développe ensuite un modèle général de calcul et on définit la notion de "l'implémentation optimale d'un algorithme". On pose le problème de la détermination d'un algorithme efficace permettant de trouver cette implémentation.

Dans la dernière partie, on décrit les implémentations optimales de différents algorithmes de calcul matriciel :

- produit de deux matrices - résolution d'un système linéaire - inversion d'une matrice,
- transmutation d'une matrice sous l'une des trois formes condensées suivantes : Hessenberg - tridiagonale - Frobénius.
- calcul des vecteurs propres et des valeurs propres par la méthode de Jacobi-Tridiagonalisation d'une matrice symétrique : méthode de Givens, d'Householder.

On donne une estimation théorique du gain en appels de pages obtenu sur les algorithmes classiques.

I . PRINCIPALES DEFINITIONS

Le présent paragraphe consiste essentiellement en un exposé des notions classiques de mémoire virtuelle - page - stratégie d'appels - de recherche de pages...bien connues des informaticiens. Il est donc uniquement destiné à clarifier les notations et à montrer l'intérêt des questions que l'on traite par la suite.

1. Mémoires hiérarchisées

Dans un ordinateur, toutes les mémoires disponibles n'ont pas la même importance. On distingue essentiellement la mémoire centrale et les mémoires secondaires, celles-ci se subdivisant en mémoires rapides, lentes, à accès direct ou séquentiel. Seule, la partie d'un programme résidant en mémoire centrale peut être exécutée.

Lorsque la taille d'un programme (code instructions + données) dépasse la taille allouée en mémoire centrale, une partie du programme doit se trouver sur une mémoire auxiliaire. Il est alors nécessaire, au cours de l'exécution du programme, de déplacer des parties du programme de la mémoire secondaire dans la mémoire principale quand celles-ci doivent être exécutées (quand elles ne sont pas déjà en mémoire centrale) et de la même façon certaines parties seront renvoyées sur la mémoire secondaire.

Pendant longtemps, ce fut au programmeur d'organiser ces mouvements. Avec l'apparition des langages de haut niveau et des ordinateurs de la troisième génération, est apparue la nécessité de le décharger de cette tâche : ceci a été réalisé grâce au concept de mémoire virtuelle (33).

2. Mémoire virtuelle

Le programmeur définit son espace mémoire en utilisant des symboles d'un certain langage. Il ne se préoccupe plus de définir la correspondance entre cet ensemble d'adresses symboliques et les adresses réelles. Cette dissociation permet donc d'une part de faire comme si la mémoire disponible était infinie, et d'autre part permet l'écriture de programmes indépendants de la machine. C'est le système hardware et le software de l'ordinateur qui

s'occupent de déplacer les différentes parties du programme entre la mémoire centrale et la mémoire auxiliaire, ceci grâce aux mécanismes de pagination ou de segmentation. La mémoire virtuelle est cette mémoire de capacité infinie dont semble se servir le programmeur.

3. Pagination

Soit A l'ensemble des adresses symboliques utilisées par le programmeur.

La mémoire centrale de l'ordinateur est divisée en blocs de même taille : chaque bloc comprenant par exemple p zones de mémoires élémentaires (p mots par exemple). Un bloc est appelé une page. De la même façon, la mémoire virtuelle est divisée en pages de même taille.

L'ensemble A sera donc divisé en a pages. La mémoire centrale sera supposée divisée en b pages.

Chaque page sera identifiée par l'adresse de sa première zone. Une adresse virtuelle sera la donnée d'un couple de deux nombres (n,m) : $1 \leq n \leq a$, $m \leq p$, et désignera la même zone de la même page virtuelle.

Une table des adresses des pages se trouvant dans la mémoire centrale est constamment mise à jour. En face de la même entrée de cette table se trouve l'adresse du début de la page de la mémoire centrale si cette page n se trouve là dans la mémoire centrale, un blanc autrement. Dans ce cas, il sera nécessaire d'aller chercher la page n dans la mémoire auxiliaire si on en a besoin. Les différents mécanismes alors nécessaires sont expliqués dans la suite.

4. Gestion de la mémoire

Quand au cours de l'exécution d'un programme une référence est faite à une page qui ne se trouve pas en mémoire centrale, on dit qu'il se produit un "appel de page".

L'exécution du programme s'intrompt pendant que cette page est cherchée dans la mémoire principale et chargée en mémoire.

Pour réaliser cela, il peut être nécessaire :

- a. de libérer de la place en mémoire centrale : il faudra choisir quelles pages enlever : nécessité d'une stratégie de libération de place ;
- b. de déterminer quelles pages amener en mémoire centrale : stratégie de recherche de pages.

La stratégie de recherche la plus simple est celle qui consiste à attendre qu'une page soit demandée pour la chercher.

On peut imaginer des stratégies plus raffinées, qui amèneraient en mémoire centrale certaines pages avant qu'elles ne soient demandées. Ceci ne présente pas d'intérêt dans le cas de mémoires auxiliaires à accès direct. Dans la suite, on ne considère pas de telles stratégies.

5. Fonctionnement en multi-programmation et en temps partagé

Dans un tel système, l'utilisation d'une catégorie de ressources n'est disponible que pendant un bref intervalle de temps par un programme. Ces intervalles de temps peuvent résulter naturellement des passages entre phases d'exécution du programme et phases d'attente, ou être définis artificiellement par l'attribution d'un "quanta" de temps à chaque programme (temps partagé).

Le système de pagination amène en mémoire la page du programme où se trouve la dernière exécution référencée avant l'arrêt, puis effectue les appels ultérieurs. A la fin du "quanta", les pages du programme en mémoire y restent, c'est le rôle de la stratégie de libération de libérer de la place pour le programme suivant.

II. MODELES MATHEMATIQUES

Dans ce paragraphe, on formalise les concepts d'algorithmes, d'implémentation d'un algorithme, de stratégie de libération.

On définit l'optimalité d'une stratégie ou d'une implémentation d'un algorithme.

1. Programmes

Un programme sera défini par la donnée des ensembles suivants :

- a) un ensemble M de mémoires
- b) un ensemble D de données
- c) un ensemble O d'opérations
- d) un ensemble I d'instructions .

On suppose I fini et totalement ordonné .

Soit donc I_1, \dots, I_N ces instructions.

Elles seront d'un des types suivants :

- Instruction d'affectation

$$\begin{aligned} m \in M \quad m \leftarrow d \text{ affecte à } m \text{ la valeur } d \\ d \in D \end{aligned}$$

- Instruction de calcul (transformation des données)

$m_0 \leftarrow g(m_1, \dots, m_i)$ si g admet i arguments ; l'opération g porte sur le contenu des mémoires m_1, \dots, m_i , le résultat de l'opération est dans la mémoire m_0 .

- Instruction de transfert

$$m \rightarrow i, j$$

Si le contenu de m est égal à 1, alors l'instruction à exécuter sera la ième, sinon la jème.

Les instructions du programme se déroulent séquentiellement, seules les instructions de transferts peuvent rompre cette suite.

2. Implémentation d'un programme

Il s'agit de définir une bijection entre l'ensemble MUDUI et l'ensemble L des locations physiques disponibles.

On suppose dans la suite que l'on a des blocs de p mémoires élémentaires. Un tel bloc constitue une page. Une implémentation possible correspond à un partitionnement de l'ensemble MUDUI = A en un nombre a de pages de taille p .

Lors de l'exécution du programme, l'ordre d'exécution des instructions détermine un ordre d'appel des pages. Cet ordre sera toujours le même si le programme ne contient pas d'instructions de branchement conditionnel. Dans le cas contraire, on peut définir la probabilité c_{ij} de passer de la page i à la page j .

Une première définition possible d'une implémentation optimale est donc la suivante :

Trouver le partitionnement initial en pages, tel que l'on minimise la somme :

$$S = \sum_{1 \leq i, j \leq N} c_{ij}$$

On va définir dans la suite une autre notion d'optimalité.

3. Stratégie de remplacement - cf. BELADY (28), GELENBE (35).

La définition précédente d'une implémentation optimale ne fait pas intervenir la taille de la mémoire réelle.

Soit b le nombre de pages pouvant résider dans la mémoire centrale.

Dans ces conditions, pour une séquence d'appels de pages, il faut définir quelles seront les pages à mettre en mémoire et celles qu'il faut enlever.

Dans le cas où on tente de placer en mémoire centrale une page seulement quand un appel à cette page est fait, alors il suffit de définir la stratégie de remplacement, c'est-à-dire l'algorithme permettant de déterminer quelle page il faut enlever de la mémoire centrale.

Soit une séquence d'appels de pages : p_1, \dots, p_k , $p_i \in A$.

p_k est référencée au temps t_k .

p_{k+1} sera référencée au temps $t_{k+1} = t_k + \Delta$ si p_k est déjà dans la mémoire centrale (Δ = temps moyen de référence d'une page en mémoire centrale), au temps $t_{k+1} = t_k + \Delta + T$ si p_k n'est pas en mémoire centrale.

(T = temps moyen de recherche d'une page en mémoire secondaire).

En général, on a $\Delta = 1 \mu s$, $T = 1 ms$.

On a donc intérêt à minimiser le nombre d'appels de pages non en mémoire centrale. Si $b = 1$, cela revient à minimiser le nombre total d'appels de pages dans le cas déterministe ; la somme $S = \sum_{i,j} c_{ij}$ dans le cas non déterministe.

Définition d'une stratégie de remplacement

On adopte la définition suivante d'une telle stratégie.

Soit Q un ensemble fini : ensemble des états possibles. q_0 est un état particulier de Q : l'état initial. Une stratégie de remplacement sera une application de l'ensemble $Q \times L \times A$ dans l'ensemble $Q \times L$.

A est l'ensemble des a pages du programme.

L est l'ensemble de toutes les parties de A ayant b éléments au plus.

Soit f une telle application : $Q \times L \times A \xrightarrow{f} Q \times L$.

Alors, si la t ème page référencée est la page p_t , si l'on est dans l'état q , avec une mémoire centrale de configuration l , $f(q, l, p_t)$ sera égal au couple (q', l') , q' sera l'état suivant, et l' la nouvelle configuration de la mémoire centrale.

L'application f respecte les conditions suivantes :

si $p_t \in \ell$ alors $p' = p$ (seul l'état change)

si $p_t \notin \ell$ deux cas possibles :

1er cas $|\ell| < b$ alors $\ell' = \{\ell \cup p_t\}$

2ème cas $|\ell| = b$ $p_t \in \ell' \quad \exists p_t, \in p \mid p_t, \notin p'$.

p_t , est une page enlevée de la mémoire centrale. On suppose qu'il n'y en a toujours qu'une.

La donnée d'une telle application permet pour toute séquence d'appels de pages, de déterminer les états successifs de la mémoire centrale.

4. Stratégie de remplacement optimale et implémentation optimale d'un programme

a/ Stratégie de remplacement optimale.

Une telle stratégie doit, pour une séquence d'appels de pages donnée, minimiser le nombre de fois où une page doit être enlevée de la mémoire centrale (nombre de défauts de pages).

Quand la suite des appels de pages est connue, alors la stratégie de remplacement optimale est celle qui consiste à remplacer la page dont la prochaine référence est la plus lointaine. (cf. BELADY (28)).

Une telle stratégie n'est en pratique guère possible. Elle suppose que l'on ait déjà exécuté le programme et que celui-ci ne comporte pas de branchements conditionnels.

b/ Implémentation optimale d'un algorithme.

Avec la définition précédente d'une stratégie de remplacement optimale, on peut maintenant définir ce qu'est dans l'absolu une implémentation optimale d'un algorithme : c'est l'implémentation telle que l'application de la stratégie de remplacement optimale pour la séquence d'appels de pages générée conduise au nombre minimal de défauts de pages.

C'est un problème extraordinairement complexe à résoudre pour une distribution quelconque d'instructions et de données. Dans la suite, on détermine cependant les implémentations optimales de certains algorithmes pour lesquels on peut appréhender facilement l'enchaînement des instructions, quand le nombre de pages pouvant coexister en mémoire centrale est le plus petit nombre possible pour que l'algorithme puisse être exécuté.

3 . ETUDE DE DIFFERENTS ALGORITHMES

Dans cette partie, on va étudier à la fois sur le plan théorique et sur le plan pratique, les performances des principaux algorithmes de calcul matriciel dans un environnement paginé, ainsi que les variantes que l'on peut imaginer.

Rappelons tout d'abord quelques notations :

p sera la taille d'une page

b sera le nombre de pages pouvant coexister en mémoire centrale

$M_{n,n}$ désigne l'ensemble des matrices carrées $n \times n$ à éléments réels.

Dans toute la suite, on suppose pour simplifier :

$$p = qn \quad n = qa^2 \quad (=> p = a^2q^2).$$

1. Stockage d'une matrice

Si la matrice A est stockée ligne par ligne, une page contiendra exactement q lignes.

Si la matrice A est maintenant considérée comme constituée de a^2 sous matrices carrées $aq \times aq$, on voit qu'on peut stocker un tel bloc par page : stockage par blocs de la matrice A .

Dans le cas où l'on n'a pas les relations $p = qn$ et $n = qa^2$, on considère que l'on a $\lfloor \frac{p}{n} \rfloor$ lignes par pages et que l'on a des blocs de $\lfloor \sqrt{p} \rfloor \times \lfloor \sqrt{p} \rfloor$: cela entraîne une perte de place en mémoire, mais facilite l'adressage à l'intérieur d'une page.

Obtention d'une ligne ou d'une colonne complète de A.

Stockage en lignes : 1 appel pour 1 ligne , a^2 appels pour 1 colonne complète.

Stockage en blocs : a appels pour 1 ligne, ou 1 colonne complète.

Par conséquent, à moins que le nombre d'appels de lignes soit moindre que celui des colonnes, le stockage en blocs d'une matrice est celui qui donne en moyenne le moins d'appels de pages. (Résultat de Coffman (32)).

2. Opérations élémentaires sur les matrices

a/ Transposition d'une matrice.

On veut obtenir la matrice $B = A^t$ ($B_{ij} = A_{ji}$).

Considérons l'algorithme suivant de calcul de B.

```

      Pour I:=1 jusqu'à N faire
I      Pour J:=1 jusqu'à N faire
          B(I,J):=A(J,I) ;

```

Stockage en lignes de A et de B : $O(qa^4)$ appels de pages.

Stockage en blocs de A et de B : $O(qa^3)$ appels de pages.

Pour cet algorithme, le stockage en blocs est le meilleur. Peut-on écrire différemment (I) de façon à diminuer le nombre d'appels de pages ?

1) Pour le stockage en blocs, il est immédiat de constater que l'on doit faire les opérations entre blocs : $O(2a^2)$ appels de pages seulement dans ce cas.

2) Dans le cas du stockage par lignes.

Il faut écrire le programme de telle façon que l'on fasse le maximum d'échanges entre pages : lorsqu'une page de B est en mémoire, les q premiers éléments des p lignes correspondantes de B se trouvent dans la première page de A, les q suivants dans la seconde On effectue la transmutation de ces sous-matrices $q \times q$ et l'on aura : $O(a^4)$ appels de pages.

Le stockage par blocs est bien le meilleur.

b/ Produit de deux matrices.

Soit $C = A \times B$.

Considérons l'algorithme ordinaire pour effectuer le produit de deux matrices :

```

pour I:=1 jusqu'à N faire
pour J:=1 jusqu'à N faire
  début
    S:=0 ;
    pour k:=1 jusqu'à N faire
      S:=S+A(I,K) B(K,J) ;
    C(I,J):=S ;
  fin ;

```

Comptons le nombre d'appels de pages.

c1) Stockage en lignes

$q^2a^6 + 2a^2$ appels de pages

c2) Stockage en blocs

$2q^2a^5 + a^2$ appels de pages.

On peut chercher maintenant quelle est la meilleure façon d'écrire l'algorithme pour chacune de ces organisations des données.

b'_1) Stockage en lignes.

Chacune des trois matrices A,B,C est considérée comme partitionnée en $a^2 \times a^2$ sous-matrices $q \times q$. L'algorithme optimal consiste simplement à calculer la matrice produit C par les formules obtenues en effectuant les produits par blocs.

Il est facile de calculer le nombre d'appels de pages nécessaires : $a^6 + a^2$.

b'_2) Stockage par blocs.

Les trois matrices sont considérées comme partitionnées en $a \times a$ matrices $qa \times qa$.

L'algorithme optimal est l'analogie du précédent pour ce partitionnement.

Le nombre d'appels de pages devient : $2a^3 + a^2$.

3. Résolution d'un système linéaire.

De la même façon que pour les algorithmes précédents, on peut voir que le stockage en blocs est celui qui donne le nombre minimum d'appels de pages. Cette étude a été faite par COFFMAN-McKELLAR (32).

4. Transmutation d'une matrice sous forme canonique

(Hessenberg, Tridiagonale, Frobenius).

Désignons par E_{ij} la matrice de $M_{n,n}$ dont tous les éléments sont nuls sauf l'élément (i,j) égal à un. Soit $E_{ij}(a)$ la matrice :
 $I + a E_{ij}$ ($a \in R$).

Pour obtenir l'une des trois formes canoniques d'une matrice A , on peut effectuer des transmutations par des matrices du type $E_{ij}(a)$, chaque transmutation élémentaire permettant d'obtenir un nouveau zéro dans une position convenable. On va étudier le coût en appels de pages de ces algorithmes. On décrira les modifications à apporter à ces algorithmes pour en minimiser ce coût.

a / Transmutation par une matrice élémentaire

Soit $E_{ij}(a)$ cette matrice. Soit A' la nouvelle matrice :

$$A' = E_{ij}(a) A E_{ij}(-a).$$

On a :

$$a'_{ik} = a_{ik} + a a_{jk} \quad k \neq j$$

$$a'_{kj} = a_{kj} - a a_{ki} \quad k \neq i$$

$$a'_{ij} = a_{ij} + a a_{jj} - a a_{ii} - a^2 a_{ji}$$

Le coût en appels de pages d'une seule transmutation est de $1 + a^2$ ou $2 + a^2$ appels pour le stockage en lignes de A suivant que les lignes i et j sont, ou ne sont pas, dans les mêmes pages.

Pour le stockage par blocs, on aurait un coût de $2a$ ou de $4a$ appels.

Dès que $a \geq 4$, le stockage en blocs est meilleur à ce point de vue que le stockage en lignes.

b/ Transmutation sous forme d'Hessenberg

- Algorithme classique (7^A).

Il consiste en une suite de transmutations élémentaires :

$E_{3,2}, E_{4,2}, \dots, E_{4,3}, E_{5,3}, \dots, E_{n,n-1}$ permettant d'obtenir un zéro successivement en les positions $(3,1), (4,1), \dots, (n,1), (4,2), (5,2), \dots, (n,2), \dots, (n,i), \dots, (n,n-2)$. Pour effectuer la transmutation $E_{j,i+1}(a)$, on calcule la valeur de a :

$a = -\frac{A(j,i)}{A(i+1,i)}$, puis on ajoute a fois la $i+1$ ème ligne à la j ème et on

retranche a fois la j ème colonne à la $i+1$ ème.

Pour obtenir tous les zéros voulus dans la i ème colonne, il faut faire $(n-i-1)$ telles transmutations.

- Stockage en ligne.

Si l'on suppose que trois pages peuvent coexister en mémoire centrale, il faudra alors un nombre d'appels de :

$$T = \sum_{i=1}^{n-2} \left((n-i)a^2 + \left\lfloor \frac{n-i}{q} \right\rfloor \right)$$

$$\text{Soit : } \frac{q^2 a^6}{2} + \frac{qa^2}{2} - 2a - 1 \quad (T = 0 \left(\frac{q^2 a^6}{2} \right)) .$$

Si deux pages seulement peuvent coexister dans la mémoire, le coût en appels deviendra en $q^2 a^6$ car à chaque fois il faudra réobtenir la $i+1$ ème ligne et la $i+1$ ème colonne :

$$\frac{(n-2)(n-1)}{2} (1+a^2) \text{ appels supplémentaires.}$$

- Stockage en blocs

Si trois pages peuvent coexister en mémoire centrale, le coût sera de :

$$T = \sum_{i=1}^{n-2} \left[(n-i)a + \left\lceil \frac{n-i+1}{qa} \right\rceil (n-i) \right] .$$

$$\text{Soit : } a \frac{n(n-1)}{2} + \sum_{i=0}^{a-1} (i+1) \sum_{j=1}^{qa} (iqa-1+j) - 2$$

ce qui donne $T = O(\frac{5}{6} q^2 a^5)$.

Dans le cas de deux pages seulement, on aurait :

$$T = O(\frac{5}{3} q^2 a^5).$$

-Modifications de l'algorithme

. Stockage en lignes

Dans ce cas, il est clair que les opérations entre colonnes sont très coûteuses. On cherchera donc à ne faire que des opérations sur les lignes.

Pour cela, on regroupe en une seule étape le calcul de :

$$E_{n,i+1}, \dots, E_{i+2,i+1} A E_{i+2,i+1}^{-1}, \dots, E_{n,i+1}^{-1}$$

Il sera en fait nécessaire d'avoir $n-2$ mémoires supplémentaires. On a tout d'abord les opérations entre les lignes $i+1, i+2, i+3, \dots, n$ en stockant les valeurs

$$\frac{A(i+2,i)}{A(i+1,i)}, \dots, \frac{A(n,i)}{A(i+1,i)}.$$

On calcule ensuite les nouvelles valeurs des éléments de la colonne $i+1$ par les formules :

$$A(j,i+1) = A(j,i+1) + \sum \frac{A(k,i)}{A(i+1,i)} \times A(j,k)$$

Si l'on peut avoir trois pages en mémoire centrale, on a un coût en appels de pages de :

$$T = \sum_{i=1}^{n-2} (a^2 + \lceil \frac{n-i}{q} \rceil)$$

$$\text{Soit } T = \frac{3qa^4}{2} - 3a + \frac{qa^2}{2} - 1 \quad T = O\left(\frac{3qa^4}{2}\right).$$

Remarquons que l'on peut encore économiser quelques appels en commençant les calculs sur la $i+1$ ème colonne à partir de la même ligne (en mémoire après les calculs sur la ligne) : cela économise $qa^2 - 2$ appels supplémentaires.

. Stockage par blocs

On calculera les valeurs $\frac{A(i+1,2)}{A(i+1,i)}, \dots, \frac{A(n,i)}{A(i+1,i)}$ en effectuant toutes les opérations entre les lignes se trouvant dans les $\lceil \frac{n-i}{qa} \rceil$ pages nécessaires.

Si l'on dispose de qa mémoires supplémentaires, on pourra calculer qa

sommes partielles $\sum_{k=n-qa+1}^n \frac{A(k,i)}{A(i+1,i)} A(j,k)$ simultanément pour $j = n, n-1, \dots,$

$\dots, n-qa+1$, et ainsi de suite, ceci dès que dans une page on aura fait toutes les opérations entre la $i+1$ ligne et celle de la page.

On aura donc ainsi un coût de :

$$\sum_{i=1}^{n-2} (a+1) \left\lceil \frac{n-i}{qa} \right\rceil + \left\lceil \frac{n-i}{qa} \right\rceil + \left\lceil \frac{n-i}{qa} \right\rceil$$

$$\text{Soit } T = (a+1) \sum_{i=2}^{n-1} \left\lceil \frac{i}{qa} \right\rceil + qa \left(\sum_{i=1}^{a-1} i^2 \right) - 1 - a^2$$

$$\Rightarrow T = O\left(\frac{5}{6} qa^4\right).$$

Remarquons, cependant, que si l'on avait exactement n mémoires supplémentaires, on pourrait faire ceci pour toutes les lignes en même temps.

$$\text{Le coût deviendrait : } \sum_{i=1}^{n-2} (a+1) \left\lceil \frac{n-i}{qa} \right\rceil = \frac{qa^4}{2} + a^2 \left(\frac{q}{2} - 1\right) - 2a - 1 + qa^3$$

$$\text{Soit } T = O\left(\frac{qa^4}{2}\right).$$

Le stockage en blocs reste donc meilleur mais nécessite cependant plus de mémoires. Comme pour le stockage en lignes, on a seulement trois fois plus au maximum d'appels, l'algorithme correspondant est donc préférable.

c / Tridiagonalisation

On réduit tout d'abord la matrice A sous la forme d'Hessenberg. Ensuite, on effectue les transmutations successives $E_{i+1,j}$, $j=i+2, \dots, n$, $i=1, \dots, n-2$ de façon à obtenir les zéros en les positions successives $(1,3), (1,4), \dots, (1,n), (2,3), \dots, (2,n), \dots, (n-2,n)$.

On modifie très facilement cet algorithme, de manière à faire toutes les opérations sur une même ligne en même temps.

On regroupe en une même étape les transmutations permettant d'obtenir les zéros voulus dans la ième ligne.

Cela nécessite $n-2$ mémoires supplémentaires de façon à pouvoir stocker les

$$\text{valeurs } B(j) = -\frac{A(i,j)}{A(i,i+1)} \quad j = i+2, \dots, n .$$

On fera ensuite $A(i,j) = 0$; on n'a plus besoin de la ième ligne.

On fera ensuite pour $k = i+2$ jusqu'à N les calculs suivants :

$$\begin{aligned} A(i+1,L) &= A(i+1,L) - B(k)A(k,L) \quad L = k-1 \text{ jusqu'à } N . \\ A(k,L) &= A(k,L) + B(L) \times A(k,i+1) \text{ pour } L = k \text{ jusqu'à } N . \\ A(i+1,k) &= A(i+1,k) + B(k) \times A(i+1,i+1). \end{aligned}$$

Le coût de cet algorithme sera donc :

$$\sum_{i=1}^{n-2} \left\lceil \frac{n-i+1}{q} \right\rceil = \frac{qa^2(a^2+1)}{2} - 2 .$$

Pour le stockage en blocs, il faut modifier les calculs de façon à faire tous les calculs à l'intérieur d'une page (analogue au cas précédent pour l'obtention de la matrice d'Hessenberg). On aura alors seulement :

$$\sum_{i=1}^{n-2} \left\lceil \frac{n-i}{qa} \right\rceil \left\lceil \frac{n-i+1}{qa} \right\rceil \text{ appels.}$$

$$\text{Soit } qa^2 \frac{(a+1)(2a+1)}{6} - \frac{a(a+1)}{2} - 1 .$$

$$T = O\left(\frac{qa^4}{3}\right) .$$

d/ Transmutation sous la forme de Frobenius

On suppose A non dérogatoire. On transmue d'abord A sous forme d'Hessenberg, puis on effectue les transmutations élémentaires permettant d'obtenir des zéros de la ième colonne :

$$E_{j,i+1} \quad j=1,\dots,i$$

Les zéros seront donc obtenus dans l'ordre : (1,1)(1,2)(2,2),(1,3),(2,3), (3,3),..., (1,n-1),(2,n-1),..., (n-1,n-1) .

Pour le stockage en lignes de A, l'algorithme classique n'a pas besoin d'être modifié. Il coûte en appels de pages :

$$\sum_{i=1}^{n-2} \lceil \frac{i+1}{q} \rceil$$

Soit $\frac{qa^4}{2} + \frac{qa^2}{2} - 1 - a^2$.

Pour le stockage en blocs, il faut par contre modifier l'enchaînement des opérations de façon à faire toutes les opérations concernant une page en même temps : cela nécessite de stocker les coefficients permettant d'annuler les éléments d'une colonne : n-1 mémoires supplémentaires. Le coût sera alors :

$$\sum_{i=1}^{n-2} \lceil \frac{i+1}{qa} \rceil \times \lceil \frac{n-i+1}{qa} \rceil$$

$$T = qa \sum_{j=1}^a j \times (a-j+1) + \sum_{j=1}^a j - 2a - 1 .$$

$$T = \frac{qa}{6} (a+1)(a+2) + \frac{a(a+1)}{2} - 2a - 1 .$$

Le stockage en blocs est donc meilleur que le stockage en ligne.

5. Tridiagonalisation d'une matrice symétrique.

a/ Méthode de Givens.

La méthode de Givens permet d'obtenir une matrice tridiagonale symétrique en effectuant une suite de transmutations par des matrices de rotation de la matrice initiale.

Soit $V_{ij}(\theta)$ une matrice de rotation

$$V_{ij}(\theta) = \begin{matrix} & \begin{matrix} \leftarrow i & \leftarrow j \end{matrix} \\ \begin{matrix} i \rightarrow \\ j \rightarrow \end{matrix} & \begin{pmatrix} 1 & & & \\ & \cos \theta & -\sin \theta & \\ & & 1 & \\ & \sin \theta & & \cos \theta \\ & & & & 1 \\ & & & & & 1 \end{pmatrix} \end{matrix}$$

$$A' = V_{ij}(\theta) A V_{ij}(-\theta)$$

On a les formules :

$$\begin{aligned} a'_{ik} &= \cos \theta a_{ik} - \sin \theta a_{jk} & a'_{ki} &= \cos \theta a_{ki} - \sin \theta a_{kj} & k \neq i, j \\ a'_{jj} &= \sin \theta a_{ik} + \cos \theta a_{jk} & a'_{kj} &= \sin \theta a_{ki} + \cos \theta a_{kj} \\ a'_{ii} &= \cos^2 \theta a_{ii} + \sin^2 \theta a_{jj} - 2 \sin \theta \cos \theta a_{ij} \\ a'_{jj} &= \cos^2 \theta a_{jj} + \sin^2 \theta a_{ii} + 2 \sin \theta \cos \theta a_{ij} \\ a'_{ij} &= (\cos^2 \theta - \sin^2 \theta) a_{ij} + \sin \theta \cos \theta (a_{ii} - a_{jj}) \end{aligned}$$

Ces formules montrent que l'on peut annuler le terme $(i-1, j)$ en prenant :

$$\tan \theta = - \frac{a_{i-1, j}}{a_{i-1, i}} \quad \text{Alors} \quad A'_{i-1, j} = A_{j, i-1} = 0$$

$$\text{On calcule en fait : } \cos \theta = \frac{a_{i-1, i}}{\sqrt{a_{i-1, i}^2 + a_{i-1, j}^2}} \quad \text{et} \quad \sin \theta = \frac{a_{i-1, j}}{\sqrt{a_{i-1, i}^2 + a_{i-1, j}^2}}$$

Comme la matrice est symétrique, il ne faut utiliser que les termes de la moitié supérieure par exemple. Pour minimiser le nombre d'appels de pages, il faut tâcher d'organiser les calculs entre lignes dans le cas du stockage par lignes (le stockage en blocs ne peut guère servir dans le cas des matrices symétriques si on veut stocker seulement une moitié de la matrice).

Modification de l'algorithme de Givens

On regroupe dans une même étape tous les calculs correspondant aux transmutations $V_{k+1,k+2}, \dots, V_{k+1,n}$ qui permettent d'obtenir les zéros de la kème ligne.

($k=1, \dots, n-2$).

Cet algorithme nécessite $2(n-2)$ mémoires supplémentaires pour le stockage des $\cos \theta_{k+1,j}$ et $\sin \theta_{k+1,j}$ ($j=k+2, \dots, n$).

Soit $c(j) = \cos \theta_{k+1,j}$ et $s(j) = \sin \theta_{k+1,j}$.

En ayant en mémoire la kème ligne, on peut calculer les $c(j)$ et les $s(j)$ en modifiant à chaque obtention la valeur de $a_{k,k+1}$ par la formule :

$$a_{k,k+1} = c(j) a_{k,k+1} - s(j) a_{k,j}$$

Ensuite, on effectue les transformations des éléments de A ligne par ligne, en gardant constamment la k+1ème ligne en mémoire (du moins les éléments de cette ligne se trouvant dans la moitié supérieure de la matrice). Il faut remarquer que l'élément i,j est modifié si l'on fait la transmutation par $V_{k+1,i}$ mais aussi par $V_{k+1,j}$.

On utilisera donc les formules suivantes :

pour $j=k+2$ jusqu'à N faire :

$$\begin{aligned} a'_{k+1,k+1} &= c^2(j) a_{k+1,k+1} + s^2(j) a_{j,j} - 2 c(j) s(j) a_{k+1,j} \quad , \\ a'_{k+1,j} &= (c^2(j) - s^2(j)) a_{k+1,j} + c(j) s(j) (a_{k+1,k+1} - a_{j,j}) \quad , \\ a'_{j,j} &= a_{j,j} + a_{k+1,k+1} - a'_{k+1,k+1} \quad , \end{aligned}$$

pour $L=j+1$ jusqu'à N :

$$\begin{aligned} a'_{k+1,L} &= c(j) a_{k+1,L} - s(j) a_{j,L} \quad , \\ a'_{j,L} &= s(j) a_{k+1,L} + c(j) a_{j,L} \quad , \end{aligned}$$

et pour $L=j+1$ jusqu'à N :

$$\begin{aligned} a'_{k+1,j} &= c(L) a_{k+1,j} - s(L) a_{j,L} \quad , \\ a'_{j,L} &= s(L) a_{k+1,j} + c(L) a_{j,L} \quad . \end{aligned}$$

Le coût en appels de pages sera, si l'on considère que toute la matrice est stockée :

$$\sum_{i=1}^{n-2} \left\lceil \frac{n-i+1}{q} \right\rceil = \frac{qa^2(a^2+1)}{2} - 2 .$$

Cet algorithme reste encore valable si la moitié de la matrice seulement est stockée (les formules que nous donnons pourraient être réécrites même si la matrice A symétrique était stockée sous forme d'un tableau à une seule dimension).

b/ Tridiagonalisation par la méthode de Householder

Regrouper en une seule étape l'obtention des zéros de la ième ligne (et colonne) comme précédemment, fait penser à la méthode de Householder. cf. WILKINSON (7').

Dans cette méthode, on utilise pour faire la transmutation à la ième étape la matrice $p_i = I - V_i V_i^t / 2k_i^2$

avec

$$\begin{aligned} V_i(j) &= 0 & j=1, \dots, i \\ V_i(i+1) &= a_{i,i+1} \pm S_i \\ V_i(j) &= a_{i,j} & j=i+2, \dots, n \\ S_i &= \sum_{j=i+1}^n a_{i,j}^2 & \text{et} \quad 2k_i^2 = S_i^2 + a_{i,i+1}^2 S_i \end{aligned}$$

Cette méthode ne coûte que $\frac{2}{3} n^3$ multiplications au lieu des $\frac{4}{3} n^3$ de la méthode de Givens.

On va montrer qu'au point de vue appels de pages, on peut la modifier de façon à avoir un nombre d'appels voisin de celui de la méthode de Givens.

Il suffit de considérer les formules que donne WILKINSON (7') dans son livre (p. 292) :

$$A_i = p_i A_{i-1} p_i$$

$$A_i = (I - v_i v_i^t / 2k_i^2) A_{i-1} (I - v_i v_i^t / 2k_i^2) .$$

On note $p_i = A_{i-1} v_i / 2k_i^2$

On aura donc : $A_i = A_{i-1} - v_i p_i^t - p_i v_i^t + v_i (v_i^t p_i) v_i^t / 2k_i^2$

Si $q_i = p_i - \frac{1}{2} v_i (v_i^t p_i / 2k_i^2)$, on aura :

$$A_i = A_{i-1} - v_i q_i^t - q_i v_i^t .$$

Cette dernière expression montre que le calcul de la nouvelle matrice A_i à partir de A_{i-1} de q_i et de v_i ne nécessite que $\lceil \frac{n-i}{q} \rceil + 2$ appels de pages au maximum (si v_i et q_i sont sur des pages différentes). Mais le calcul de p_i nécessite lui aussi $\lceil \frac{n-i}{q} \rceil$ appels.

On aurait aussi deux fois plus d'appels que pour la méthode de Givens. Cependant, il est clair que l'on peut commencer à déterminer v_{i+1} et $p_i v_{i+1}$ au fur et à mesure du calcul de A_i .

Par conséquent, le nombre total d'appels sera au maximum :

$$\sum_{i=1}^{n-2} (\lceil \frac{n-i}{q} \rceil + 2) + \lceil \frac{n}{q} \rceil .$$

Ce nombre est voisin de celui correspondant à la méthode de Givens.

6/ Calcul des valeurs propres et des vecteurs propres par la méthode de Jacobi.

La méthode de Jacobi permet le calcul des valeurs propres et des vecteurs propres d'une matrice carrée symétrique ou hermitienne.

Raisonnons pour simplifier dans le cas réel.

$$\text{Soit } V_{ij}(\theta) = \begin{matrix} & & \downarrow i & & \downarrow j \\ & & 1 & & \\ i \rightarrow & & 0 & \cos \theta & - \sin \theta \\ & & & 1 & \\ & & & & 1 \\ j \rightarrow & & 0 & \sin \theta & \cos \theta \\ & & & & 1 \\ & & & & & 1 \end{matrix} \quad \text{une matrice de rotation}$$

L'algorithme de Jacobi consiste à effectuer une suite de transmutations par de telles matrices en annulant successivement tous les termes en-dessous de la diagonale qui restent supérieurs en valeur absolue à un réel ϵ fixé.

Les valeurs propres se trouvent sur la diagonale. Les vecteurs propres sont obtenus en effectuant le produit des matrices de rotation utilisées.

On va étudier cet algorithme du point de vue du nombre d'appels de pages entraînés.

Soit $A' = v_{ij}(\theta) A v_{ij}(-\theta)$.

On a les formules suivantes :

$$a'_{ik} = \cos \theta a_{ik} - \sin \theta a_{jk}$$

$$a'_{jk} = \sin \theta a_{ik} + \cos \theta a_{jk}$$

$$a'_{ki} = \cos \theta a_{ki} - \sin \theta a_{kj}$$

$$a'_{kj} = \sin \theta a_{ki} + \cos \theta a_{kj}$$

$k \neq i, j$

$$a'_{ii} = \cos^2 \theta a_{ii} + \sin^2 \theta a_{jj} - \sin \theta \cos \theta (a_{ji} + a_{ij})$$

$$a'_{jj} = \cos^2 \theta a_{jj} + \sin^2 \theta a_{ii} + \sin \theta \cos \theta (a_{ji} + a_{ij})$$

$$a'_{ij} = \cos^2 \theta a_{ji} - \sin^2 \theta a_{ij} + \sin \theta \cos \theta (a_{ii} - a_{jj}) .$$

Pour annuler les termes a'_{ij} , il faut choisir θ tel que :

$$\cos^2 \theta a_{ji} - \sin^2 \theta a_{ij} + \sin \theta \cos \theta (a_{ii} - a_{jj}) = 0 .$$

Ce qui donne :

$$\sin = \frac{a}{\sqrt{b^2 + a^2}} , \quad \cos = \frac{b}{\sqrt{b^2 + a^2}}$$

avec $a = (a_{ii} - a_{jj}) + \sqrt{(a_{ii} - a_{jj})^2 + 4a_{ij}^2}$

$$b = 2a_{ij} .$$

h1) Algorithme de Jacobi normal.

On annule successivement à l'aide des matrices de rotation les termes en $(2,1), (3,1), \dots, (n,1), (3,2), \dots, (n,n-1)$ et on réitère le processus.

Le nombre d'appels de pages pour un cycle complet sera de l'ordre de :

$\frac{3q^2}{4} a^6$ appels pour le stockage en lignes

$\frac{3}{2} q^2 a^5$ appels pour le stockage en blocs.

On décompose maintenant deux modifications possibles de l'algorithme de Jacobi, dans le cas du stockage en lignes.

h2) Première modification.

On veut ne faire que des opérations sur les lignes. Pour cela, on peut remarquer que le but ultime de l'algorithme de Jacobi est l'obtention de la matrice orthogonale Q telle que :

$$D = Q A Q^T$$

D sera la matrice des valeurs propres, Q la matrice des vecteurs propres. Elle est obtenue comme produit de matrices de rotation élémentaires. Si l'on veut obtenir non seulement D mais Q , la modification suivante peut se justifier.

On fera : $A_{k+1} = V_k A_k$ $A_0 = A$, V_k étant la même matrice de rotation que dans l'algorithme ordinaire.
 $Q_{k+1} = V_k Q_k$

$(A_k Q_k^T = Q_k A_0 Q_k^T)$ et donc à la limite on obtient bien ainsi Q et la matrice DQ .

Pour annuler l'élément (i,j) au même pas de ce procédé, il faut faire un produit par une matrice $v_{ij}(\theta_k)$.

θ_k sera déterminée en fonction des éléments (i,j) , (j,j) et (i,i) de $Q_k A_k Q_k^T = A_k Q_k^T$: ces trois éléments seront donc obtenus comme produit des lignes i et i , j et j et i et j de A_k et Q_k respectivement.

Pour faire une itération complète, il faudra donc faire au maximum huit appels de pages (quatre au minimum). Pour un cycle complet, il faudra donc au maximum $4q^2 a^4$ appels de pages. Cependant, le nombre de multiplications est fortement augmenté.

h3) Deuxième modification.

On modifie l'ordre habituel des annulations de façon à faire le maximum de calculs par lignes appelées en mémoires. On peut par exemple effectuer les opérations résultant de deux transmutations v_{i_1, j_1} v_{i_2, j_2} successives en même temps si i_1, j_1, i_2, j_2 sont tous distincts. On peut au maximum regrouper $\lfloor \frac{n}{2} \rfloor$ telles transmutations et donc annuler $\lfloor \frac{n}{2} \rfloor$ éléments.

Comme il faut annuler dans un cycle complet $\frac{n(n-1)}{2}$ éléments, le nombre minimum de transmutations sera :

$$\left\lceil \frac{n(n-1)}{2 \lfloor \frac{n}{2} \rfloor} \right\rceil .$$

La question qui se pose est la suivante : existe-t-il une politique d'annulation ayant ce nombre minimum d'étapes ? Si oui, quelle est-elle ? On aurait alors besoin de

$$a^2 \times \frac{\lceil qa^2(qa^2-1) \rceil}{2 \lfloor \frac{qa^2}{2} \rfloor} \quad \text{appels de pages pour un cycle complet}$$

(qa^4 environ). Il faudrait évidemment $2 \times \lceil \frac{n}{2} \rceil$ mémoires supplémentaires pour stocker les $\cos \theta$ et $\sin \theta$ indispensables.

Sameh (24') a donné deux telles politiques (une pour n quelconque, l'autre pour $n = 2^k$). En utilisant l'une de ces politiques, l'algorithme de Jacobi converge toujours (on passe une infinité de fois sur chaque élément) et devient très performant au point de vue appels de pages.

On a vu dans le chapitre IV, que cette modification conduisait aussi à l'utilisation optimale du parallélisme dans ce calcul.

i) Algorithmes QR et LR

Pour ces algorithmes de calcul des valeurs propres, la forme d'Hessenberg est invariante. On peut donc transmuter la matrice A dont on cherche les valeurs propres sous forme d'Hessenberg, puis on applique l'algorithme QR ou LR.

Soit H la matrice d'Hessenberg.

On cherche Q sous forme d'un produit de matrices de rotations telles que $QH = R$ triangulaire supérieure.

Cette étape nécessite a^2 appels de pages.

Ensuite, on effectue RQ. Pour cette étape, on peut regrouper les calculs par lignes de façon à ne nécessiter aussi que a^2 appels de pages en tout.

On a donc au total $2a^2$ appels de pages (au lieu de $2^2 + (qa^2 - 1)a^2$).

L'algorithme LR peut être modifié de la même manière.

CONCLUSION

En passant en revue les principaux algorithmes de calcul matriciel, on a mis en évidence la grande influence de l'organisation des données sur leur coût en appels de pages et on a montré comment réduire ce coût de façon radicale en modifiant l'ordre des calculs et en combinant certaines étapes.

On peut se demander si ce genre de modifications peut être pris en charge par le compilateur ou si le programmeur ne devrait pas avoir plus de connaissances du fonctionnement réel de la machine. Un certain "style de programmation" peut être conseillé pour programmer en langage de haut niveau dans un environnement paginé (36) de façon d'une part à améliorer son propre programme, mais aussi le fonctionnement global du système.

Ce qui précède avait déjà fait l'objet d'un séminaire (41). Depuis l'aspect mathématique du problème de l'implémentation optimale d'un programme a été étudié par LACOLLE (39), (40).

Les références (27), (29), (30), (31), (34), (35) concernent l'étude théorique et pratique du fonctionnement global des systèmes avec pagination.

REFERENCES SUR LE CHAPITRE CV

- (27) BARD, Y., "Characterization of program paging in a time sharing environment".
I.B.M. J. Res. Developp.(Sept.1973), 387-393.
- (28) BELADY, CA., "A case study of replacement algorithm for a virtual storage computer".
I.B.M. System Journal, vol.5, n° 2, (1966), 78-101.
- (29) BELADY, CA., PALERMO, FP., "On line measurement of paging behavior by the multivalued MIN Algorithm".
I.B.M. J. Res. Developp, (January 1974), 2-19.
- (30) CHAMBERLIN, DD., FULLER, SH., LIU, LY.,
"An analysis of page allocation strategies for multiprogramming systems with virtual memory".
I.B.M. J. Res. Developp. (september 1973), 404-412.
- (31) CHOW, CK. "On optimization of storage hierarchies".
I.B.M. J. Res. Developp. (May 1974), 194-203.
- (32) COFFMAN, MCKELLAR, " Organizing matrices and matrix operations for paged memory systems".
Comm. ACM, vol 12, n° 3 (March 1969), 153-165.
- (33) DENNING, PJ., " Virtual memory".
Computing surveys, vol.2, n°3, (sept. 1970), 152-189.
- (34) BENSEL, SLUTZ, TRAIGER, "Evaluation techniques for storage hierarchies".
I.B.M. System Journal n°2, (1970), 78-117.
- (35) HEINBE, E., A unified approach to the evaluation of a class of replacement algorithms".
IEEE Trans. on E.C., vol.C22, n°6, (June 1973), 611-618.

- (36) GUERTIN, RC., "Programming in a paging environment".
Datamation, (February 1972), 50-55.
- (37) HATFIELD, DJ., "Experiments on page size, program access patterns,
and virtual memory performance".
I.B.M. J. Res. Develop. (January 1972), 58-66.
- (38) KANEKO, T., "Optimal task switching policy for a multilevel storage
system".
I.B.M. J. Res. Developp. (July 1974), 310-314.
- (39) LACOLLE, B., "Aspects théoriques et pratiques du problème du parti-
tionnement". Séminaire d'Analyse Numérique, Grenoble, n° 211,
(novembre 1974).
- (40) LACOLLE, B., "Bi-partitionnement d'une matrice du type somme de deux
antisymétriques". Séminaire d'Analyse Numérique,
Grenoble, n° 232, (octobre 1975).
- (41) LAFON, J.C., "Influence de la pagination sur la rapidité d'exécution
des algorithmes". Séminaire d'Analyse Numérique, Grenoble, n° 190,
(janvier 1974).
- (24') SAMEH, A.H., "On Jacobi and Jacobi like algorithms, for a parallel
computer". Mathematics of computation, vol.25, N° 115,
(july 1971), 579-590.
- (7') WILKINSON, JH., "The algebraic eigenvalue problem".
Clarendon press. Oxford, (1968).

ERRATA

Page A 8 ; 3e ligne, il faut remplacer X et Y par x et y .

Page A11 ; milieu de la page, Dans la phrase : "On désigne par \bar{M} (resp. M)..." ,
il faut remplacer "opérations mathématiques" par "opérations
arithmétiques-".

Page A34 ; référence 130, il faut lire "boolean" à la place de "bloolean".

Page B 8 ; dans la remarque 1, il faut lire (p un nombre premier) au lieu de
(p le nombre premier).

Page B9, B10 ; remplacer X_i et Y_j par x_i et y_j , et p_1, \dots, p_p par P_1, \dots, P_p
(dans la remarque 4).

Page B20 ; dix lignes avant la fin, il faut lire :

$$X^t B_i Y \equiv X^t \left(\sum_{j=1}^q \alpha_j^i (A_1^j B_2^{jt} + A_2^j B_1^{jt}) \right) Y ,$$

et au-dessus :

$$Y^t \left(\sum_{j=1}^q \alpha_j^i B_1^j B_2^{jt} \right) Y \equiv 0 \quad (\text{au lieu de } X^t (\quad) Y \equiv 0).$$

Page B24 ; paragraphe 2, il faut lire :

$$\langle j \rangle \equiv A_2^k X + B_2^k Y , \quad A_1^k, A_2^k \in K^m , \quad B_1^k, B_2^k \in K^n$$

Page B36 ; dans le corollaire 2, il faut remplacer min par max et dans la
propriété 10 Z_1, \dots, Z_p par z_1, \dots, z_p .

Page B43 ; dans l'expression de B_i^1 et B_i^2 remplacer B_j par A_j'

Page B46 ; remplacer z_1, \dots, z_4 par Z_1, \dots, Z_4

Page B57 ; il faut écrire $Z^t = (Z_0, Z_1, \dots, Z_{n-1})$ et dans la ligne suivante il
faut remplacer z_k par Z_k

Page B60, B62 ; remplacer $H_n(\mathcal{E}_n)$ par $H^n(\mathcal{E}^n)$

Page B81 ; dans S' remplacer s_{21} par s_{12} .

Le résultat du corollaire 1 est $n \frac{(n+1)}{2} - q$.

Page 123 ; remplacer $W_2^t W = 0$ par $W_2^t Z = 0$.

Page 130 ; dans l'énoncé du théorème 8, cinq doit être remplacé par six.

Page C 6 ; dans l'expression de $A(\alpha, \beta)$ il faut remplacer β_k par β_l

Page C11 ; dans l'expression de A^{-1} (propriété 6) remplacer P_1, \dots, P_{n-1} par P_1', \dots, P_{n-1}'

Page C22 ; ligne 4, lire G_i/G_{i+1} au lieu de $G_i + G_{i+1}$

7 lignes en-dessous lire seulement

$$g_{n-k}^{m_{n-k}-1} \cdot G_{n-k+1}$$

Page C64 ; dans la proposition 5 changer avant en après.

A la fin de la page (Deuxième procédé), il faut lire :
pour obtenir un zéro en (k, i) (au lieu de (i, k)) et ensuite
 $E_{k, i+1}$ au lieu de $E_{k, k+1}$.

Page C75 ; dans (1), remplacer $I(n) \leq D_\infty(n-1)$ par $I_\infty(n) \leq D_\infty(n)+1$

Page C80 ; début de la page, lire $A'_{kj} = A_{k,j} - a A_{ki}$ (au lieu de
 $A'_{ki} = A_{k,j} - a^2 A_{ki}$).

Page C106 ; changer $A'_{i-1, j}$ et $A_{j, i-1}$ en $a_{i-1, j}$ et $a_{j, i-1}$.