



Quelques algorithmes systoliques pour le calcul scientifique

Yves Robert

► To cite this version:

Yves Robert. Quelques algorithmes systoliques pour le calcul scientifique. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1982. Français. NNT : . tel-00305305

HAL Id: tel-00305305

<https://theses.hal.science/tel-00305305>

Submitted on 23 Jul 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

l' Université Scientifique et Médicale de Grenoble

pour obtenir le grade de

DOCTEUR DE 3ème CYCLE

Mathématiques Appliquées

par

Yves ROBERT



QUELQUES ALGORITHMES SYSTOLIQUES

POUR LE CALCUL SCIENTIFIQUE.



Thèse soutenue le 16 décembre 1982 devant la commission d'examen.

N. GASTINEL Président

F. ANCEAU

C. BELLISSANT

J. DELLA DORA Examineurs

P. QUINTON

F. ROBERT

THESE

présentée à

l' Université Scientifique et Médicale de Grenoble

pour obtenir le grade de

DOCTEUR DE 3ème CYCLE

Mathématiques Appliquées

par

Yves ROBERT



QUELQUES ALGORITHMES SYSTOLIQUES

POUR LE CALCUL SCIENTIFIQUE.



Thèse soutenue le 16 décembre 1982 devant la commission d'examen.

N. GASTINEL Président

F. ANCEAU

C. BELLISSANT

J. DELLA DORA Examineurs

P. QUINTON

F. ROBERT

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

année scolaire 1980-1981

Président de l'Université : M. J.J. PAYAN

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

PROFESSEURS DE 1ère CLASSE

Mlle	AGNIUS DELORD Claudine	Biophysique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Clinique dermatologie
	AMBROISE THOMAS Pierre	Parasitologie
	ARNAUD Paul	Chimie
	ARVIEU Robert	Physique nucléaire
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale A
	BEAUDOING André	Clinique pédiatrie et puériculture
	BELORISKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
Mme	BERIEL Hélène	Pharmacodynamie
M.	BERNARD Alain	Mathématiques pures
Mme	BERTRANDIAS Françoise	Mathématiques pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques pures
	BEZES Henri	Clinique chirurgicale & traumatologie
	BILLET Jean	Géographie
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET EYMARD Joseph	Clinique Hépto-gastro-entérologie
Mme	BONNIER Jane-Marie	Chimie générale
MM.	BOUCHERLE André	Chimie et toxicologie
	BOUCHET Yves	Anatomie
	BOUCHEZ Robert	Physique nucléaire
	BRAVARD Yves	Géographie

.../...

MM. BUTEL Jean	Orthopédie
CABANEL Guy	Clinique rhumatologie et hydrologie
CARLIER Georges	Biologie végétale
CAU Gabriel	Médecine légale et toxicologie
CAUQUIS Georges	Chimie organique
CHARACHON Robert	Clinique O.R.L.
CHATEAU Robert	Clinique neurologique
CHIBON Pierre	Biologie animale
COEUR André	Chimie analytique et bromotologique
COUDERC Pierre	Anatomie pathologique
CRABBE Pierre	C.E.R.M.O.
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude	M.I.A.G.
DELORMAS Pierre	Pneumo-phtisiologique
DENIS Bernard	Clinique cardiologique
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DODU Jacques	Mécanique appliquée IUT 1
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique
GASTINEL Noël	Analyse numérique
GAVEND Jean-Michel	Pharmacologie
GEINDRE Michel	Electro-radiologie
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
JANIN Bernard	Géographie
JEANNIN Charles	Pharmacie galénique
JOLY Jean-René	Mathématiques pures
KAHANE André	Physique
KAHANE Josette	Physique
KLEIN Joseph	Mathématiques pures
KOSZUL Jean-Louis	Mathématiques pures
LACAZE Albert	Hermodynamique
LACHARME Jean	Biologie cellulaire
LAJZEROWICZ Joseph	Physique

Mme	LAJZEROWICZ Jeannine	Physique
MM.	LATREILLE René	Chirurgie thoracique
	LATURAZE Jean	Biochimie pharmaceutiques
	LAURENT Pierre	Mathématiques appliquées
	LE NOC Pierre	Bactériologie virologie
	LLIBOUTRY Louis	Géophysique
	LOISEAUX Jean-Marie	Sciences nucléaires
	LOUP Jean	Géographie
	LUU DUC Cuong	Chimie générale et minérale
	MALINAS Yves	Clinique obstétricale
Mlle	MARIOTTE Anne-Marie	Pharmacognostie
MM.	MAYNARD Roger	Physique du solide
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	NEGRE Robert	Mécanique IUT 1
	MOZIERES Philippe	Spectrométrie physique
	OMONT Alain	Astrophysique
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY PEYROULA Jean-Claude	Physique
	PERRET Jean	Sémeiologie médicale (neurologie)
	PERRIER Guy	Géophysique
	PIERRARD Jean-Marie	Mécanique
	RACHAIL Michel	Clinique médicale B
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
Mme	RENAUDET Jacqueline	Bactériologie
M.	REVOL Michel	Urologie
Mme	RINAUDO Marguerite	Chimie CERMAV
MM.	DE ROUGEMONT Jacques	Neuro-chirurgie
	SARRAZIN Roger	Clinique chirurgicale B
Mme	SEIGLE MURANDI Françoise	Botanique et cryptogamie
MM.	SENGEL Philippe	Biologie animale
	SIBILLE Robert	Construction mécanique IUT 1
	SOUTIF Michel	Physique
	TANCHE Maurice	Physiologie
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire

MM. VAN CUTSEM Bernard
 VAUQUOIS Bernard
 VERAÏN Alice
 VERAÏN André
 VIGNAIS Pierre

Mathématiques appliquées
 Mathématiques appliquées
 Pharmacie galénique
 Biophysique
 Biochimie médicale

PROFESSEURS DE 2ème CLASSE

MM. ARNAUD Yves
 AURIAULT Jean-Louis
 BEGUIN Claude
 BOITET Christian
 BOUTHINON Michel
 BRUGEL Lucien
 BUISSON Roger
 CASTAING Bernard
 CHARDON Michel
 CHEHIKIAN Alain
 COHEN Henri
 COHENADDAD Jean-Pierre
 COLIN DE VERDIERE Yves
 CONTE René
 CYROT Michel
 DEPASSEL Roger
 DOUCE Roland
 DUFRESNOY Alain
 GASPARD François
 GAUTRON René
 GIDON Maurice
 GIGNOUX Claude
 GLENAT René
 GOSSE Jean-Pierre
 GROS Yves
 GUITTON Jacques
 HACQUES Gérard
 HERBIN Jacky
 HICTER Pierre
 IDELMAN Simon
 JOSELEAU Jean-Paul
 JULLIEN Pierre
 KERCKOVE Claude

Chimie IUT 1
 Mécanique IUT 1
 Chimie organique
 Mathématiques appliquées
 E.E.A. IUT 1
 Energétique IUT 1
 Physique IUT 1
 Physique
 Géographie
 E.E.A. IUT 1
 Mathématiques pures
 Physique
 Mathématiques pures
 Physique IUT 1
 Physique du solide
 Mécanique des fluides
 Physiologie végétale
 Mathématiques pures
 Physique
 Chimie
 Géologie
 Sciences nucléaires
 Chimie organique
 E.E.A. IUT 1
 Physique IUT 1
 Chimie
 Mathématiques appliquées
 Géographie
 Chimie
 Physiologie animale
 Biochimie
 Mathématiques appliquées
 Géologie

MM.	KRAKOWIACK Sacha	Mathématiques appliquées
	KUHN Gérard	Physique IUT 1
	KUPKA Yvon	Mathématiques pures
	LUNA Domingo	Mathématiques pures
	MACHE Régis	Physiologie végétale
	MARECHAL Jean	Mécanique
	MICHOULIER Jean	Physique IUT 1
Mme	MINIER Colette	Physique IUT 1
MM.	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique IUT 1
	OUDET Bruno	Mathématiques appliquées
	PEFFEN René	Métallurgie IUT 1
	PELMONT Jean	Biochimie
	PERRAUD Robert	Chimie IUT 1
	PERRIAUX Jean-Jacques	Géologie minéralogie
	PERRIN Claude	Sciences nucléaires
	PFISTER Jean-Claude	Physique du solide
	PIERRE Jean-Louis	Chimie organique
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	Mathématiques appliquées
	RICHARD Lucien	Biologie végétale
	ROBERT Gilles	Mathématiques pures
	ROBERT Jean-Bernard	Chimie physique
	ROSSI André	Physiologie végétale
	SAKAROVITCH Michel	Mathématiques appliquées
	SARROT REYNAUD Jean	Géologie
	SAXOD Raymond	Biologie animale
Mme	SOUTIF Jeanne	Physique
MM.	STUTZ Pierre	Mécanique
	VIALON Pierre	Géologie
	VIDAL Michel	Chimie organique
	VIVIAN Robert	Géographie

CHARGES D'ENSEIGNEMENT PHARMACIE

MM.	ROCHAS Jacques	Hygiène et hydrologie
	DEMENGÉ Pierre	Pharmacodynamie

PROFESSEURS SANS CHAIRE (médecine)

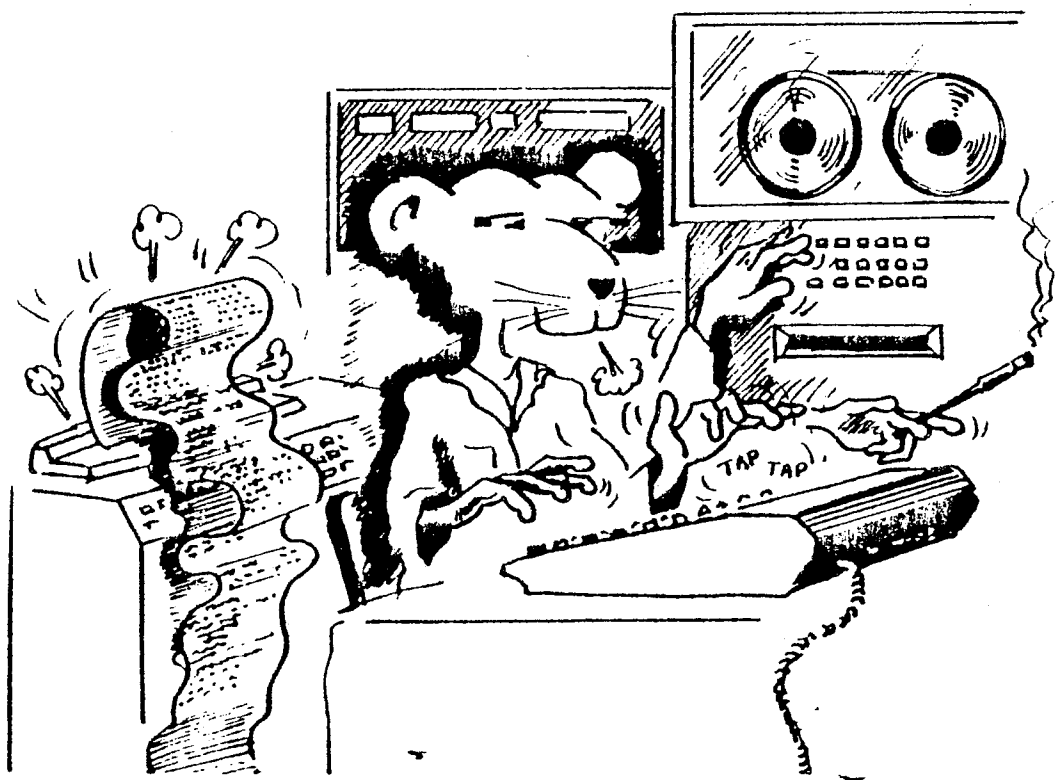
M.	BARGE Michel	Neuro-chirurgie
----	--------------	-----------------

MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie
	CHAMBAZ Edmond	Biochimie (hormonologie)
	CHAMPETIER Jean	Anatomie
	COLOMB Maurice	Biochimie
	COULOMB Max	Radiologie
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GROULADE Joseph	Biochimie A
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Gérontologie
	JALBERT Pierre	Histologie
	MAGNIN Robert	Hygiène
	PHELIP Xavier	Rhumatologie
	REYMOND Jean-Charles	Chirurgie générale
	STIEGLITZ Paul	Anesthésiologie
	VROUSOS Constantin	Radiothérapie

MAITRES DE CONFERENCES AGREGES (médecine)

MM.	BACHELOT Yvan	Endocrinologie
	BENABID Alim Louis	Médecine et chirurgie
	BERNARD Pierre	Gynécologie obstétrique
	CONTAMIN Charles	Chirurgie thoracique
	CORDONNIER Daniel	Néphrologie
	CROUZET Guy	Radiologie
	DEBRU Jean-Luc	Médecine interne
	DYON Jean-François	Chirurgie infantile
	FAURE Claude	Anatomie et organogénèse
	FAURE Gilbert	Urologie
	FLOYRAC Roger	Biophysique
	FOURNET Jacques	Hépatogastro-entérologie
	GAUTIER Robert	Chirurgie générale
	GIRARDET Pierre	Anesthésiologie
	GUIDICELLI Henri	Chirurgie générale
	GUIGNIER Michel	Thérapeutique (réanimation)
	JUNIEN-LAVILLAULOY Claude	Clinique O.R.L.
	KOLODIE Lucien	Hématologie biologique
	MALLION Jean-Michel	Médecine du travail
	MASSOT Christian	Médecine interne
	MOUILLON Michel	Ophtalmologie

MM.	PARAMELLE Bernard	Pneumologie
	RACINET Claude	Gynécologie-Obstétrique
	RAMBAUD Pierre	Pédiatrie
	RAPHAEL Bernard	Stomatologie
	SCHAEFER René	Cancérologie
	SEIGNEURIN Jean-Marie	Bactériologie-virologie
	SOTTO Jean-Jacques	Hématologie
	STOEBNER Pierre	Anatomie-pathologique



A Sylvie

J'exprime toute ma reconnaissance à

Monsieur Noël GASTINEL pour l'honneur qu'il me fait en présidant le jury, pour les discussions toujours enrichissantes que j'ai pu avoir avec lui

Monsieur François ROBERT, qui a dirigé cette thèse, pour m'avoir chaleureusement accueilli à Grenoble, pour ses nombreux conseils et encouragements qui ont rendu possible ce travail .

J'adresse mes vifs remerciements aux membres du jury :

Monsieur François ANCEAU, qui m'a initié à Chalmazel à la réalisation des circuits intégrés

Monsieur Camille BELLISSANT, qui est à l'origine d'une partie de ce travail

Monsieur Jean DELLA DORA, qui a introduit l'algorithme systolique dans l'équipe d'Analyse Numérique

Monsieur Patrice QUINTON, qui a bien voulu faire le déplacement depuis Rennes à cette occasion .

Je voudrais également remercier

Monsieur François HIRSCH, Directeur d'Etudes à l'Ecole Normale Supérieure de l'Enseignement Technique, pour son soutien tout au long de ma scolarité

Monsieur Pierre AMBLARD, pour m'avoir permis de suivre une formation sur la conception de circuits intégrés à l'I.N.P.G.

Je souhaite témoigner de l'amitié qui me relie aux membres du groupe de travail "Comportement d'Itérations", en particulier Michel COSNARD, André EBERHARD, Eric GOLES et Maurice TCHUENTE .

Mes remerciements vont enfin à tous les membres du Service de Reprographie de l'I.M.A.G. pour l'excellente qualité de leur travail .

SOMMAIRE

INTRODUCTION

CHAPITRE 1 : RESEAUX SYSTOLIQUES

Sous-chapitre 1.1 : Principe des réseaux systoliques

Sous-chapitre 1.2 : Etude d'un exemple

Sous-chapitre 1.3 : Les cellules élémentaires

CHAPITRE 2 : EXEMPLES DE RESEAUX UTILISANT DES CELLULES A MODE DE CONNEXION ORTHOGONAL

Sous-chapitre 2.1 : Récurrences vectorielles

Sous-chapitre 2.2 : Modélisation d'évolutions discrètes
issues de l'automatique

Sous-chapitre 2.3 : Calcul polynomial

Sous-chapitre 2.4 : Convolution bidimensionnelle

CHAPITRE 3 : EXEMPLES DE RESEAUX UTILISANT DES CELLULES A MODE DE CONNEXION HEXAGONAL

Sous-chapitre 3.1 : Factorisation incomplète de
matrices-bandes

Sous-chapitre 3.2 : Algorithme de la médiane

CHAPITRE 4 : SIMULATION SEQUENTIELLE D'UN ALGORITHME SYSTOLIQUE

Sous-chapitre 4.1 : Contexte de la simulation

Sous-chapitre 4.2 : Listings

CHAPITRE 5 : CONCEPTION ET DESSIN DES MASQUES D'UN CIRCUIT INTEGRE POUR LE LISSAGE PAR MEDIANE

Sous-chapitre 5.1 : Aspect logique

Sous-chapitre 5.2 : Aspect temporel

Sous-chapitre 5.3 : Dessin des masques

CONCLUSION

ANNEXES

Annexe 1 : "Organisation du calcul en parallèle de
réurrences vectorielles"

Annexe 2 : "Systolic algorithms for convolution and
median smoothing"

Annexe 3 : "Regular incomplete factorizations of real
positive definite matrices"

REFERENCES

INTRODUCTION

Grâce aux progrès de la technologie actuelle, il est devenu possible de concevoir et de réaliser des circuits intégrés à très haute densité de composants.

Bien mieux, à la condition que les coûts dominants - conception et test - soient réduits par le choix d'une configuration géométrique régulière, la production de circuits spécialisés (ie dont le champ d'application reste limité) peut elle-même devenir concurrentielle.

Devant une telle évolution, il importe, pour le mathématicien, de reconsidérer les algorithmes usuels afin de leur donner une structure à fort degré de parallélisme qui soit adaptée aux exigences de la nouvelle technologie .

L'adéquation à ces exigences du modèle systolique introduit par H.T.KUNG est si remarquable que de très nombreux réseaux systoliques ont été proposés dans les domaines les plus divers, allant du traitement du signal ([09],[15],[16],[22]) à la reconnaissance de la parole ([05],[10]) , en passant par l'arithmétique matricielle ([01],[06],[12],[21]) et les bases de données ([20],[23]).

Une présentation générale du modèle systolique fait l'objet du premier chapitre. Dans les deux chapitres suivants sont étudiés de nouveaux algorithmes systoliques, conçus pour le calcul algébrique et le traitement du signal. Le chapitre quatre est consacré à la simulation sur micro-ordinateur de l'algorithme de la médiane (modélisé au chapitre trois), tandis que le chapitre cinq constitue le premier pas vers une implémentation sur silicium : il s'agit du dessin de masques, à l'aide d'outils de C.A.O., du réseau de la médiane.

Par ailleurs, l'examen détaillé de certains algorithmes des chapitres deux et trois a été renvoyé aux annexes un et deux. Enfin, l'annexe trois donne un cadre théorique pour

l'étude des factorisations incomplètes (algorithme du chapitre trois).

NOTE : Les résultats des chapitres 2.1,2.2,2.4 et du chapitre 3 ont été établis en collaboration avec M.TCHUENTE et sont à paraître. Le chapitre 5 résume un travail conduit avec Y.CLAUZEL qui sera détaillé dans un rapport de recherche IMAG. Les résultats théoriques de l'annexe trois sont publiés.

CHAPITRE 1

TITRE : RESEAUX SYSTOLIQUES

RESUME : On présente le modèle systolique. L'étude d'un exemple permet d'introduire les notions de parallélisme temporel (pipe-line) et d'expliquer pourquoi la rapidité de calcul n'est pas limitée par la vitesse d'acquisition des données. On définit enfin les cellules élémentaires.

Sous-chapitre 1.1:

Principe des réseaux systoliques

Un système systolique est un modèle introduit par H.T.KUNG et est constitué par un réseau de processeurs (ou cellules)

- qui ont tous la même structure
- qui sont connectés entre eux
- qui, au même rythme, effectuent des opérations élémentaires sur les variables en provenance des cellules voisines en entrée, puis envoient celles-ci vers les cellules voisines en sortie.

Ainsi chaque cellule a-t-elle un fonctionnement semblable à celui du coeur humain, d'où la terminologie.

La première étape dans la conception d'un algorithme systolique pour le calcul d'une fonction

$$x = (x_1, \dots, x_m) \rightarrow (f_1(x), \dots, f_n(x))$$

consiste à mettre chaque f_i sous la forme

$$f_i(x) = b_i + \sum_{(j_1, \dots, j_r) \in V_i} g_{i, j_1, \dots, j_r}(x_{j_1}, x_{j_2}, \dots, x_{j_r})$$

On cherche alors un réseau où chaque cellule a $r+1$ entrées et $r+1$ sorties :

- toutes les variables-données x_j circulent d'une cellule à l'autre sans changer de valeur
- chaque variable-résultat y_i , initialisée à b_i , circule de manière à rencontrer tous les r -uplets dont elle a besoin, et à la rencontre de $(x_{j_1}, x_{j_2}, \dots, x_{j_r}), (j_1, \dots, j_r) \in V_i$, on a

$$y_i := y_i + g_{i, j_1, \dots, j_r}(x_{j_1}, \dots, x_{j_r})$$

y_i se calcule donc par accumulations successives .

En résumé, pour calculer une fonction f , on la décompose en opérations élémentaires pouvant être effectuées simultanément: on obtient ainsi le parallélisme dans l'espace. L'obtention du parallélisme dans le temps (Pipe-line) sera analysée à l'aide de l'exemple présenté ci-dessous.

Auparavant, nous avons besoin de préciser la définition que nous adoptons pour l'efficacité d'un réseau systolique :

soit (P) un problème donné. On note $t(k)$ le temps nécessaire à la résolution de (P) sur un réseau comportant k processeurs. Une bonne mesure de l'efficacité du parallélisme est donnée par la quantité

$$e = \frac{t(1)}{k \cdot t(k)}$$

Comme on peut toujours simuler sur un seul processeur en un temps $k \cdot t(k)$ un algorithme à k processeurs, on vérifie que $e \leq 1$.

Sous-chapitre 1.2:

Etude d'un exemple [17]

Le calcul de la suite réelle

$$y^i = a x^i + b x^{i-1} + c x^{i-2}$$

à partir de la suite de données (x^i) , $i \geq 0$, se modélise comme indiqué figure 1a. On explicite quelques pulsations successives à la figure 1b, pour mettre en relief le croisement du flot des variables-données avec celui des variables-résultats .

Moyennant l'ajout d'un processeur de retard, le même réseau peut être utilisé pour l'évaluation de la récurrence réelle

$$x^{i+1} = a x^i + b x^{i-1} + c x^{i-2} ,$$

x^0, x^1, x^2 donnés (figure 1c) .

Dans les deux cas, l'efficacité est $e = 1/2$ car chaque cellule est active une pulsation sur deux. De même, ces deux réseaux illustrent la possibilité de commencer le calcul d'une variable-résultat y_{i+1} avant d'avoir terminé celui de y_i : en fait l'exemple a valeur de règle, la structure des réseaux systoliques conduit naturellement au parallélisme dans le temps .

D'autre part, plusieurs calculs sont effectués par chaque variable-donnée x_j qui entre dans le réseau : ainsi la performance de l'ensemble n'est pas limitée par la vitesse d'acquisition des données - c'est souvent là le goulot d'étranglement d'un processeur - mais bien par la rapidité de calcul du circuit. Ici encore, cette remarque reste valable pour toutes les architectures systoliques .

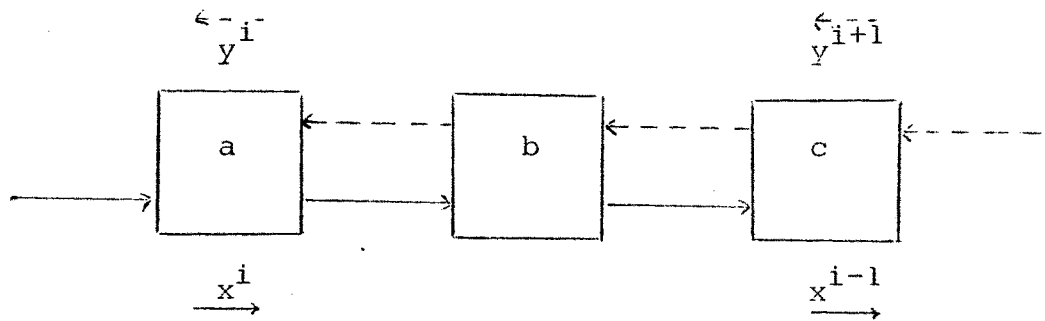
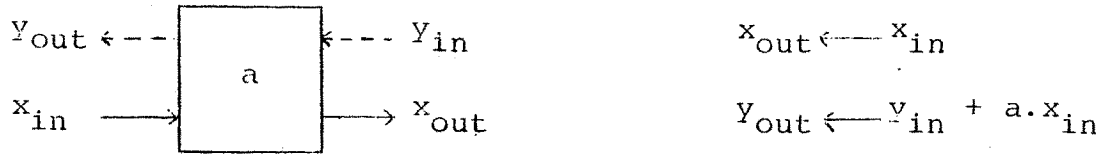


FIGURE 1a

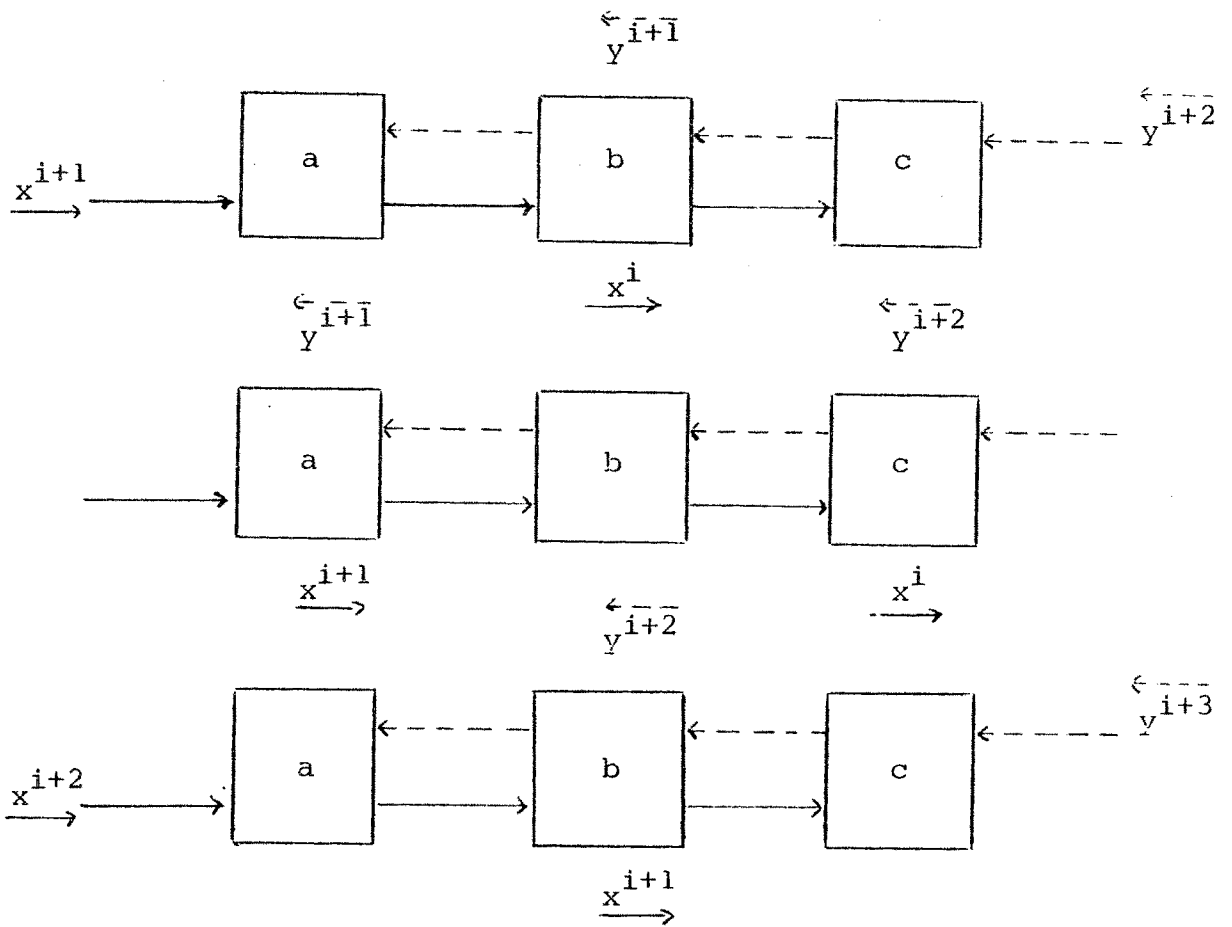


FIGURE 1b



Processeur de retard

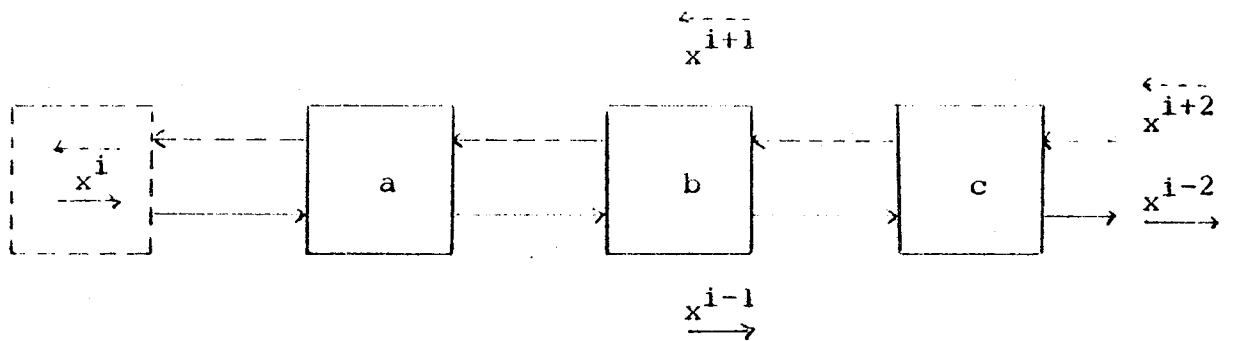


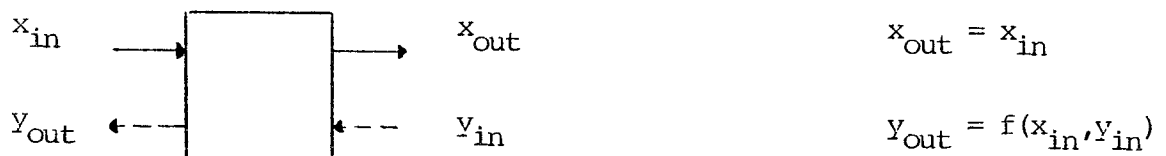
FIGURE 1c

Sous chapitre 1.3:

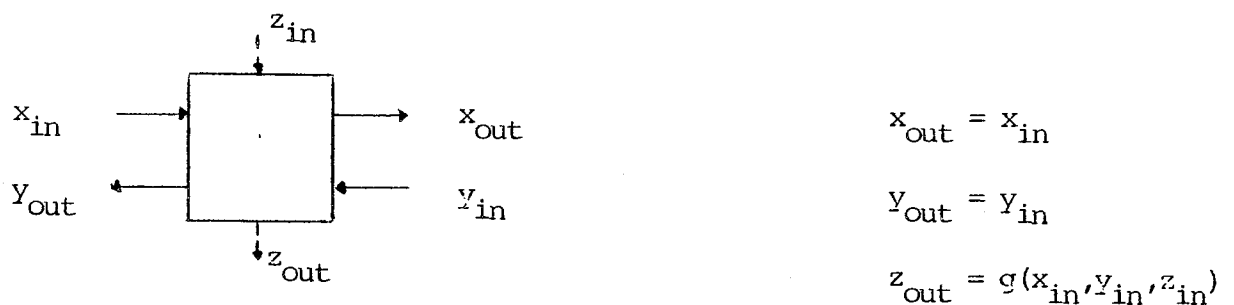
Les cellules élémentaires

On distingue deux types de cellules élémentaires, suivant que l'on utilise un seul ou deux flots de variables-données :

- type 1 : mode de connexion orthogonal



- type 2 : mode de connexion hexagonal



Le choix le plus simple est

$$f(x, y) = y + a x \text{ (exemple précédent)}$$

$$g(x, y, z) = z + x y$$

Chaque type de cellule est adapté à la modélisation de problèmes spécifiques; chacun des deux chapitres suivants illustre l'utilisation de l'un de ces deux types de cellules élémentaires .

CHAPITRE 2

TITRE : EXEMPLES DE RESEAUX UTILISANT DES CELLULES A MODE DE CONNEXION ORTHOGONAL

RESUME : On présente dans ce chapitre des réseaux dont le champ d'application relève essentiellement du domaine du calcul numérique :

- récurrences vectorielles
- évolutions discrètes issues de l'automatique
- calcul polynomial
- convolution bidimensionnelle

La plupart des réseaux obtenus délivrent leurs sorties en temps réel. Quand tel n'est pas le cas (récurrences vectorielles), l'étude de la complexité de l'algorithme est détaillée .

Sous chapitre 2.1 :

Réurrences vectorielles

L'objectif est de généraliser à la dimension n les résultats obtenus par H.T.KUNG en dimension un : on a vu au chapitre 1 que le calcul d'une suite réelle avait la même complexité que le calcul récurrent; il n'en est pas ainsi pour les dimensions deux ou plus .

Résumons les résultats exposés dans l'annexe 1 :

A) Calcul en temps réel d'une suite vectorielle

soient A, B deux matrices carrées d'ordre n . Etant donnée une suite $(x^i)_{i \geq 0}$ de vecteurs à n composantes, on veut calculer la suite $(y^i)_{i \geq 1}$, où

$$y^i = A x^i + B x^{i-1}$$

à l'aide d'un réseau comportant $2n^2$ cellules (puisque celles-ci n'effectuent que des opérations scalaires). Un tel réseau est explicité à la figure 1 (annexe 1) pour $n = 3$. Son efficacité est $e = 1/2$.

La généralisation au calcul de

$$y^{i+p-1} = A(p-1)x^{i+p-1} + \dots + A(1)x^{i+1} + A(0)x^i$$

est immédiate .

B) Réurrence linéaire à p termes en dimension n

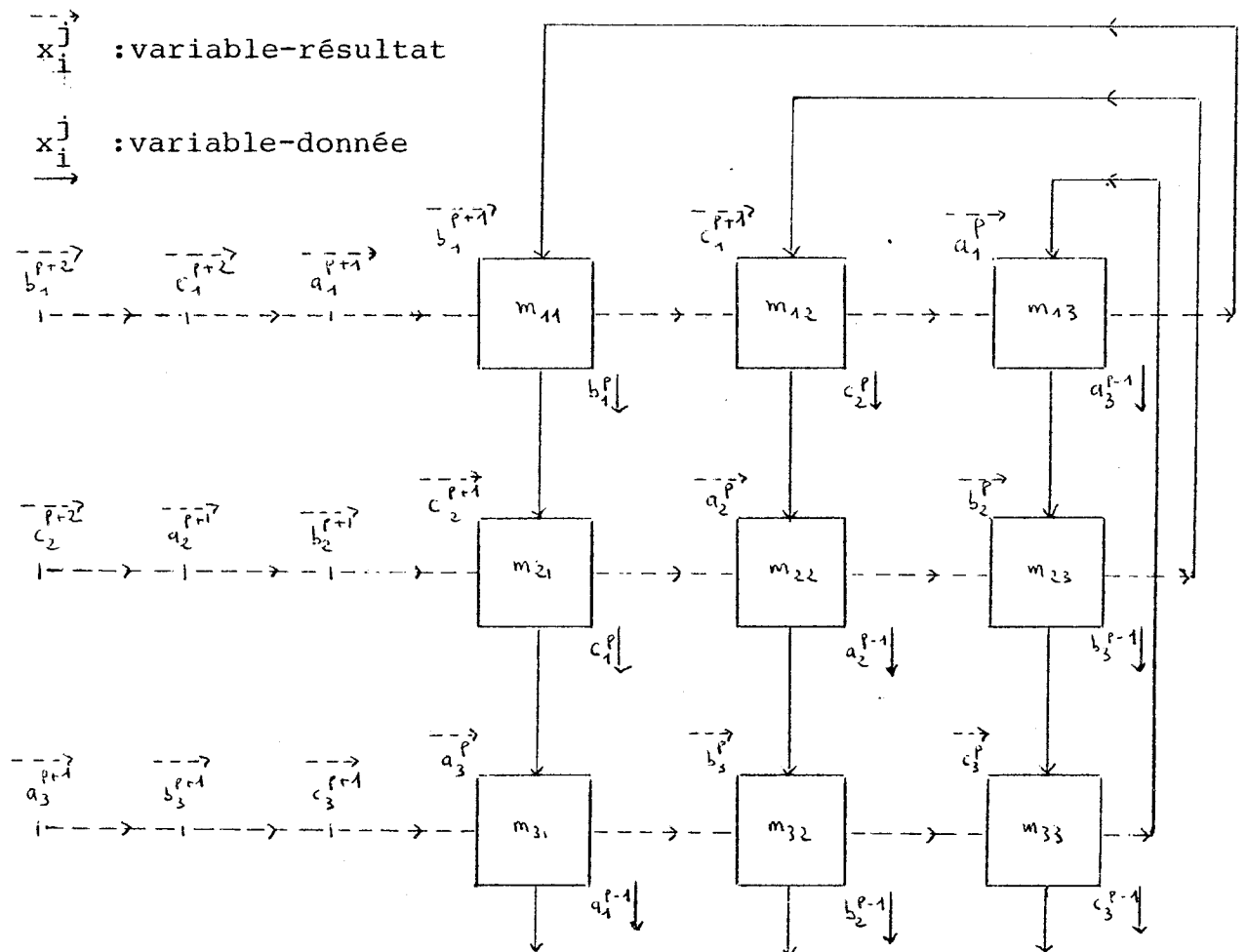
soient A, B deux matrices carrées d'ordre n , x^0 et x^1 deux vecteurs à n composantes. On veut calculer la suite $(x^i)_{i \geq 1}$, où

$$x^{i+1} = A x^i + B x^{i-1}$$

- l'efficacité e d'un réseau solution satisfait à $e \leq 1/n+1$

- on a des estimations plus précises en faisant des hypothèses sur l'entrée des composants des itérés successifs dans le réseau

- un mode d'approche analytique assure d'une solution régulière cylindrique d'efficacité $e = 1/2n$. Une telle solution est représentée figure 8 (annexe 1) pour $n = 3$.



M est une matrice 3×3 , de vecteurs colonnes A,B,C.

On note

$$M^p = (A^p, B^p, C^p)$$

On note $X = (x_1, x_2, x_3)$ un vecteur X à 3 composantes.

Calcul des puissances successives
d'une matrice

Figure 1

Sous-chapitre 2.2 :

Modélisation d'évolutions discrètes issues de l'automatique

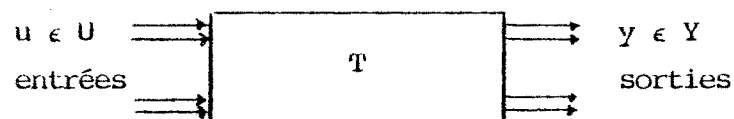
A) Calcul des puissances successives d'une même matrice

les réseaux proposés au sous-chapitre précédent permettent de résoudre le problème du calcul des puissances successives d'une même matrice M d'ordre n (fréquent en automatique) : l'idée est de superposer sur un même réseau le calcul des n récurrences vectorielles $X_i^{p+1} = M X_i^p$, $p \geq 0, 1 \leq i \leq n$, correspondant à chaque vecteur-colonne X_i de M . On obtient ainsi le réseau de la figure 1, qui est d'efficacité $e = 1$.

B) Cas d'un système discret

considérons un système linéaire du premier ordre, discret: il est constitué par un espace d'entrées noté U , de dimension m , par un espace de sorties noté Y , de dimension p , et par un espace d'états du système, noté X , de dimension n .

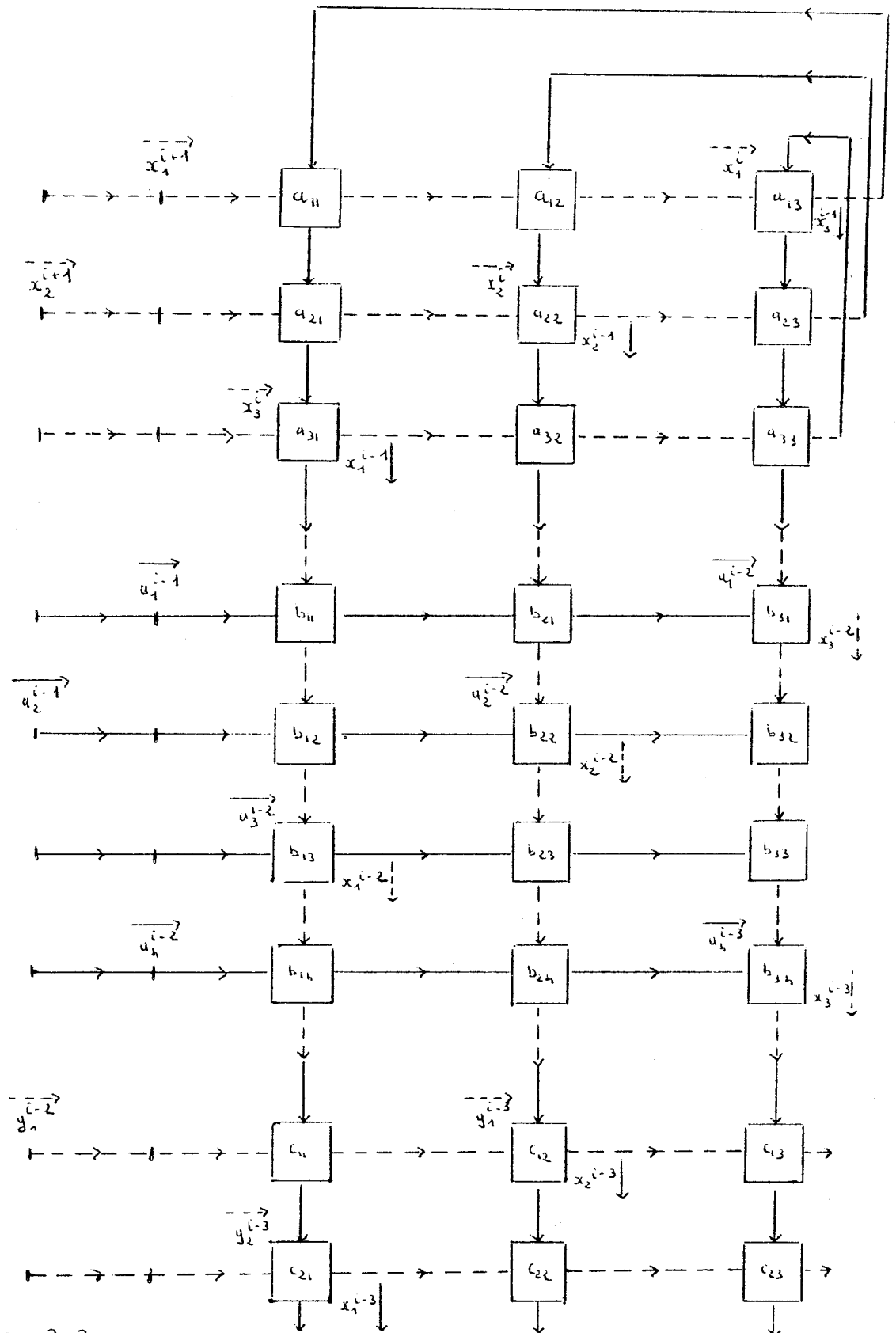
A chaque entrée $u^t = u(t) \in U$ et pour un état $x^t = x(t) \in X$, on associe une sortie $y^t = y(t) \in Y$ liée à u^t et x^t par une fonction de transfert T :



Le système est alors de la forme
$$\begin{cases} x^{t+1} = A x^t + B u^t \\ y^{t+1} = C x^t \end{cases}$$

A matrice $n.n$, B matrice $n.m$, C matrice $p.n$.

Ce problème est modélisé figure 2 pour $(n,m,p) = (3,4,2)$.



A matrice 3x3

B matrice 3x4

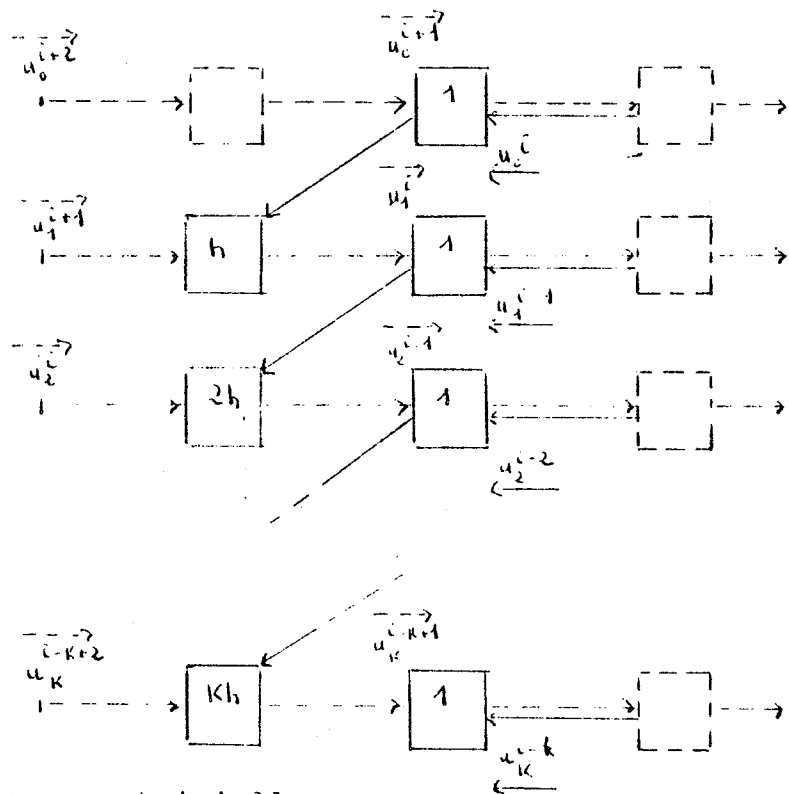
C matrice 2x3

x_i^j : variable-résultat

x_i^j : variable-donnée

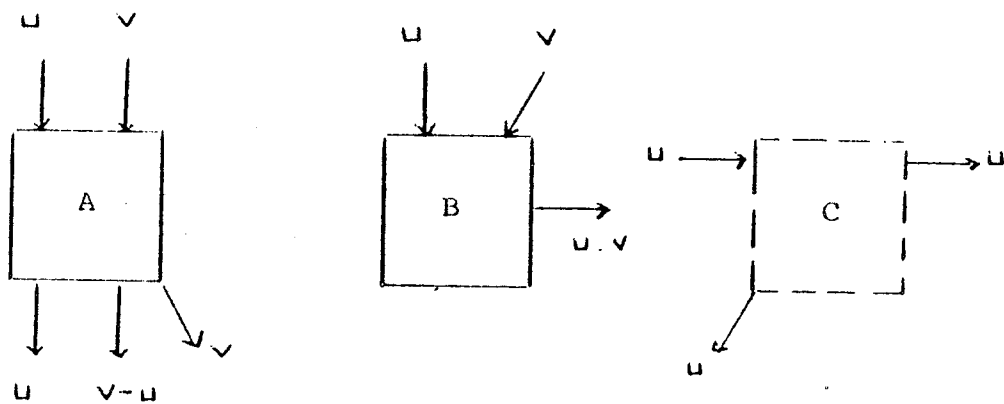
$$\begin{cases} X^{i+1} = A X^i + B U^i \\ Y^{i+1} = C X^{i+1} \end{cases}$$

Figure 2



Réseau de récurrence matricielle

Figure 3a



Cellules du réseau d'initialisation

Figure 3b

Sous-chapitre 2.3 :

Calcul polynomial

Dans la conférence ACM 1981 sur le calcul symbolique et algébrique, H.T.KUNG [19] propose quelques réseaux systoliques pour le calcul polynomial, avec l'objectif d'illustrer l'emploi de la technologie VLSI dans ce domaine du calcul scientifique .

Partant du même ordre d'idées (mais plus modestement) on s'attache dans ce paragraphe à résoudre le problème suivant, abstraitement posé :

soit P un polynôme à coefficients réels de degré K ,
 soit $x_0 \in \mathbb{R}$ une origine et $h > 0$ un pas de calcul donnés
 Trouver un réseau systolique délivrant en temps réel les
 valeurs $(P (x_0 + i h))$, $i = 0, 1, \dots, n, \dots$

A) Développement de Newton

considérons la base de $\mathbb{R}_K[X]$: $U_0=1$, $U_1=X$, $U_2=X(X-h)$,
 $U_j=X(X-h)\dots(X-(j-1)h), \dots$

En posant $P = \sum_{j=0}^K \alpha_j U_j$ et en remarquant que

$$\Delta_h U_j = U_j(X+h) - U_j(X) = j h U_{j-1}(X) , 1 \leq j \leq K$$

on obtient

$$P (x_0 + n h) = \sum_{j=0}^K \alpha_j u_j^n \text{ où } u_j^n = U_j(x_0 + n h)$$

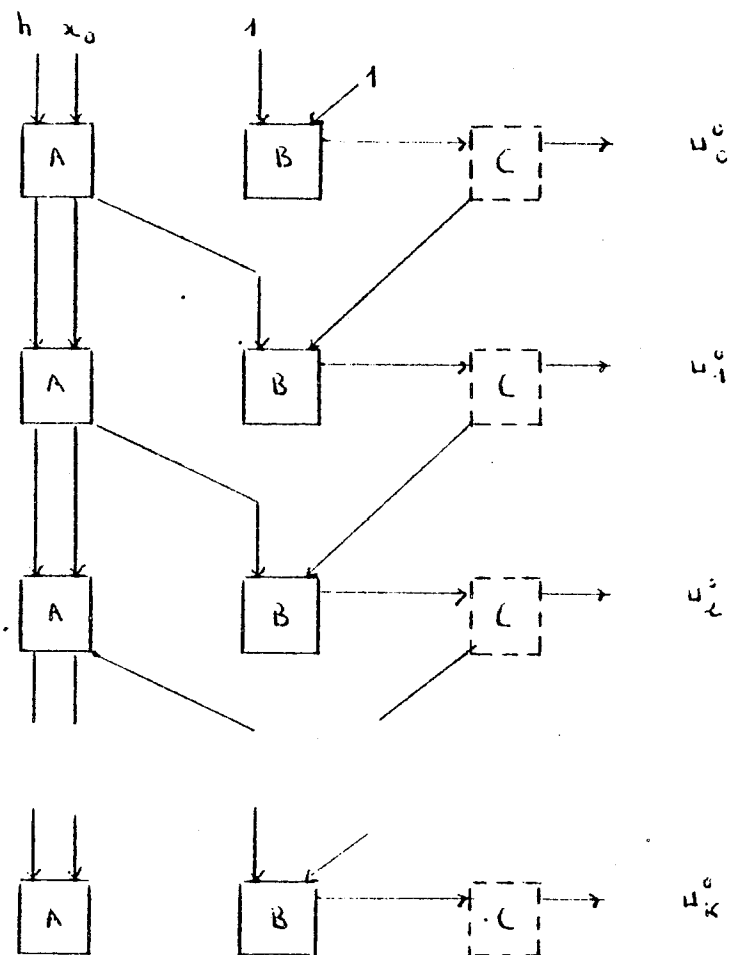
avec la récurrence $u_j^{n+1} = u_j^n + j h u_{j-1}^n$ $1 \leq j \leq K$

qu'on peut écrire sous forme matricielle

$$\begin{bmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_K^{n+1} \end{bmatrix} = \begin{pmatrix} 1 & & & \\ & h & & \\ & & \ddots & \\ & & & Kh & 1 \end{pmatrix} \begin{bmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_K^n \end{bmatrix}$$

B) Réseau proposé

la formulation matricielle précédente conduit directement au réseau de la figure 3a; l'initialisation est décrite aux



Réseau d'initialisation

Figure 3c

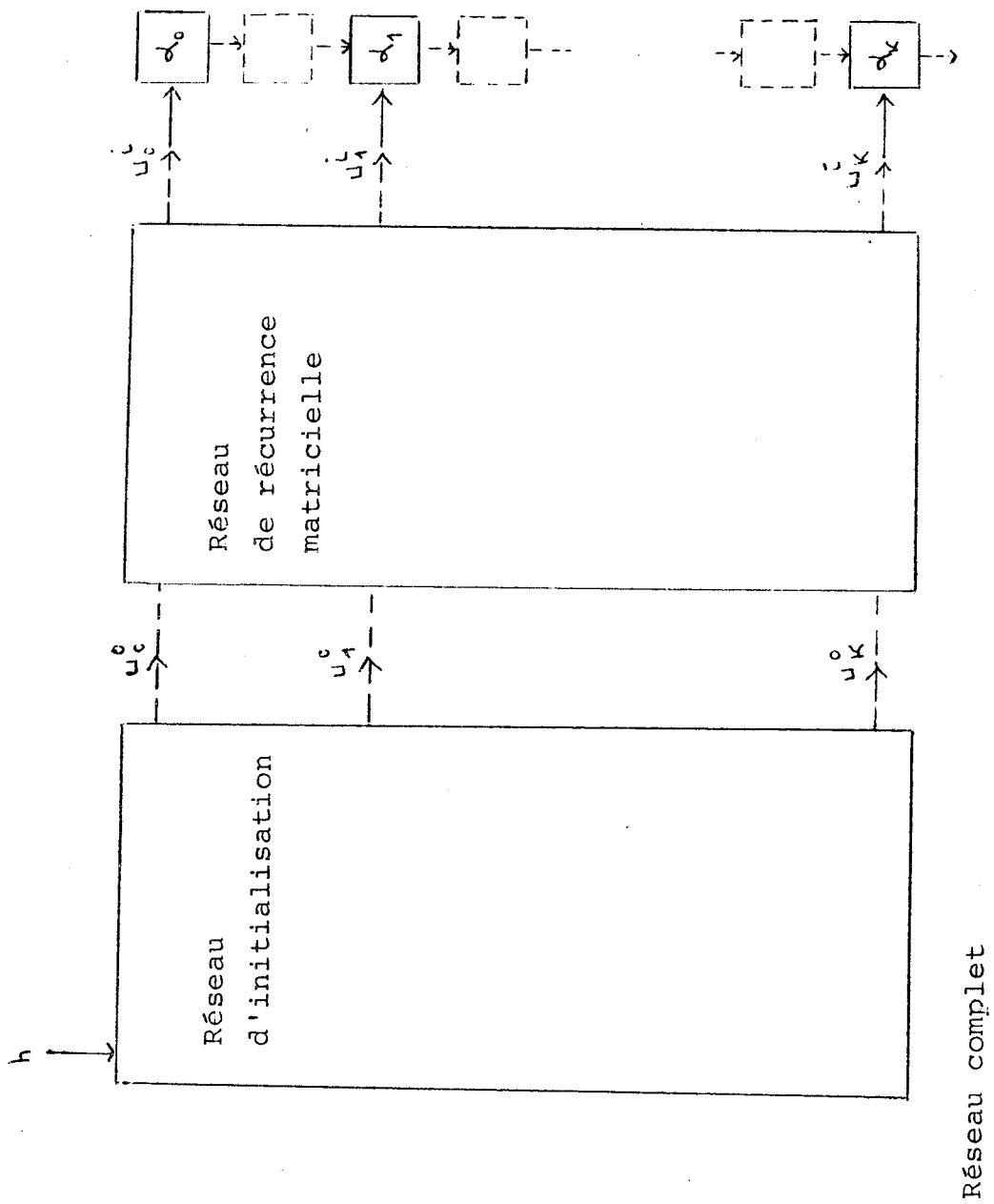


Figure 3d

figures 3b et 3c; le réseau complet est présenté figure 3d, son efficacité est $e = 1/2$.

application 1 : équations différentielles

Le calcul en pipe-line des récurrences de la forme

$$P(n) a_n = Q(n) a_{n-1} + R(n) a_{n-2}$$

où P, Q et $R \in \mathbb{R}[X]$ et a_0, a_1 sont donnés, peut être modélisé à l'aide de l'étude précédente. De telles récurrences sont liées à l'utilisation des séries entières pour la résolution des équations différentielles linéaires

exemple (*) $(t^2+1)y'' + ty' - q^2y = 0, q \in \mathbb{R}_+$

Cherchant s'il existe une série entière $\sum_{n \geq 0} a_n t^n$ de rayon de convergence $R > 0$ telle que $\phi : t \mapsto \sum_{n=0}^{+\infty} a_n t^n$ soit solution de (*) sur $] -R, R[$, on obtient la condition nécessaire

$$\forall n \in \mathbb{N}, (n+2)(n+1)a_{n+2} + (n^2 - q^2)a_n = 0.$$

L'étude des conditions suffisantes n'est pas notre propos (dans tous les cas, $R \geq 1$), mais un réseau permettant de calculer les $(a_n)_{n \geq 2}$ en fonction de a_0 et a_1 est donné figure 3e.

application 2

Soit $f : [x_0, x_0+h] \rightarrow \mathbb{R}$ suffisamment régulière. On cherche à interpoler f par un polynôme de degré K . La modification du pas d'interpolation h s'envisage comme suit :

$$\text{si } p = \sum_{j=0}^K \alpha_j^h u_j^h \quad (\text{avec des notations évidentes})$$

$$\text{alors } \alpha_j^h = \frac{1}{j! h^j} \Delta^j f(x^0) \quad (\Delta^j \text{ est la } j\text{-ème puissance de l'opérateur } \Delta : \Delta f(X) = f(X+h) - f(X))$$

$$\text{Ainsi si } h' = \lambda h, \quad \alpha_j^{h'} = \frac{1}{\lambda^j j!}$$

d'où la modification des cellules du réseau exposé figure 3f.

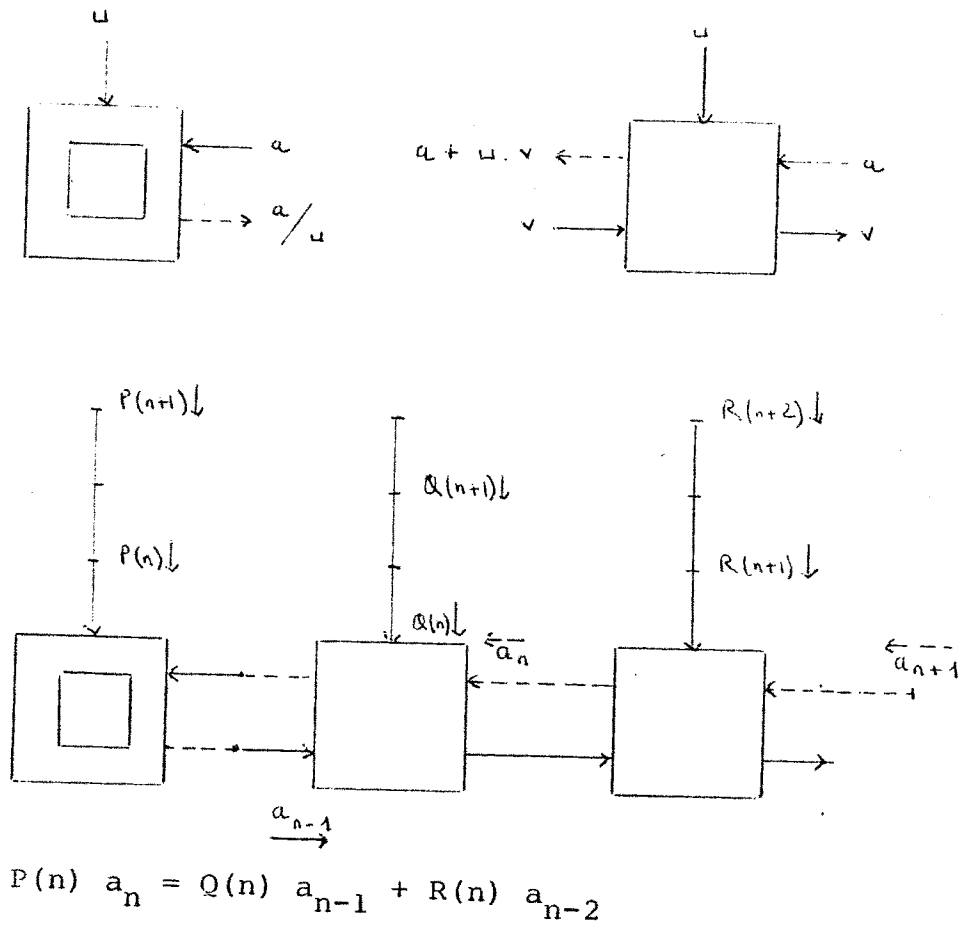
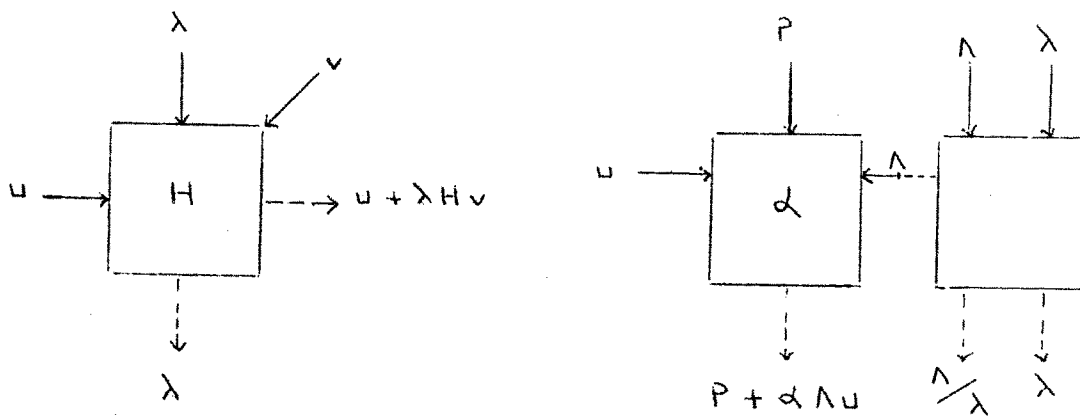


Figure 3e



Réseau de
récence matricielle
Modifications proposées
Figure 3f

Multiplication par
les coefficients α_j

Réseau complet

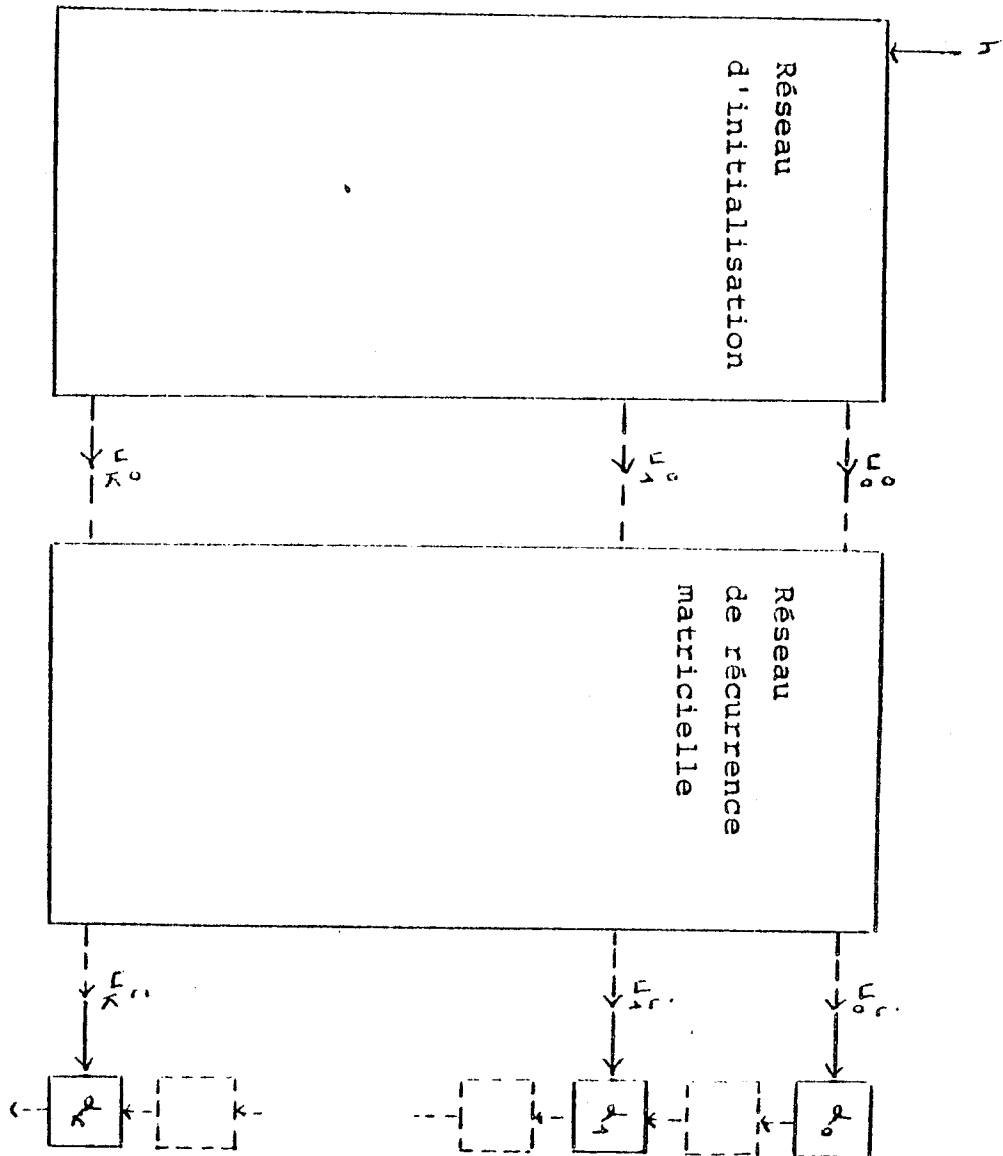


Figure 3d

note un polynôme de degré K qui interpole la fonction f coïncide a priori avec f en $K+1$ valeurs. L'utilisation de P pour approcher f sur le segment $[x_0, x_0+nh]$ ($nh \leq H$) peut cependant s'envisager compte tenu de la formule d'erreur :

$$|(f-P)(x)| \leq \frac{|\omega(x)|}{(K+1)!} \|f^{(K+1)}\|_{\infty}$$

qui utilise les notations habituelles [14], qui suppose f de classe C^{K+1} , et qui se démontre à l'aide du théorème de Rolle généralisé.

application 3

On utilise les résultats précédents pour calculer les valeurs en

$(x_0 + i h, y_0 + j h)$, $1 \leq j \leq N$, $i \geq 0$
d'un polynôme $P \in \mathbb{R}[X]$ de degrés partiels H et K en X et Y respectivement. Pour simplifier, on expose la méthode de calcul de

$$(P(n, m)), \quad 1 \leq m \leq N, \quad n \geq 0$$

Ecrivaint

$$P(X, Y) = \sum_{i=1}^H \sum_{j=1}^K a_{ij} U_i(X) U_j(Y)$$

et notant $u_n^i = U_i(n)$, il vient $u_n^i = u_{n-1}^i + i u_{n-1}^{i-1}$ ($n \geq 1$)

Si $P_j(n) = \sum_{i=1}^H a_{ij} u_n^i$, alors $P(n, m) = \sum_{j=1}^K P_j(n) u_j(m)$,
écriture qui conduit au réseau de la figure 3g.

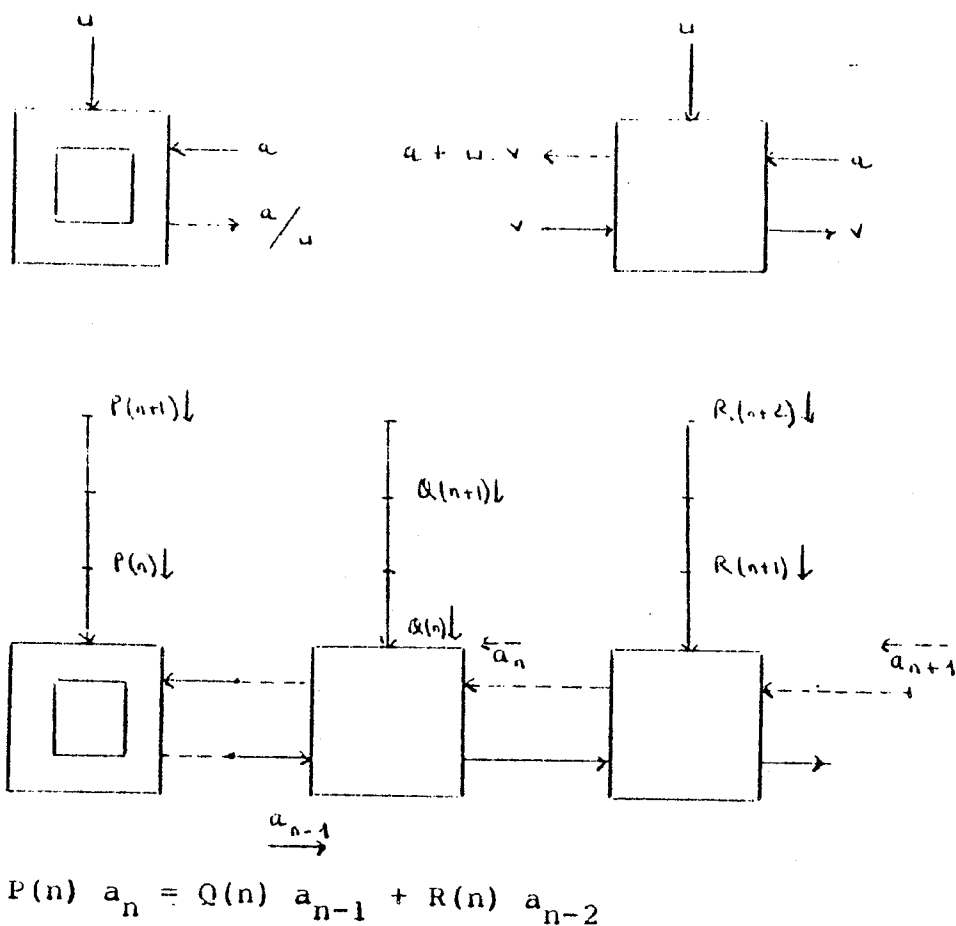
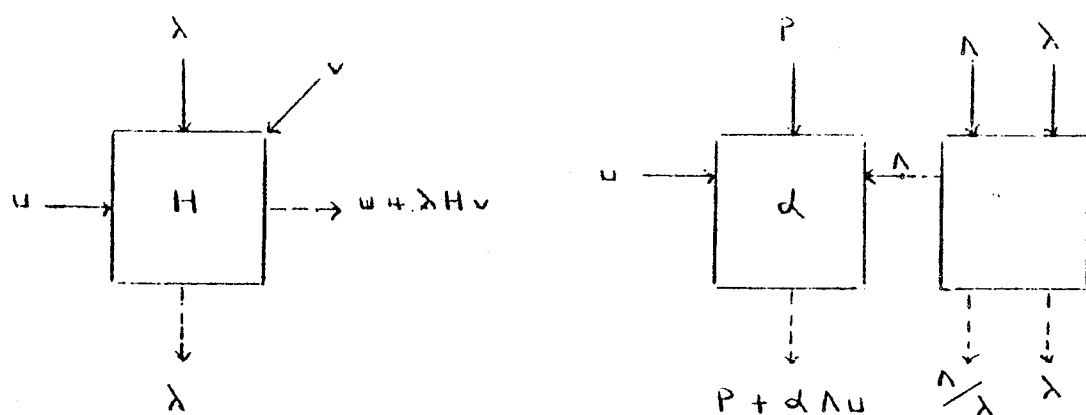


Figure 3e



Réseau de
récence matricielle
Modifications proposées
Figure 3f

Multiplication par
les coefficients a_j

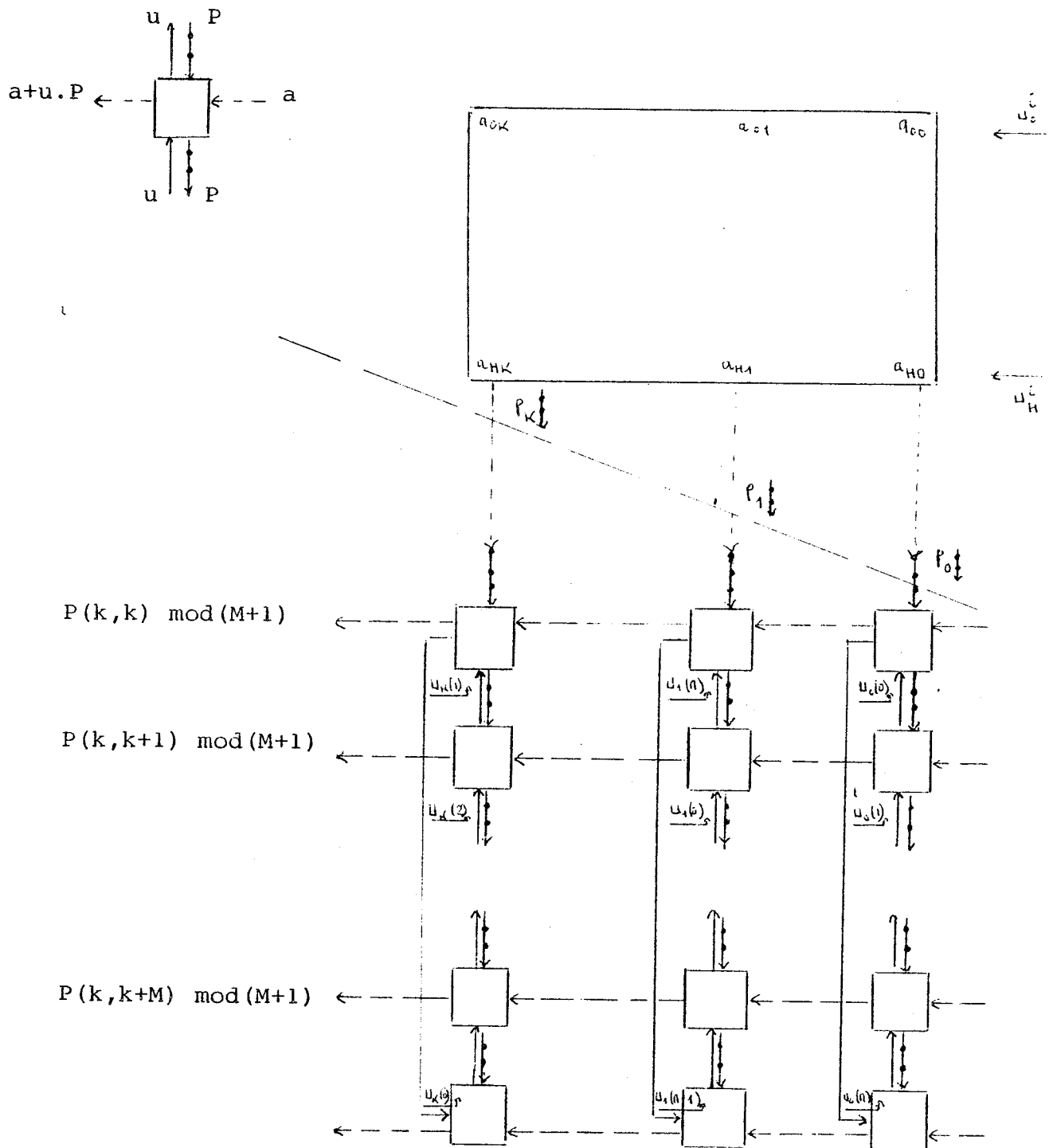


Schéma du réseau pour le calcul à deux variables
Figure 3g

Sous-chapitre 2.4 :

Convolution bidimensionnelle

La convolution bidimensionnelle (ou 2D-convolution) est une technique de filtrage largement utilisée en traitement du signal (algorithme de lissage d'une image).

Le problème peut s'énoncer ainsi : étant données n lignes infinies de données x_j^i , $1 \leq j \leq n$, et une fenêtre de taille $k.k$ avec des coefficients de pondération w_{ij} , $1 \leq i, j \leq k$, calculer

$$\sum_{i=1}^k \sum_{j=1}^k w_{ij} x_{r+i-2}^{s+j-2}, \quad 1 \leq r \leq n \text{ et } s \geq 0$$

Ce problème a été résolu par H.T.KUNG et S.W.SONG[22]. On donne dans l'annexe 2 une solution plus simple, utilisant uniquement les cellules de base de type 1, sous la forme d'un réseau régulier d'efficacité $e = 1/2$.

Par ailleurs, on explicite dans cette annexe quelques cas particuliers intéressants .

CHAPITRE 3

TITRE : EXEMPLES DE RESEAUX UTILISANT DES CELLULES
A MODE DE CONNEXION HEXAGONAL

RESUME : Deux algorithmes non linéaires sont modélisés dans ce chapitre :

- à partir des réseaux proposés en [21] pour la décomposition $A = L.U$ d'une matrice bande A , on étudie la factorisation incomplète de A (méthode itérative pour résoudre le système $A x = b$, décrite en annexe 3)
- après la convolution (chapitre précédent), on étudie une nouvelle technique de lissage, appelée algorithme de la médiane, modélisée dans le cas unidimensionnel cette fois (à cause de la complexité de l'algorithme, due à sa non-linéarité) .

Sous-chapitre 3.1 :

Factorisation incomplète de matrices-bandes

A) Factorisation L.U

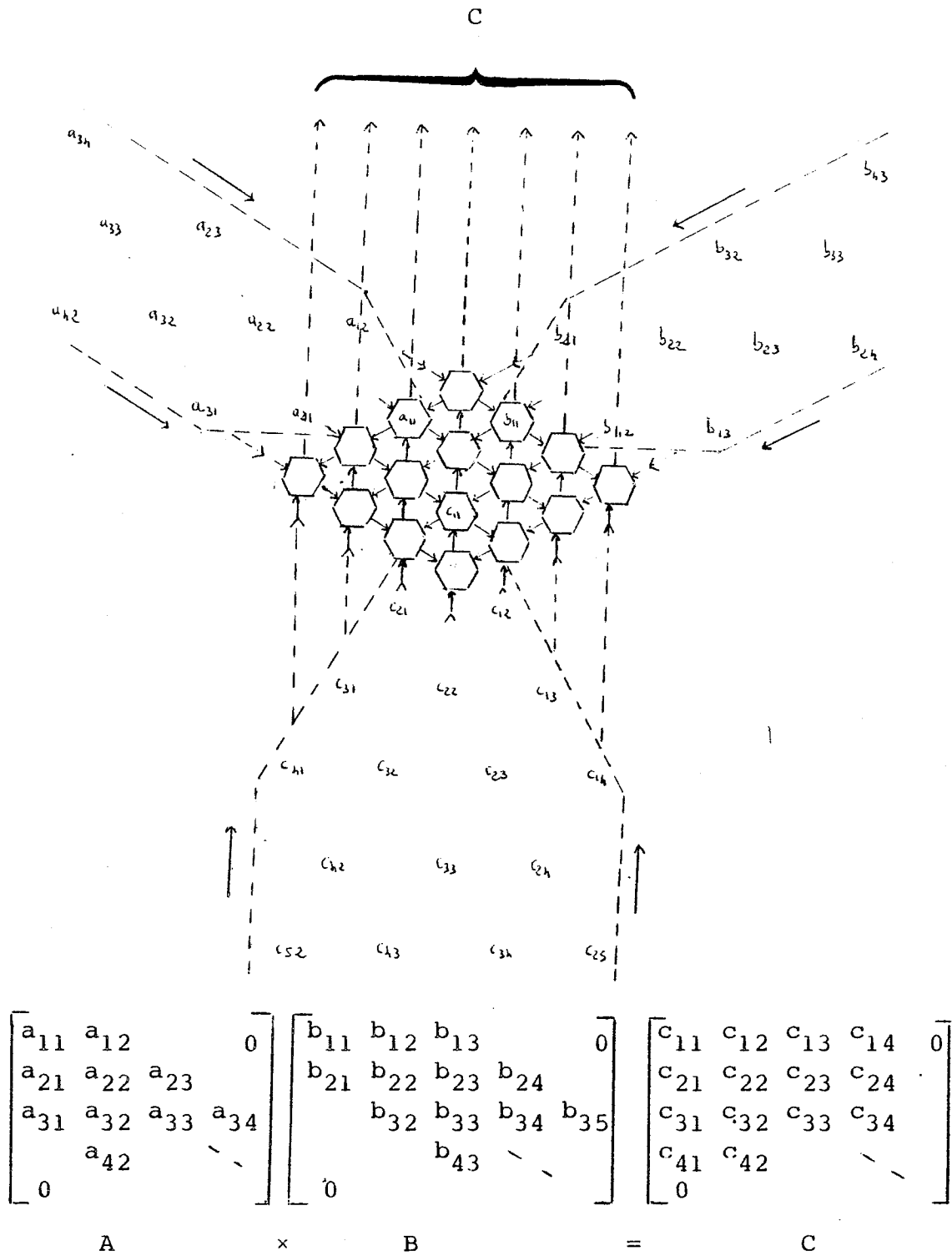
Les réseaux des figures 1 et 2 sont dus à H.T.KUNG et C.E.LEISERSON [21]; le réseau de la figure 1 modélise le produit $C = C_1.C_2$ de deux matrices creuses à structure bande, celui de la figure 2 la décomposition $A = L.U$ d'une matrice-bande .

Le lien entre ces deux réseaux peut s'expliquer ainsi: pour connaître les coefficients de L et U, il suffit de déterminer $A' = L + U$, en notant $L' = L - I$, puisque L' et U sont disjointes. Or $A' = A + L'.(I - U)$, les premiers coefficients sont calculés directement à partir de A puis réinjectés dans un réseau comme celui de la figure 1 ($C_1 = L'$, $C_2 = I - U$, A initialise le réseau qui calcule $C = A + C_1.C_2$) .

B) Factorisation incomplète

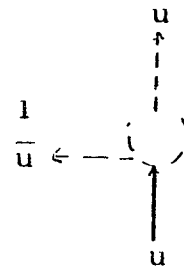
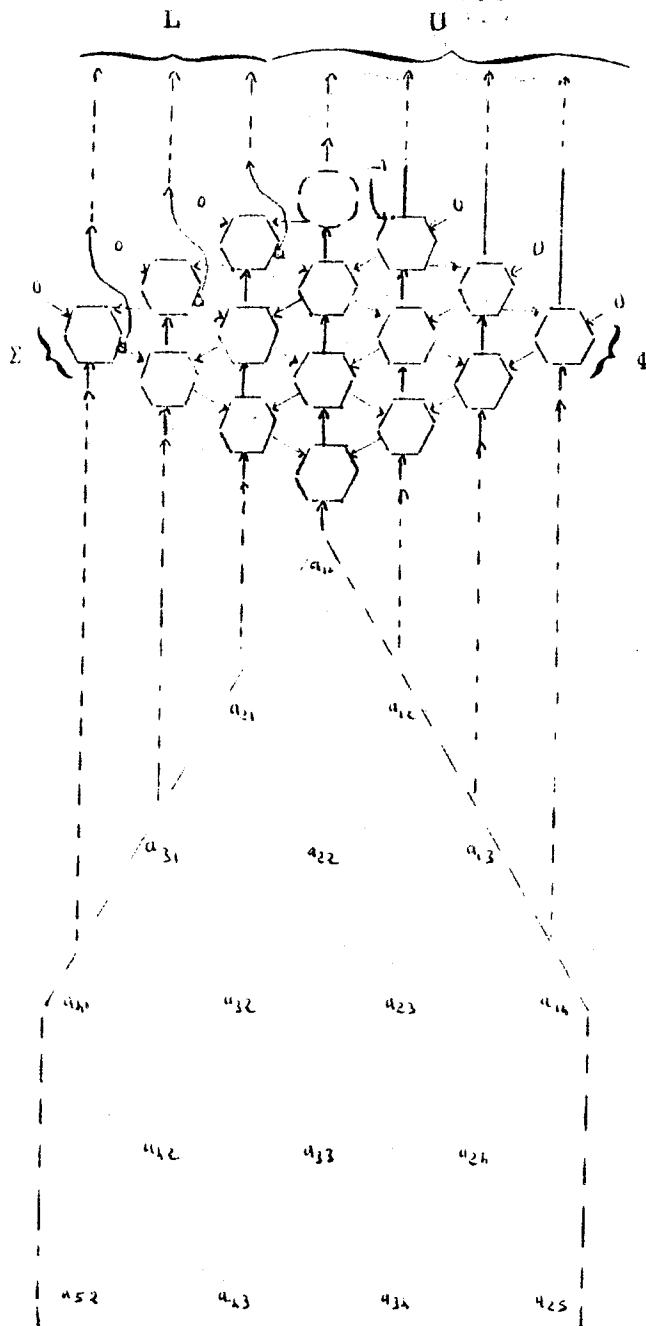
Considérons une matrice A (par exemple issue de la discrétisation d'un problème d'équations aux dérivées partielles) ayant la forme symétrique suivante :

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



Multiplication de matrices-bandes

Figure 1



Les cellules de la
rangée Σ
(resp: de la rangée Ψ)
ont subi une rotation
de 120° dans le sens
(resp: sens inverse)
des aiguilles d'une montre

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & & \\ & a_{52} & a_{53} & & \\ 0 & & & & \end{bmatrix}$$

$= A = L \times U$, L triangulaire inférieure à
diagonale unité et
U triangulaire supérieure

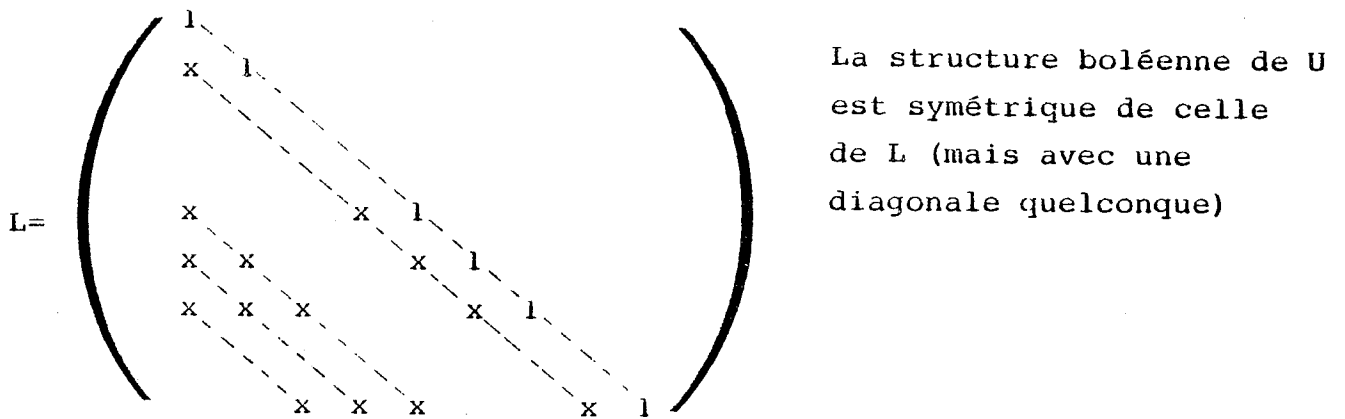
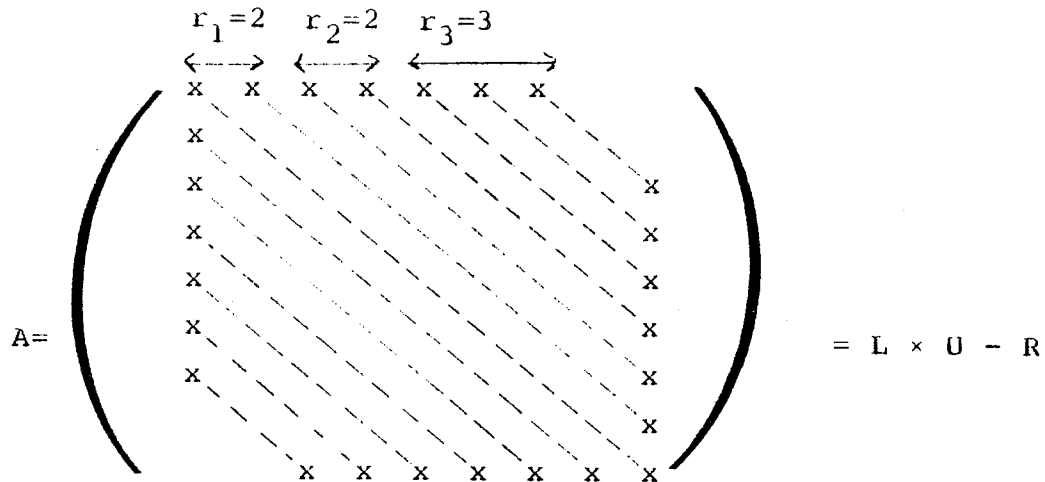
Factorisation L.U d'une matrice-bande
Figure 2

La décomposition L.U de A utilise un nombre de cellules directement lié à la largeur $p + r_1 + r_2$ de la bande de A. Pour tirer parti de la structure creuse de A (et éliminer les processeurs inutiles), on peut envisager une factorisation incomplète [31] : cette méthode, présentée en détail dans l'annexe 3, est d'autant plus avantageuse que p est grand (exemple de la matrice du potentiel). On obtient alors une décomposition $A = L.U - R$, où L est triangulaire inférieure de diagonale unité et U triangulaire supérieure, avec

$$l_{ij} = u_{ij} = 0 \text{ si } r_1 < |i-j| < r_1 + r_2 + p$$

$$r_{ij} = 0 \text{ si } 0 < |i-j| \leq r_1 \text{ ou } |i-j| \geq r_1 + r_2 + p$$

Un réseau à mode de connexion hexagonal est proposé figure 3, sur un exemple, pour le calcul des facteurs L et U de cette décomposition (l'algorithme itératif lié à la factorisation incomplète ne nécessite pas la connaissance de R). Le nombre de cellules ne dépend que de r_1 et r_2 , et non de p. D'autre part, la figure 4 offre la possibilité d'une comparaison visuelle entre les deux méthodes .



Factorisation incomplète : principe

Figure 3a

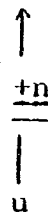
La variable u est retardée de n pulsations



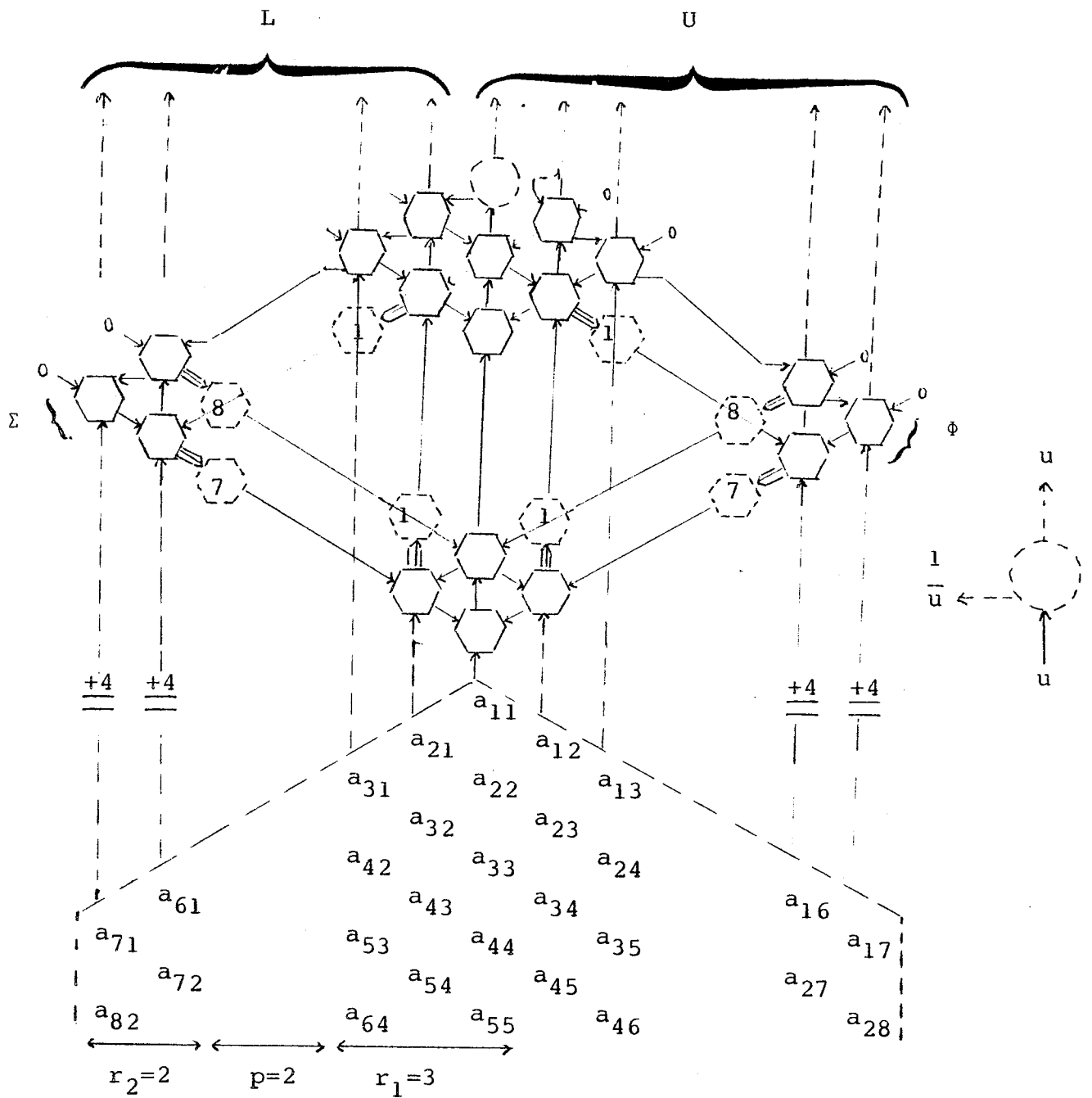
en traversant cette cellule

Voir figure 3b

La donnée u



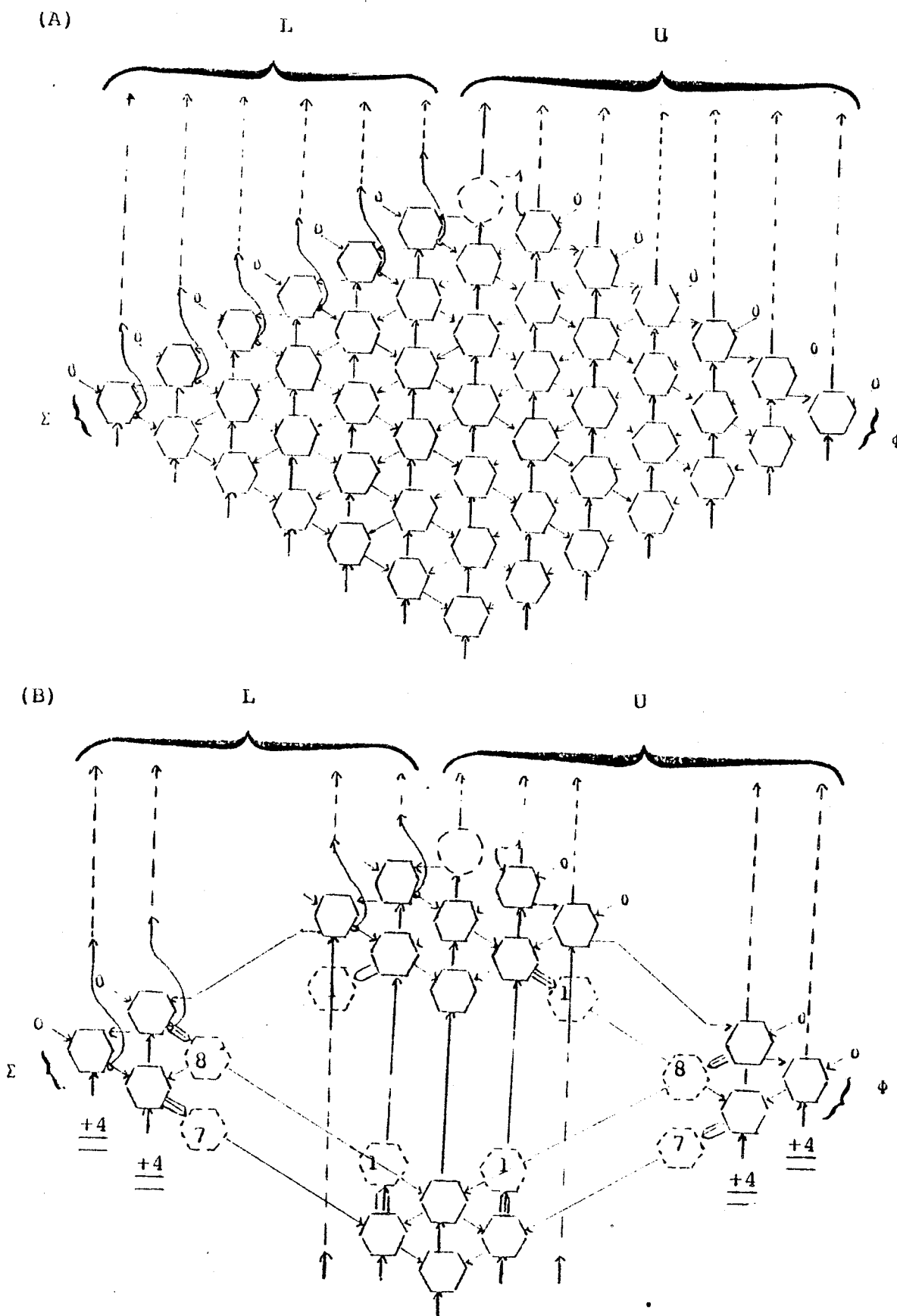
entre en fait n pulsations plus tôt dans le réseau



Les cellules de la rangée Σ (resp: Φ) ont subi une rotation de 120° dans le sens (resp: sens inverse) des aiguilles d'une montre

Factorisation incomplète d'une matrice-bande

Figure 3b



(A) Décomposition L.U : 49 cellules actives
(B) Factorisation incomplète : 19 cellules actives,
8 processeurs de retard

Figure 4 : Comparaison des deux réseaux

Sous-chapitre 3.2 :

Algorithme de la médiane

Le lissage par médiane est une technique de filtrage - utilisée en traitement du signal - qui consiste à remplacer chaque point de l'image par la médiane à l'intérieur d'un certain voisinage local. Contrairement à la convolution du sous-chapitre 2.4, le lissage par médiane a l'avantage de ne pas introduire de nouvelles valeurs dans l'échantillon des données...malheureusement, la non-linéarité de l'algorithme conduit à des structures compliquées, comme celle de A.L.FISHER[09] .

Nous proposons dans l'annexe 2 des réseaux n'utilisant que des cellules définies au chapitre 1 pour les cas particuliers où la médiane est prise sur un voisinage de 3 ou 5 valeurs (application immédiate au lissage d'un tableau à deux dimensions sur le voisinage de VON NEUMANN

$$\begin{array}{ccccc} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{array})$$

La médiane de $2k-1$ valeurs $a_1, \dots, a_{2k-1}$ s'exprime par la formule

$$i_1 \quad \dots \quad i_k \quad (a_{i_1} \wedge \dots \wedge a_{i_k})$$

Les cellules de base sont définies à l'aide des opérateurs $\wedge$ (minimum) et $\vee$ (maximum), les réseaux sont présentés figures 5 et 6 (annexe 2) .

CHAPITRE 4

TITRE : SIMULATION SEQUENTIELLE
D'UN ALGORITHME SYSTOLIQUE

RESUME : On simule séquentiellement sur micro-ordinateur (en PASCAL) l' exécution de l'algorithme de la médiane à 3 et 5 éléments. Cette simulation teste la validité des réseaux proposés au chapitre précédent .

Sous-chapitre 4.1 :

Contexte de la simulation

Dans un algorithme comme celui de la médiane (à 3 ou 5 éléments), chaque cellule est active une fois toutes les deux pulsations; distinguant les cellules paires (actives aux pulsations d'ordre pair) des cellules impaires, on peut simuler l'exécution - par nature en parallèle - de l'algorithme systolique d'une manière séquentielle comme suit: on fait jouer à un compteur le rôle d'une horloge, et, à chaque incrémentation (chaque pas dans une boucle FOR...DO en PASCAL), on active par une boucle interne

- d'abord les cellules impaires
- ensuite, les cellules paires .

C'est avec cette idée que nous avons écrit un programme en PASCAL simulant l'exécution de l'algorithme de la médiane sur un réseau systolique...pour obtenir un programme de plus de 200 lignes, alors que la transcription directe de l'algorithme n'aurait pas nécessité plus de 50 lignes ! L'intérêt de la simulation réside bien sûr dans la preuve de validation du réseau qu'elle apporte: le bon fonctionnement de ce réseau est ainsi établi autrement que par une page d'écriture formelle suivie d'un dessin comme dans l'annexe 2 .

Les outils de vérification du bon tracé des réseaux dessinés sont fort peu nombreux, malgré la tentative [08] destinée plutôt à fournir un moyen de trouver un réseau-solution qu'à le vérifier. il peut alors sembler intéressant de noter que la simulation écrite en PASCAL sur micro-ordinateur MICRAL 90 a conduit à un programme n'occupant que 28 K en place mémoire sur les 100 K disponibles par disquette, et ce pour un réseau comportant 25 cellules actives. On peut donc envisager de simuler de cette façon le fonctionnement de réseaux systoliques comportant jusqu'à 100 cellules actives .

Sous-chapitre 4.2 :

Listings

Le programme est écrit en PASCAL MT+86; les procédures sont précédées par des commentaires expliquant leur objet .

L'utilisateur rentre 30 entiers, puis choisit l'un des nombres 3 ou 5 selon l'algorithme désiré. On donne un exemple de résultats pour chacun de ces choix :

- pour la médiane à 3, six séquences suffisent à vérifier la validité
- pour la médiane à 5, il en faudrait 120, aussi ne donne-t-on dans ce cas qu'un seul exemple d'illustration .

```

1: PROGRAM MEDIANE (INPUT,OUTPUT) ;
2:
3: (* CE PROGRAMME SIMULE SEQUENTIELLEMENT L'EXECUTION EN PARALLELE DE
4: L'ALGORITHME DU LISSAGE PAR MEDIANE D'UNE SUITE UNIDIMENSIONNELLE DE
5: DONNEES .
6: CET ALGORITHME EST DEFINI DANS LE RR 295 , IMAG MARS 1982 .
7:
8: CE PROGRAMME DEMANDE A L'UTILISATEUR DE RENTRER UN TABLEAU ENTIER
9: A N COMPOSANTES , PUIS DE DECIDER SI LA MEDIANE DOIT ETRE PRISE SUR
10: UN VOISINAGE DE 3 OU 5 ELEMENTS .
11: LE PROGRAMME AFFICHE ALORS LE RESULTAT CORRESPONDANT . *)
12:
13:
14: CONST N = 30 ;
15:     N1 = 29 ;
16:     N2 = 28 ;
17:     N3 = 27 ;
18:     N4 = 26 ;
19: TYPE TABO = ARRAY [1..N] OF INTEGER ;
20:     TAB1 = ARRAY [1..N1] OF INTEGER ;
21:     TAB2 = ARRAY [1..N2] OF INTEGER ;
22:     TAB3 = ARRAY [1..N3] OF INTEGER ;
23:     TAB4 = ARRAY [1..N4] OF INTEGER ;
24: VAR A : TABO ;
25:     MED3 : TAB2 ;
26:     MED5 : TAB4 ;
27:     CHOIX : INTEGER ;
28:
29: (* LES FONCTIONS MIN ET MAX DEFINIES CI-DESSOUS CORRESPONDENT AUX
30: OPERATIONS ELEMENTAIRES EFFECTUEES EN PARALLELE PAR LES CELLULES
31: DU RESEAU SYSTOLIQUE *)
32:
33: FUNCTION MIN (X1,Y1 : INTEGER) : INTEGER ;
34:     BEGIN
35:         IF (X1 < Y1) THEN MIN := X1
36:             ELSE MIN := Y1
37:     END ;
38:
39: FUNCTION MAX (X2,Y2 : INTEGER) : INTEGER ;
40:     BEGIN
41:         IF (X2 < Y2) THEN MAX := Y2
42:             ELSE MAX := X2
43:     END ;
44:
45: (* LA PROCEDURE CI-DESSOUS DETERMINE LE MODE D'ENTREE DES DONNEES :
46: UNE VALEUR DE LA SUITE PAR LIGNE *)
47:
48: PROCEDURE LIRETABL (VAR X : TABO) ;
49:     VAR J : INTEGER ;
50:     BEGIN
51:         WRITELN ('** ENTRER LE TABLEAU DES DONNEES (30 ENTIERES) **') ;
52:         FOR J := 1 TO N DO
53:             BEGIN
54:                 READ (X[J]) ;
55:                 READLN
56:             END
57:         END ;
58:
59: (* LES DEUX PROCEDURES CI-DESSOUS DETERMINENT LE MODE D'IMPRESSION
60: DES RESULTATS , SELON QUE LA MEDIANE A ETE PRISE SUR UN VOISINAGE

```

```

61: DE 3 OU DE 5 VALEURS *)
62:
63: PROCEDURE INFR3 (Y11 : TABO ; Y11 : TAB2) ;
64:     VAR K : INTEGER ;
65:     BEGIN
66:         WRITELN ('*** RESULTAT ***') ;
67:         WRITELN ;
68:         FOR K := 1 TO N-2 DO
69:             BEGIN
70:                 WRITE ('* LA MEDIANE A 3 DE ',Y11K3,' ',Y11K+11,' ',
71:                     Y11K+23,' EST ',Y11K3,' *') ;
72:                 WRITELN ;
73:                 WRITELN
74:             END
75:         END ;
76:
77: PROCEDURE INFR5 (Y12 : TABO ; Y12 : TAB4) ;
78:     VAR L : INTEGER ;
79:     BEGIN
80:         WRITELN ('*** RESULTAT ***') ;
81:         WRITELN ;
82:         FOR L := 1 TO N-4 DO
83:             BEGIN
84:                 WRITE ('* LA MEDIANE A 5 DE ',Y12L3,' ',Y12L+13,' ',
85:                     Y12L+23,' ',Y12L+33,' ',Y12L+43,' EST ',Y12L3,' *')
86:                 WRITELN ;
87:                 WRITELN
88:             END
89:         END ;
90:
91: (* LES DEUX PROCEDURES QUI SUIVENT CONSTITUENT LE COEUR DU PROGRAMME :
92: ELLES SIMULENT L'EXECUTION EN PARALLELE DE L'ALGORITHME DE LA MEDIANE,
93: POUR UN VOISINAGE A 3 ELEMENTS (PROCEDURE MEDAS) OU A 5 ELEMENTS (PRO-
94: CEDURE MEDAS).
95:
96: LEUR STRUCTURE EST IDENTIQUE :
97:     -A CHAQUE VARIABLE DEFINIE CORRESPOND EN FAIT UNE CELLULE DU
98: RESEAU SYSTOLIQUE
99:     -CHAQUE CELLULE ETANT ACTIVE TOUTES LES DEUX PULSATIONS ,
100: ON DISTINGUE LES CELLULES IMPAIRES (ACTIVES AUX PULSATIONS IMPAIRES)
101: ET LES CELLULES PAIRES .
102:     LA PROCEDURE SIMULE D'ABORD L'EXECUTION SYNCHRONE DES OPERATIONS
103: ELEMENTAIRES EFFECTUEES PAR LES CELLULES IMPAIRES .
104:     -ENFIN , LES DIFFERENTS TESTS EFFECTUES INTERVIENNENT POUR L'INI-
105: TIALISATION DU RESEAU AINSI QUE POUR LE CALCUL DES DERNIERES VALEURS *)
106:
107: PROCEDURE MEDAS (Y1 : TABO ; VAR YHED3 : TAB2) ;
108:     VAR U : TAB1 ;
109:     V,B : TAB2 ;
110:     N : INTEGER ;
111:     BEGIN
112:         FOR N := 1 TO N-1 DO
113:             BEGIN
114:                 UEN3 := MIN (Y1EN3,Y1EN+13) ;
115:                 IF (N)1 THEN BEN+13 := MAX (UEN+13,VEN+13) ;
116:                 IF (N)N-1 THEN VEN3 := MIN (Y1EN3,Y1EN+23) ;
117:                 IF (N)1 THEN YHED3EN+13 := MAX (UEN3,BEN+13)
118:             END
119:         END ;
120:

```

```

121: PROCEDURE HELDAS (YA1 : TAB0 ; VAR YHED5 : TAB4) ;
122:   VAR U1 : TAB1 ;
123:   U2,UR1,V1 : TAB2 ;
124:   U3,UR2,V2,B1,VR1,W1 : TAB3 ;
125:   U4,UR3,V3,B2,M1,VR2,BR1,W2,C1,WR1,MR2,M2,X1,XR1,M3 : TAB4 ;
126:   I : INTEGER ;
127: BEGIN
128:   FOR I := 1 TO N-1 DO
129:     BEGIN
130:       U1(I) := MIN (YA1(I),YA1(I+1)) ;
131:       IF (I>1) THEN
132:         BEGIN
133:           IF (I<N-1) THEN
134:             BEGIN
135:               UR2(I-1) := MIN (YA1(I),YA1(I+2)) ;
136:               U3(I-1) := MIN (YA1(I-1),YA1(I+2)) ;
137:             END ;
138:             V1(I-1) := MIN (U1(I-1),U2(I-1)) ;
139:           END ;
140:           IF (I>2) THEN
141:             BEGIN
142:               W1(I-2) := MIN (U2(I-2),U3(I-2)) ;
143:               IF (I<N-1) THEN
144:                 BEGIN
145:                   V3(I-2) := MIN (U1(I-2),U4(I-2)) ;
146:                   B2(I-2) := MAX (B1(I-2),V3(I-2)) ;
147:                   VR2(I-2) := MIN (UR1(I-2),UR3(I-2)) ;
148:                   BR1(I-2) := MAX (VR1(I-2),VR2(I-2)) ;
149:                   C1(I-2) := MAX (W1(I-2),W2(I-2)) ;
150:                 END ;
151:               END ;
152:             IF (I>3) THEN
153:               BEGIN
154:                 XR1(I-3) := MIN (UR1(I-2),UR1(I-1)) ;
155:                 MR2(I-3) := MAX (M1(I-3),WR1(I-3)) ;
156:                 M2(I-3) := MAX (MR2(I-3),C1(I-3)) ;
157:                 X1(I-3) := MIN (U3(I-3),U4(I-3)) ;
158:               END ;
159:             IF (I<N-1) THEN
160:               BEGIN
161:                 U2(I) := MIN (YA1(I),YA1(I+2)) ;
162:                 UR1(I) := MIN (YA1(I+1),YA1(I+2)) ;
163:               END ;
164:             IF (I>1) THEN
165:               BEGIN
166:                 IF (I<N-1) THEN
167:                   BEGIN
168:                     V2(I-1) := MIN (U1(I-1),U3(I-1)) ;
169:                     B1(I-1) := MAX (V1(I-1),V2(I-1)) ;
170:                     VR1(I-1) := MIN (UR1(I-1),UR2(I-1)) ;
171:                   END ;
172:                   IF (I<N-2) THEN
173:                     BEGIN
174:                       U4(I-1) := MIN (YA1(I-1),YA1(I+3)) ;
175:                       UR3(I-1) := MIN (YA1(I),YA1(I+3)) ;
176:                     END ;
177:                   END ;
178:             IF ( (I>2) AND (I<N-1) ) THEN
179:               BEGIN
180:                 M1(I-2) := MAX (B2(I-2),BR1(I-2)) ;

```

```

181:          WR1EI-2J := MIN (UR2EI-2J,UR3EI-2J) ;
182:          W2EI-2J := MIN (U2EI-2J,U4EI-2J) ;
183:          C1EI-2J := MAX (W1EI-2J,W2EI-2J)
184:          END ;
185:          IF (I>3) THEN
186:              BEGIN
187:                  H3EI-3J := MAX (X1EI-3J,XR1EI-3J) ;
188:                  YHED5EI-3J := MAX (H2EI-3J,H3EI-3J)
189:              END
190:          END
191:      END ;
192:
193:
194:
195:
196: (* VOICI MAINTENANT LE PROGRAMME PRINCIPAL : TRES COURT , IL APPELLE
197: LES PROCEDURES DEFINIES PLUS HAUT *)
198:
199: BEGIN
200: LIRETABL (A) ;
201: WRITE ('ENTRER LE NOMBRE 3 OU 5 SUIVANT L'ALGORITHME DESIRE') ;
202: WRITELN ;
203: READLN (CHOIX) ;
204: IF ( CHOIX = 3 ) THEN
205:     BEGIN
206:         MEDA3 (A,MED3) ;
207:         IMPR3 (A,MED3)
208:     END ;
209: IF ( CHOIX = 5 ) THEN
210:     BEGIN
211:         MEDA5 (A,MED5) ;
212:         IMPR5 (A,MED5)
213:     END ;
214: END .

```

Un exemple d'utilisation de la médiane à trois

ENTRER LE NOMBRE 3 OU 5 SUIVANT L'ALGORITHME DESIRE

3

*** RESULTAT ***

* LA MEDIANE A 3 DE 1 2 3 EST 2 *

* LA MEDIANE A 3 DE 2 3 1 EST 2 *

* LA MEDIANE A 3 DE 3 1 3 EST 3 *

* LA MEDIANE A 3 DE 1 3 2 EST 2 *

* LA MEDIANE A 3 DE 3 2 2 EST 2 *

* LA MEDIANE A 3 DE 2 2 1 EST 2 *

* LA MEDIANE A 3 DE 2 1 3 EST 2 *

* LA MEDIANE A 3 DE 1 3 2 EST 2 *

* LA MEDIANE A 3 DE 3 2 3 EST 3 *

* LA MEDIANE A 3 DE 2 3 1 EST 2 *

* LA MEDIANE A 3 DE 3 1 3 EST 3 *

* LA MEDIANE A 3 DE 1 3 1 EST 1 *

* LA MEDIANE A 3 DE 3 1 2 EST 2 *

* LA MEDIANE A 3 DE 1 2 3 EST 2 *

* LA MEDIANE A 3 DE 2 3 2 EST 2 *

10 LA MEDIANE A 3 DE 3 2 1 EST 2 *

* LA MEDIANE A 3 DE 2 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

LA MEDIANE A 3 DE 1 1 1 EST 1

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

LA MEDIANE A 3 DE 1 1 1 EST 1 x

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 *

* LA MEDIANE A 3 DE 1 1 1 EST 1 x

Un exemple d'utilisation de la médiane à cinq

ENTRER LE NOMBRE 3 OU 5 SUIVANT L'ALGORITHME DESIRE
5

*** RESULTAT ***

* LA MEDIANE A 5 DE 3 4 2 7 5 EST 4 *
* LA MEDIANE A 5 DE 4 2 7 5 4 EST 4 *
* LA MEDIANE A 5 DE 2 7 5 4 2 EST 4 *
* LA MEDIANE A 5 DE 7 5 4 2 8 EST 5 *
* LA MEDIANE A 5 DE 5 4 2 8 7 EST 5 *
* LA MEDIANE A 5 DE 4 2 8 7 5 EST 5 *
* LA MEDIANE A 5 DE 2 8 7 5 3 EST 5 *
* LA MEDIANE A 5 DE 8 7 5 3 9 EST 7 *
* LA MEDIANE A 5 DE 7 5 3 9 8 EST 7 *
* LA MEDIANE A 5 DE 5 3 9 8 4 EST 5 *
* LA MEDIANE A 5 DE 3 9 8 4 2 EST 4 *
* LA MEDIANE A 5 DE 9 8 4 2 1 EST 4 *
* LA MEDIANE A 5 DE 8 4 2 1 3 EST 3 *
* LA MEDIANE A 5 DE 4 2 1 3 5 EST 3 *
* LA MEDIANE A 5 DE 2 1 3 5 6 EST 3 *
* LA MEDIANE A 5 DE 1 3 5 6 3 EST 3 *
* LA MEDIANE A 5 DE 3 5 6 3 7 EST 5 *
* LA MEDIANE A 5 DE 5 6 3 7 5 EST 5 *
* LA MEDIANE A 5 DE 6 3 7 5 8 EST 6 *
* LA MEDIANE A 5 DE 3 7 5 8 6 EST 6 *
* LA MEDIANE A 5 DE 7 5 8 6 9 EST 7 *
* LA MEDIANE A 5 DE 5 8 6 9 3 EST 6 *
* LA MEDIANE A 5 DE 8 6 9 3 5 EST 6 *
* LA MEDIANE A 5 DE 6 9 3 5 4 EST 5 *
* LA MEDIANE A 5 DE 9 3 5 4 3 EST 4 *
* LA MEDIANE A 5 DE 3 5 4 3 3 EST 3 *

CHAPITRE 5

TITRE : CONCEPTION ET DESSIN DES MASQUES D'UN CIRCUIT
INTEGRE POUR LE LISSAGE PAR MEDIANE

RESUME : On étudie l'architecture et le comportement
d'un circuit intégré réalisant l'algorithme de
la médiane à 3 voisins. On propose un dessin
des masques à l'aide de LUCIE [13]

Sous-chapitre 5.1 :

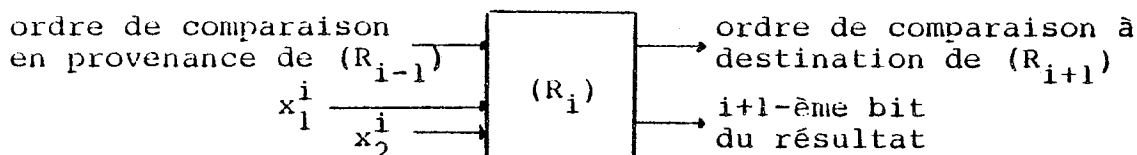
Aspect logique

La figure 1 présente le réseau de la médiane à trois voisins. En nous limitant au cas d'une suite de données entières, nous avons besoin de deux réseaux combinatoires calculant respectivement le minimum et le maximum de deux nombres codés sur n bits .

Soit (R) un réseau combinatoire recevant en entrée deux entiers x_1 et x_2 codés sur n bits et délivrant en sortie leur minimum. On notera

$$x_k = (x_k^0, \dots, x_k^{n-1}) = \sum_{i=0}^{n-1} x_k^{n-1-i} 2^i \quad k = 1, 2$$

La cellule (R_i) de traitement du $i+1$ ème bit sera de la forme



On adopte la solution suivante :

$$E_1^i : x_1[i] \quad x_2[i] \quad \text{ordre de}$$

$$E_2^i : x_2[i] \quad x_1[i] \quad \text{comparaison}$$

où $x_k[i] = (x_k^0, \dots, x_k^{i-1})$ désigne la troncature aux i premiers bits de poids fort de x_k - on a $E_1^i + E_2^i = 1$.

(R_i) délivre en sortie $E_1^{i+1} = S_1^i$ et $E_2^{i+1} = S_2^i$; l'expression de S_1^i , est donnée, en omettant les indices, par

$$S_1 = E_1 \overline{E_2} + E_1 E_2 (x_1 + \overline{x_2})$$

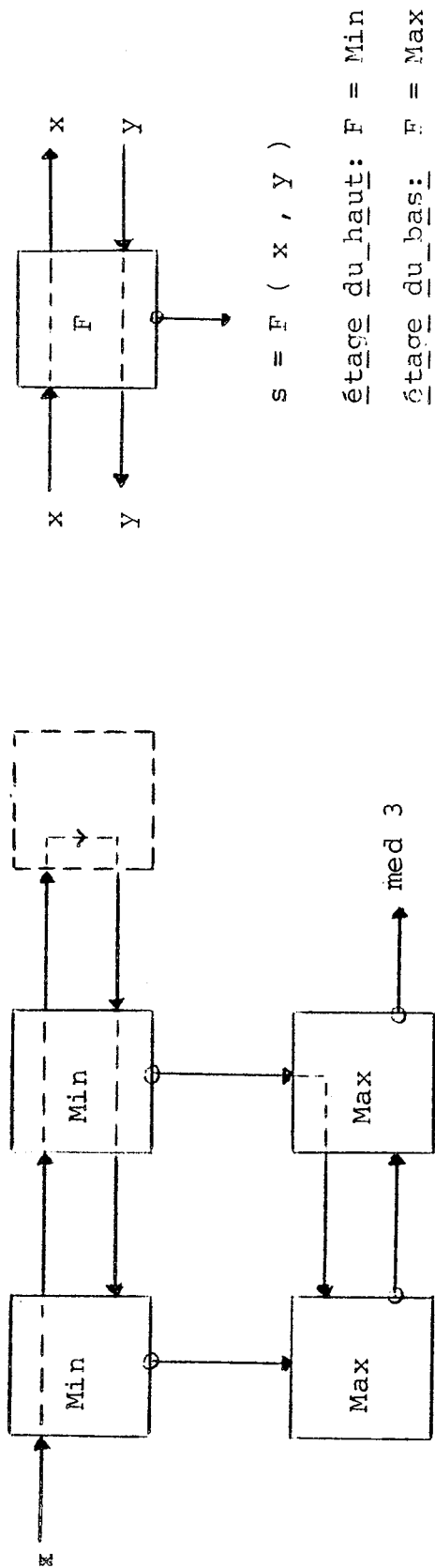
ce qui se simplifie en

$$S_1 = E_1 (\overline{E_2} + x_1 + \overline{x_2})$$

ou encore, en utilisant une synthèse en portes NOR

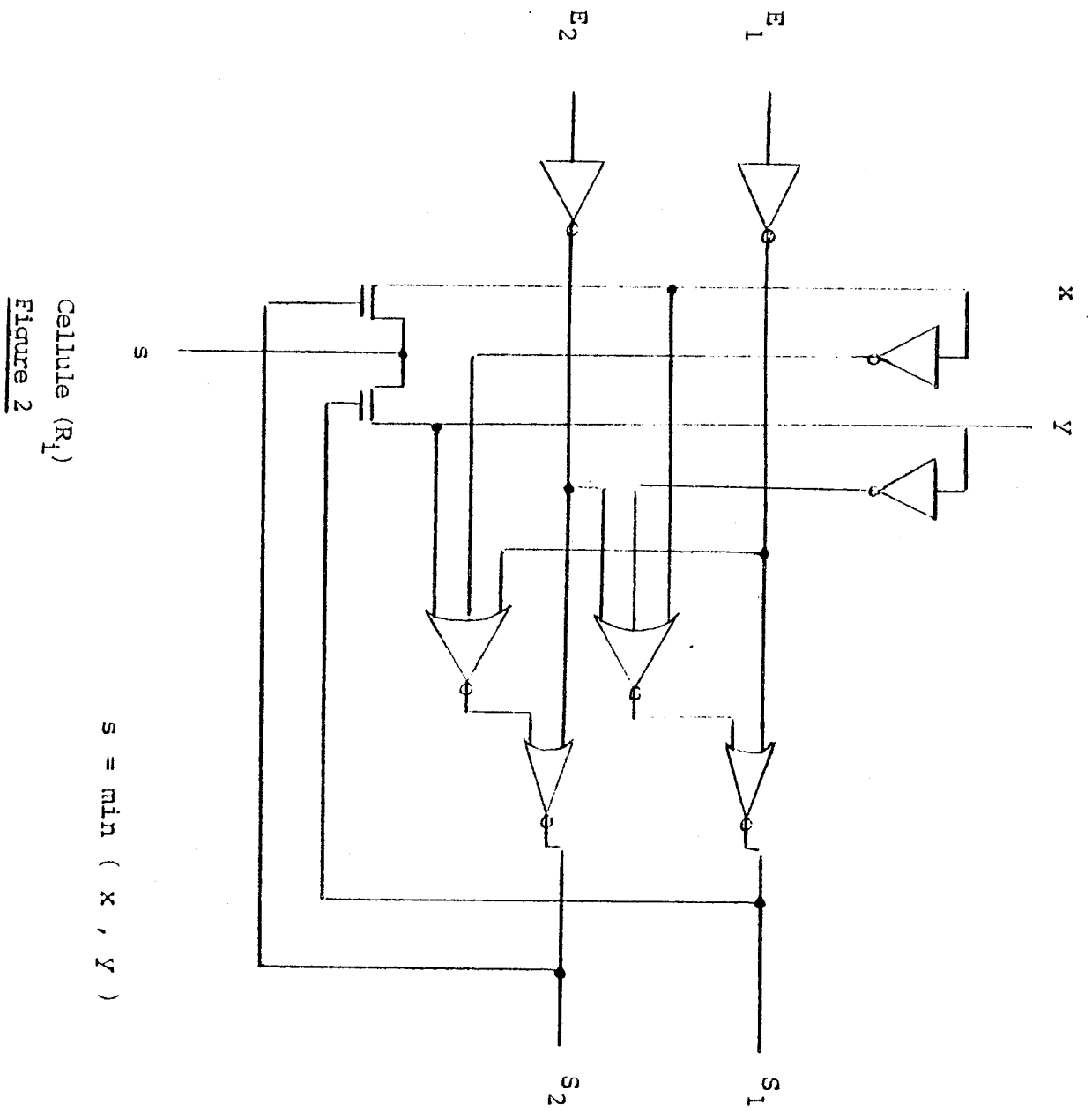
$$S_1 = \overline{E_1 + (E_2 + x_1 + \overline{x_2})}$$

Une expression analogue pour S_2^i conduit au schéma logique de la figure 2 (bien sûr, $E_1^0 = E_2^0 = 1$) .



Détail de la trajectoire des variables
dans le réseau de la médiane à trois voisins

Figure 1



Cellule (R_1)
Figure 2



Sous-chapitre 5.2 :

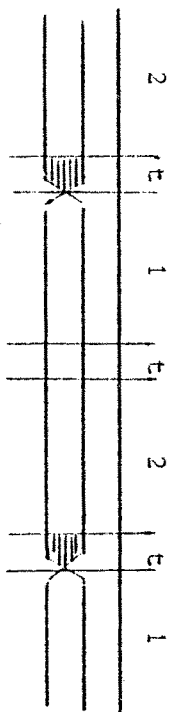
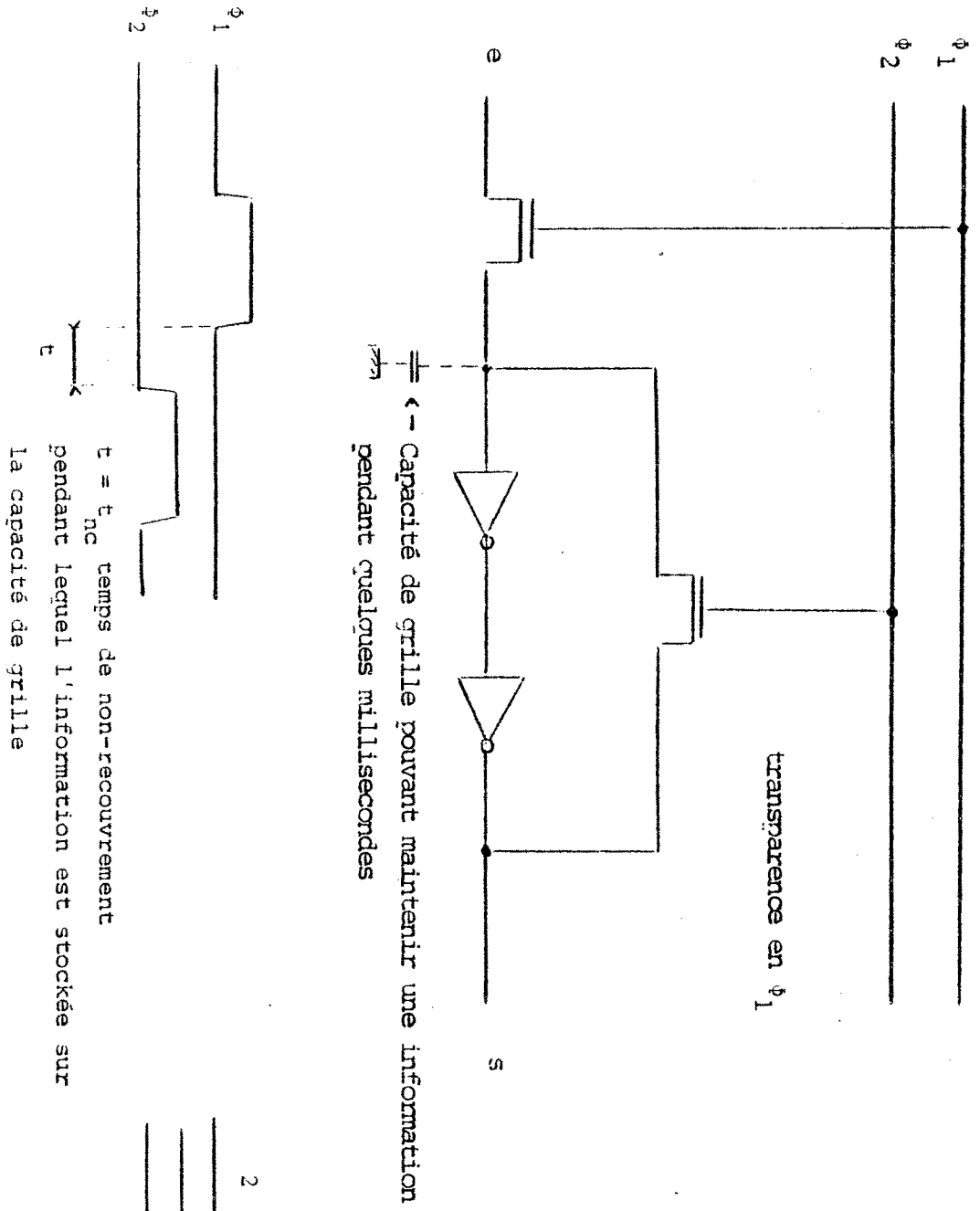
Aspect temporel

Le circuit reçoit en entrée une suite unidimensionnelle de données $(x^t)_{t \geq 0}$ qui devront être combinées à l'intérieur d'un certain voisinage local. La nécessité de disposer d'une même valeur à des instants successifs impose l'emploi d'un registre à décalage, ie d'une suite de latches commandés par des horloges alternées non recouvrantes (ϕ_1, ϕ_2) (figure 3)

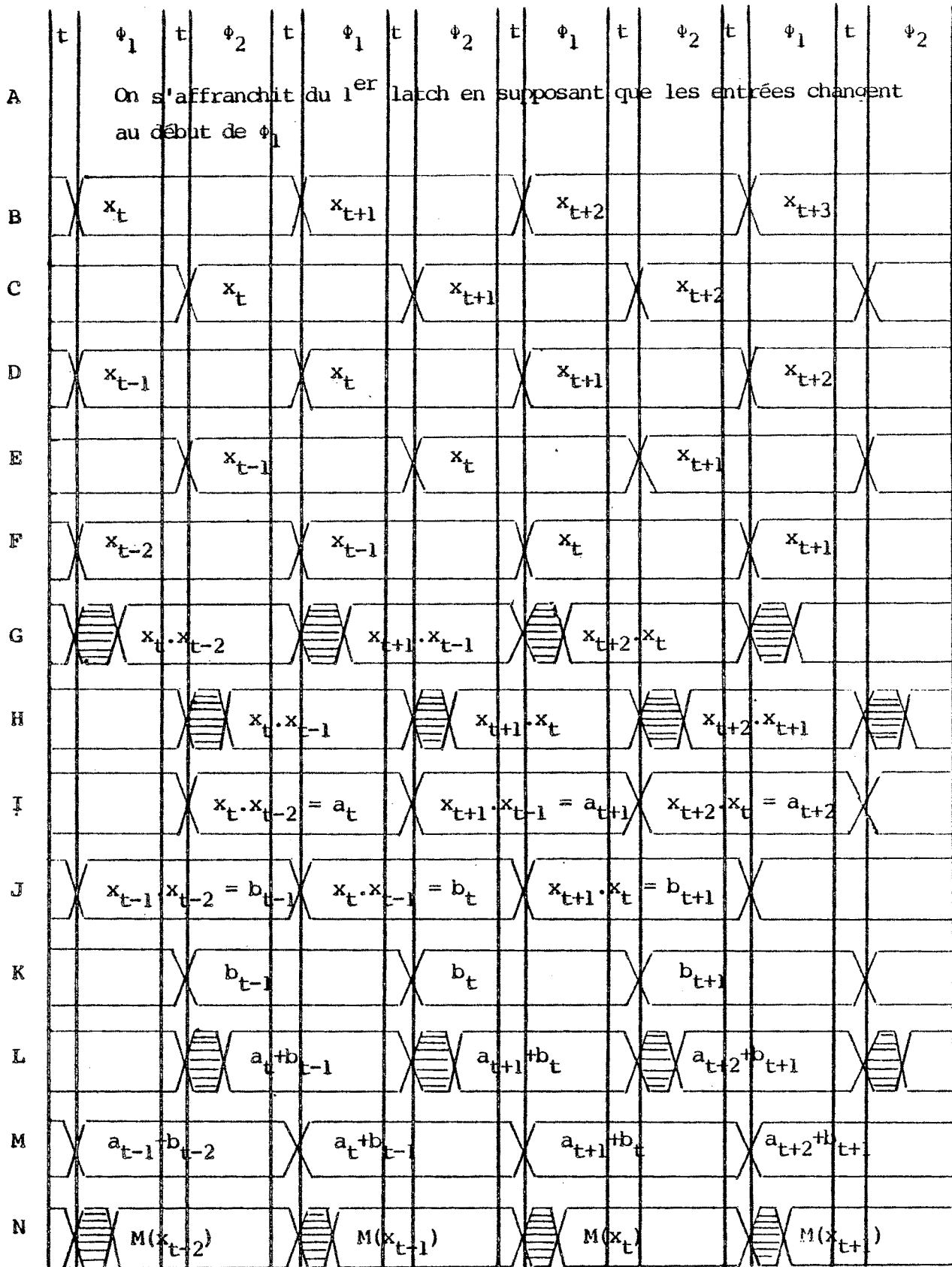
La figure 4a montre alors l'ordonnancement temporel du circuit et conduit au chronogramme de la figure 4b. Enfin, la figure 5 précise l'agencement spatial des différentes cellules: on retrouve bien le schéma de principe présenté figure 1 .

note : Cette démarche sera reprise et détaillée en [07], et complétée par

- l'étude des entrées-sorties (communication avec le monde extérieur)
- la détermination d'une fréquence de fonctionnement à l'aide d'une simulation électrique .



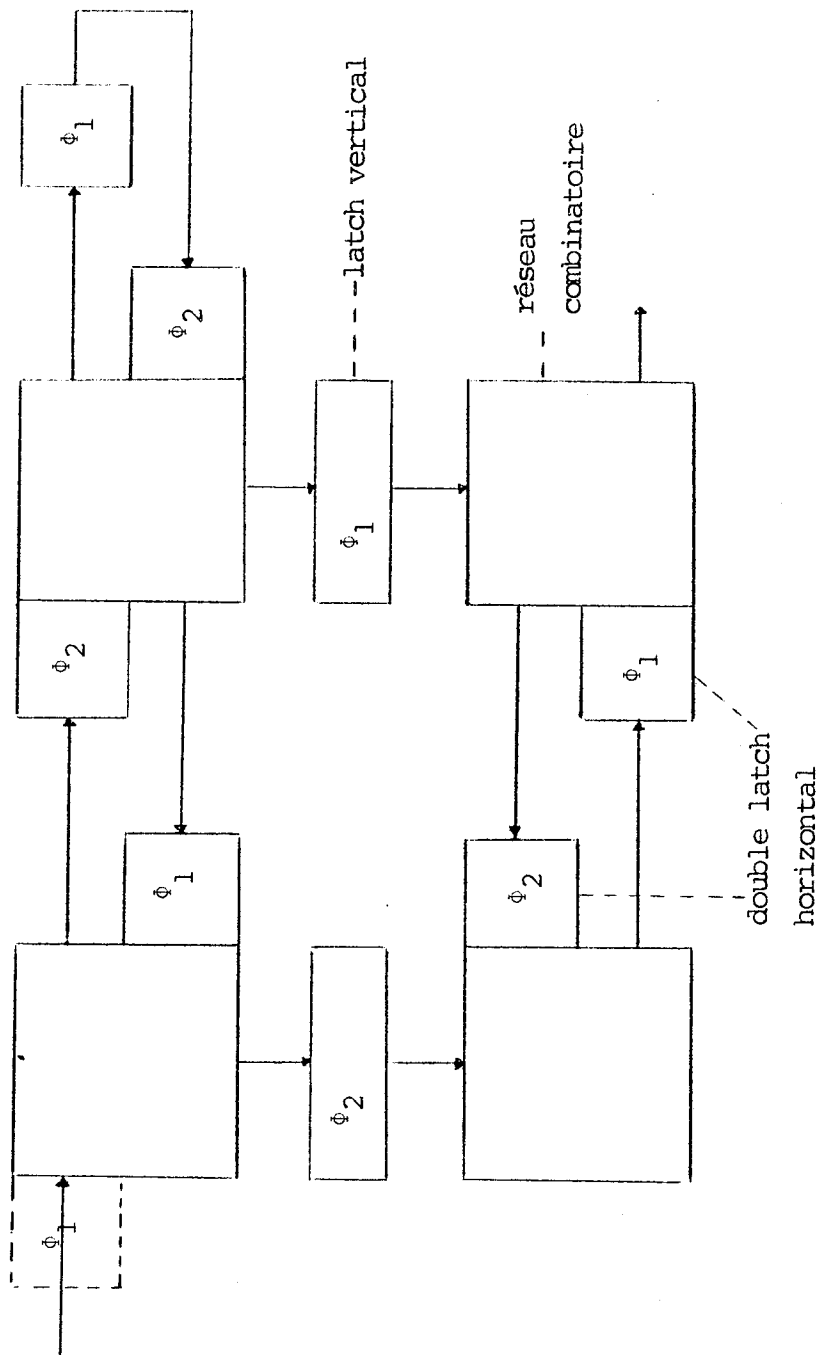
Registre semi-statique ou latch
Figure 3



On a noté : $u.v = \text{Min}(u,v)$ et $u+v = \text{Max}(u,v)$

$$M(x_i) = \text{Med}(x_{i-1}, x_i, x_{i+1}) = a_{i+1} + b_i + b_{i+1}$$

Chronogramme
Figure 4b



Agencement spatial et temporel du circuit
Figure 5

Sous-chapitre 5.3

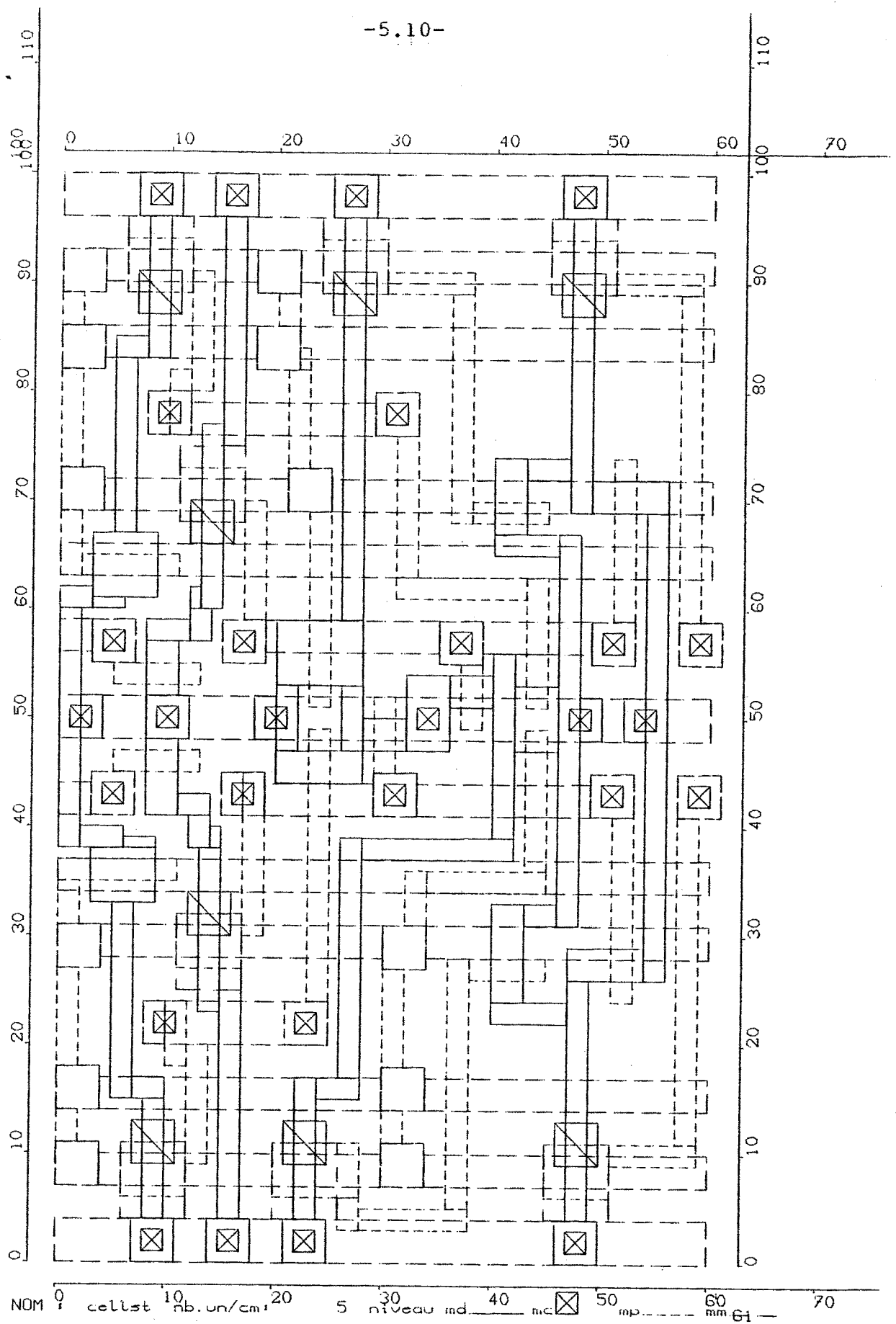
Dessin des masques

La technique d'implantation utilisée est la technique double-bande [03] selon laquelle la hauteur d'une cellule est déterminée par le nombre de lignes d'aluminium qui la traversent entre les lignes d'alimentation .

Le coût de conception est réduit par l'utilisation de briques programmables, pouvant se connecter à n'importe quel bit des bus d'entrées qui la traversent: les contacts effectifs sont réalisés après assemblage. On a choisi de traiter des nombres de 4 bits .

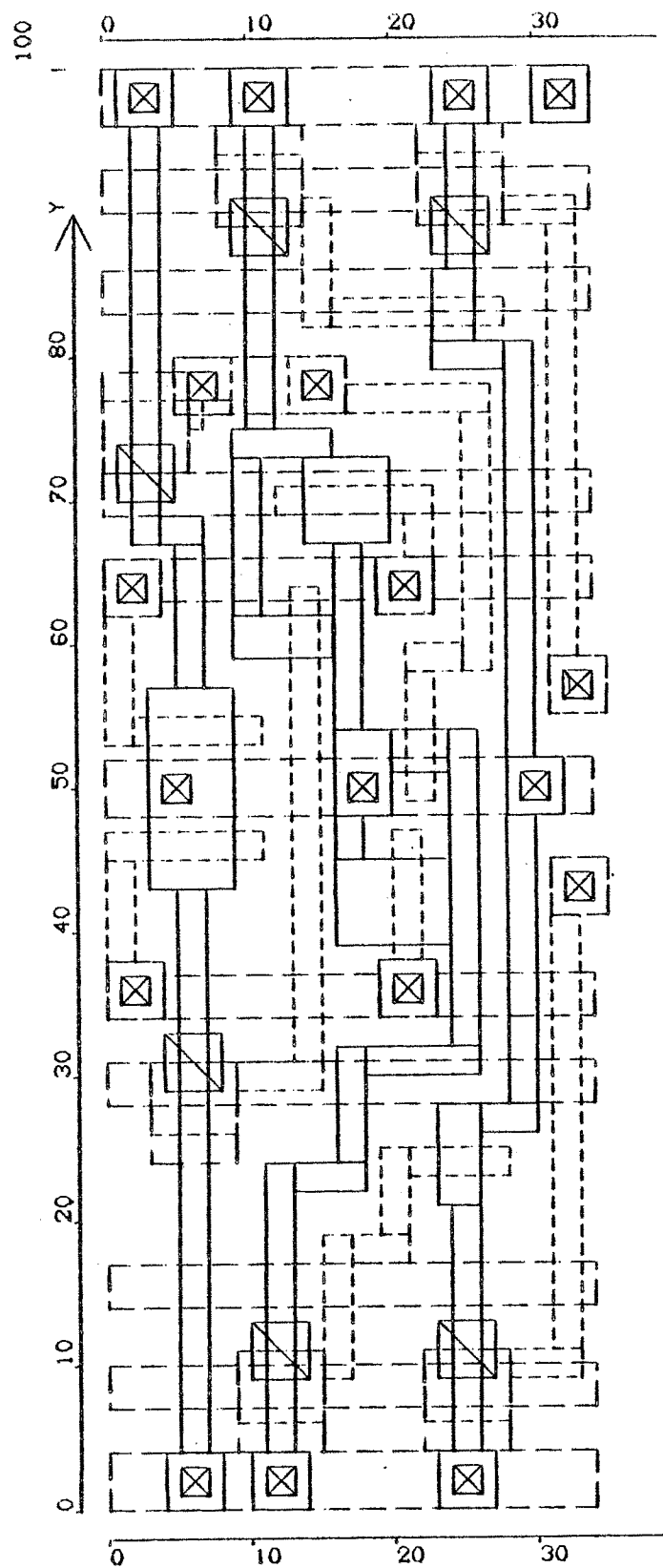
On trouvera

- figure 6: le dessin d'une cellule standardisée (R_i), sans génération de sortie
- figure 7: le dessin de (R_1) tenant compte des simplifications dues aux relations $E_1^0 = E_2^0 = 1$
- figure 8: le dessin des cellules générant la sortie des (R_i), fonction MIN ou fonction MAX
- figure 9: le dessin d'un double latch horizontal
- figure 10: le dessin des deux latches verticaux



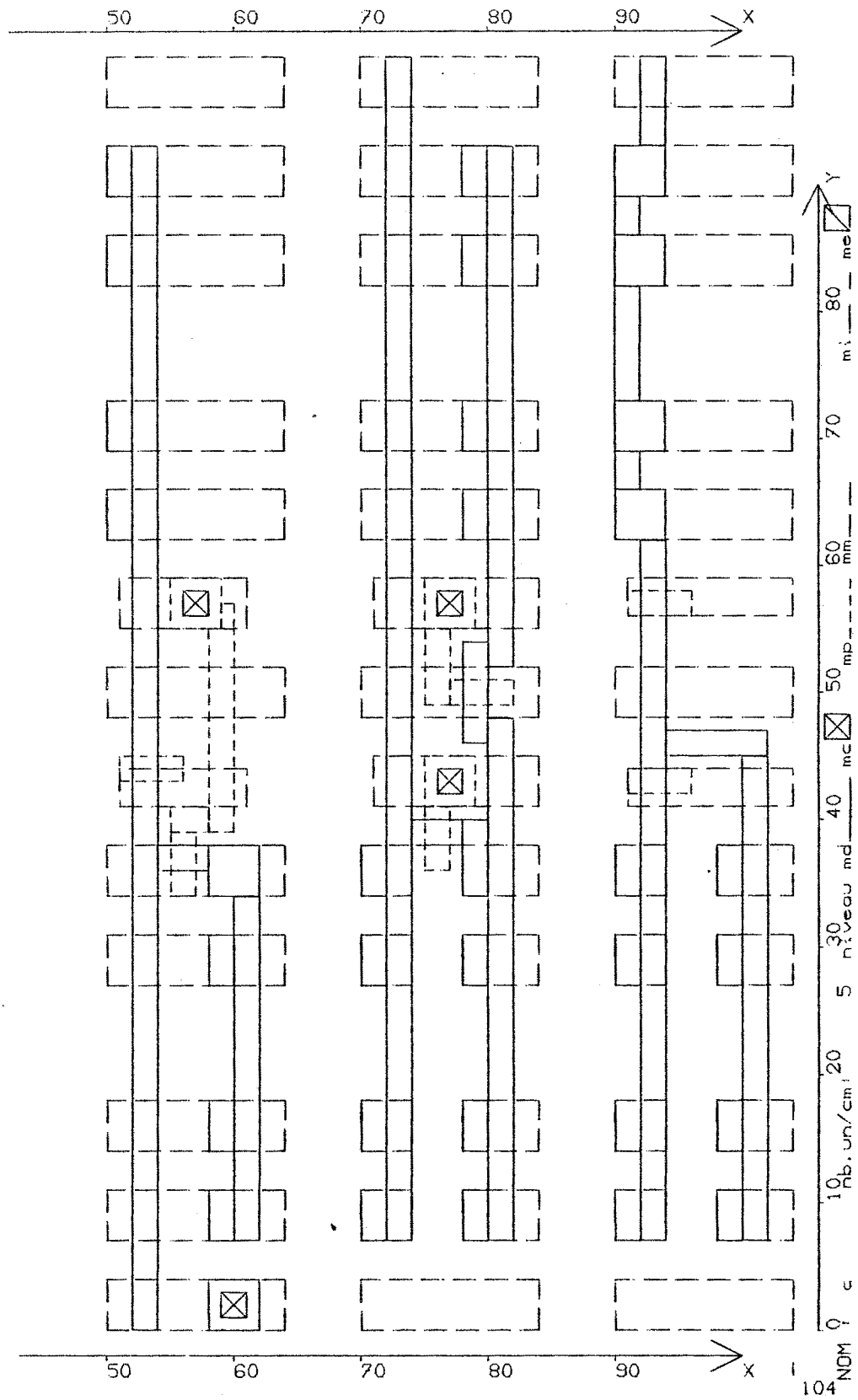
Cellule (R₁)

Figure 6



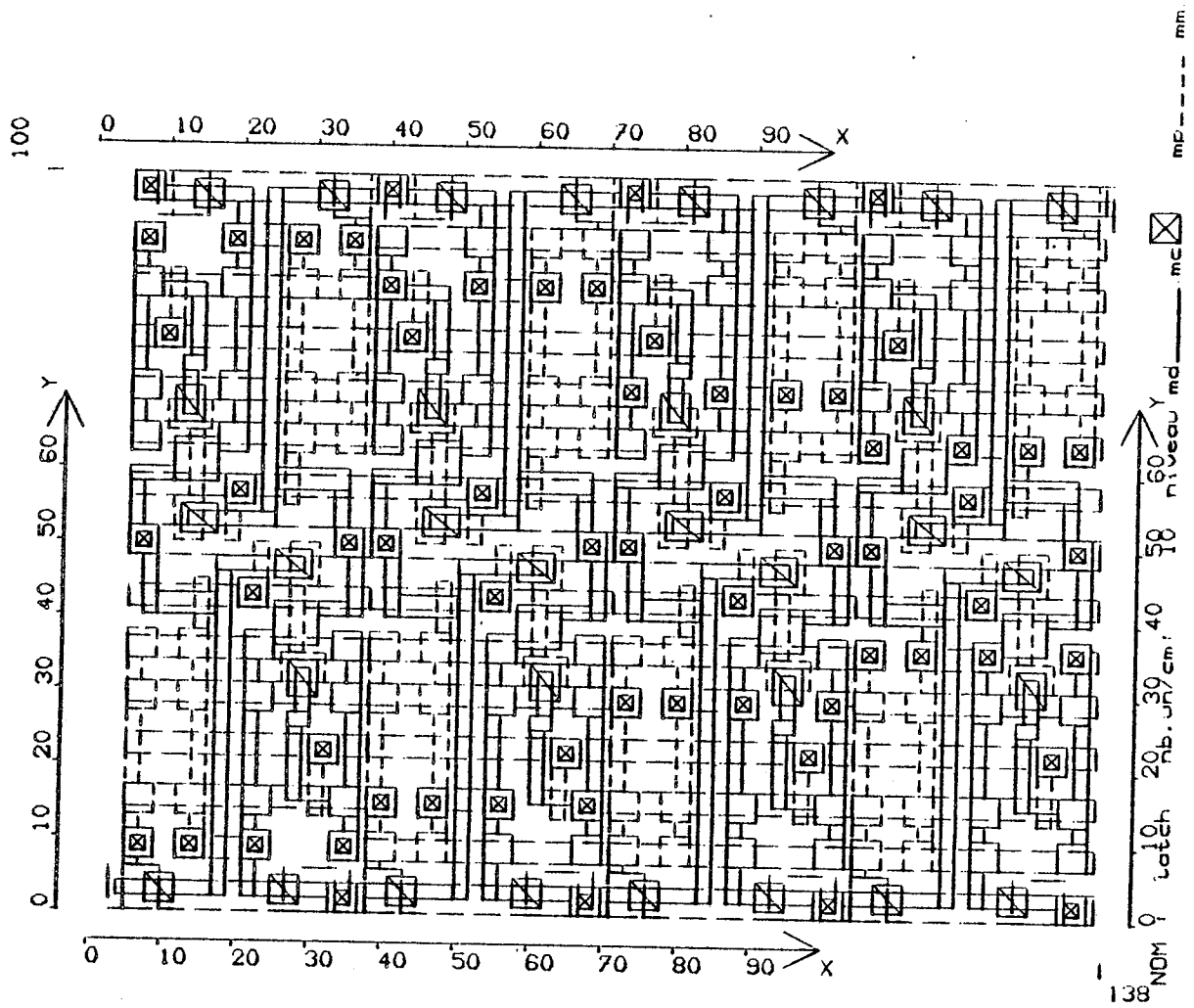
Cellule (R_1)

Figure 7



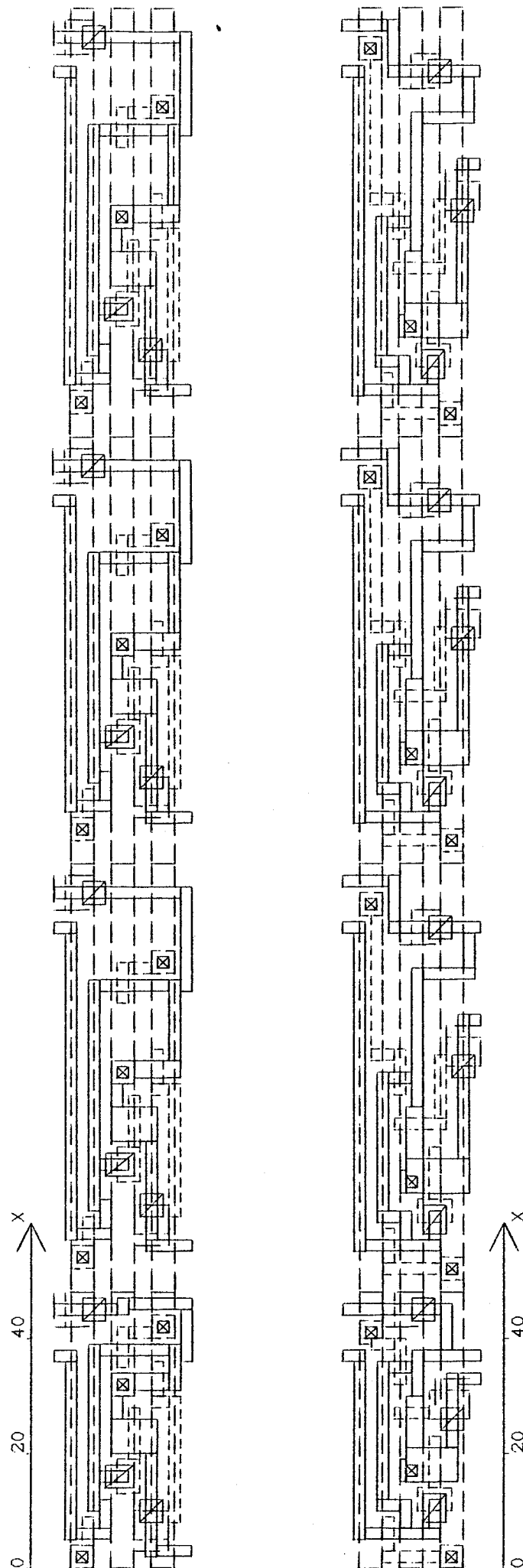
Cellule générant la sortie des (R_i)

Figure 8.



Double latch horizontal

Figure 9



Les deux latches verticaux

Figure 10

CONCLUSION

Nous avons abordé dans ce travail trois thèmes complémentaires :

- la conception d'algorithmes systoliques
- la vérification de leur validité
- la réalisation de ces algorithmes

Cette approche nous amène à mettre en relief les points suivants :

A) La richesse du modèle

Les divers exemples traités montrent que le modèle systolique combine les deux types classiques de parallélisme :

- le parallélisme dans l'espace : pour calculer une fonction F , on la décompose en opérations élémentaires pouvant être effectuées simultanément

- le parallélisme dans le temps : lorsqu'on doit calculer $F(x_t)$, $t = 0, 1, \dots$ on peut commencer le calcul de $F(x_t)$ avant d'avoir terminé celui de $F(x_{t-1})$.

Ces deux caractéristiques permettent, dans la plupart des cas, d'obtenir des algorithmes en temps réel, ie où les résultats sont délivrés au même rythme que les entrées .

De plus, la régularité et la modularité des réseaux obtenus permettent l'implémentation sur silicium de configurations comportant un très grand nombre de cellules (de telles tentatives ont été faites à l'Université de Carnegie Mellon) .

B) Le problème de la validité

La simulation du chapitre 4 serait plus convaincante si elle était cablée. Quoi qu'il en soit, vu la lourdeur de la méthode, une justification formelle (comme celle de l'annexe 2 pour l'algorithme de la médiane) reste préférable .

C) Une lacune grave au niveau de la conception

Il manque au concepteur un outil visant, sinon à automatiser, du moins à proposer des règles élaborées pour la transformation d'un algorithme usuel en un algorithme systolique. La démarche esquissée au chapitre 1 reste imprécise, et l'approche analytique de l'annexe 1, satisfaisante pour l'esprit, apparaît beaucoup trop compliquée dans la pratique pour pouvoir déboucher sur une construction effective .

Sans doute faut-il voir là l'explication du faible nombre d'algorithmes fortement non linéaires qu'on a pu modéliser sous forme systolique à ce jour (par exemple, il n'existe pas encore de réseau systolique pour un algorithme aussi usuel que la F.F.T.) .

REFERENCES

Les références données concernent seulement les réseaux systoliques; on renvoie à l'annexe 3 pour une bibliographie sur les factorisations incomplètes .

CMU : Carnegie Mellon University

TR : Technical Report

- [01] AHMED H.M., DELOSME J.M., MORF M., Higly concurrent computing structures for matrix arithmetic and signal processing, Computer magazine, January 1982 p 65-82
- [02] ANCEAU F., GUYOT A., RAYMOND J., Présentation du projet CMP, Rapport Equipe de recherche en architectures d'ordinateurs IMAG Grenoble 1981
- [03] ANCEAU F., REIS R., Complex integrated circuits design strategy, RR n° 267 IMAG Grenoble 1981
- [04] BANATRE J.P., FRISON P., QUINTON P., Un réseau de processeurs pour la détection des mots dans la parole continue, Note Technique INRIA 1982
- [05] BANATRE J.P., FRISON P., QUINTON P., Un algorithme systolique pour la reconnaissance de mots connectés, Note Technique INRIA 1982
- [06] BOJANCZYK A., BRENT R.P., KUNG H.T., Numerically stable solutions of dense systems of linear equations with mesh connected processors, T R, CMU 1981
- [07] CLAUZEL Y., ROBERT Y., Un circuit intégré systolique pour le lissage par médiane, RR IMAG Grenoble, à paraître 1983
- [08] DAVIS A.L., WEISER U., Mathematical representation for VLSI arrays, T R, University of Utah 1980
- [09] FISHER A.L., Systolic algorithms for running order statistics in signal and image processing, T R, CMU 1981
- [10] FOSTER M.J., KUNG H.T., The design of special purpose VLSI chips, Computer magazine, January 1980 p 26-40
- [11] FOSTER M.J., KUNG H.T., Recognize regular languages with programmable building blocks, in Proceedings of international VLSI conference, J.P. GRAY ed, Academic Press, August 1981 p 75-84

- [12] GENTLEMAN W.M., KUNG H.T., Matrix triangularization by systolic arrays, Proceedings of the SPIE, vol 298, Real time signal processing 4, August 1981
- [13] GUYOT A., JERRAYA A., RAYMOND J., LUCIE Langage Universitaire Conception de circuits Intégrés pour l'Enseignement, Rapport Equipe de recherche en architectures d'ordinateurs IMAG Grenoble 1981
- [14] HACQUES G., Algorithmique numérique (volume 1), Collection U Mathématiques pour l'informatique n°3, Armand Colin , Paris 1971
- [15] KUNG H.T., Let's design algorithms for VLSI systems, T R, CMU 1979
- [16] KUNG H.T., Special-purpose device for signal and image processing, Proceedings of the SPIE, vol 241, Real time signal processing 3, July 1980, p 76-84
- [17] KUNG H.T., The stucture of parallel algorithms, Advances in computers vol 19 (1980) p 65-112
- [18] KUNG H.T., Why systolic architectures, T R, CMU 1981
- [19] KUNG H.T., Use of VLSI in algebraic computation: some suggestions, Proceedings of the 1981 ACM symposium on symbolic and algebraic computation, Wang P.S.ed, ACM SIGSAM, August 1981 p 218-222
- [20] KUNG H.T., LEHMAN P.L., Systolic (VLSI) arrays for relational database operations, T R, CMU 1979
- [21] KUNG H.T., LEISERSON C.E., Systolic arrays for (VLSI), in [24] p 271-292
- [22] KUNG H.T., SONG S.W., A systolic 2D-convolution chip, T R, CMU 1981
- [23] LEHMAN P.L., A systolic (VLSI) array for processing simple relational queries, Proceedings of the CMU conference on VLSI systems and computations, KUNG H.T., SPROULL R.F., STEELE G.L.Jr ed, Computer science press, October 1981, p 285-295
- [24] MEAD C., CONWAY L., Introduction to VLSI systems, Addison-Wesley 1980
- [25] PREPARATA F.P., VUILLEMIN J., Area-time optimal VLSI networks for multiplying matrices, Information processings letters vol 11 (1980) n°2 p 77-80

- [26] ROBERT Y., TCHUENTE M., Organisation du calcul en parallèle de récurrences vectorielles, RR n°261 IMAG Grenoble 1981
- [27] ROBERT Y., TCHUENTE M., Calcul en parallèle sur des réseaux systoliques, S.A.N.G. n°368 Grenoble 1981
- [28] ROBERT Y., TCHUENTE M., Systolic algorithms for convolution and median smoothing, RR n°295 IMAG Grenoble 1981
- [29] ROBERT Y., TCHUENTE M., Quelques exemples d'algorithmes systoliques, Journées algorithmiques de Limoges 1982
- [30] ROBERT Y., TCHUENTE M., Calcul en parallele sur des réseaux systoliques, Conference "Méthodes vectorielles et parallèles en calcul scientifique", GAMNI, Mars 1983
- [31] ROBERT Y., Algorithmes systoliques pour la résolution des systèmes creux (LU et factorisation incomplète de matrices bandes), Journée G.A.M.N.I. à Lyon, 9 Décembre 1982

ANNEXE 1

TITRE : " Organisation du calcul en parallèle de
récurrences vectorielles "

On détaille ici les résultats présentés au sous-
chapitre 2.1 .

RESUME

On étudie des réseaux composés de cellules élémentaires permettant de modéliser le calcul en parallèle de suites et de récurrences linéaires à p termes en dimension n . Dans le cas d'une suite, nous proposons un réseau régulier d'efficacité maximale $e_{\max} = 1/2$ qui généralise celui de H.T. KUNG [1]. Pour le calcul récurrent nous établissons la majoration $e_{\max} \leq 1/n+1$ et nous exhibons un réseau régulier cylindrique d'efficacité $e = 1/2n$. Les deux problèmes sont donc de complexité différente dès la dimension deux contrairement à la similarité observée en dimension un [1].

ABSTRACT

We study systolic systems, i.e networks of processors which rhythmically compute and pass data through the system, for the evaluation of n -dimensional linear sequences or recurrences. For sequences, we propose a regular network of efficiency $e_{\max} = 1/2$, generalizing H.T.KUNG [1]. For recurrences, we show that $e_{\max} \leq 1/n+1$ and we exhibit a regular cylindric network of efficiency $e = 1/2n$. Thus the two problems share the same complexity in the 1-dimensional case [1] but this is not the case for dimension two or more.

I - INTRODUCTION

Nous disposons de processeurs ou cellules élémentaires ayant tous la même structure, et pouvant être connectés entre eux. Chaque processeur a trois registres :

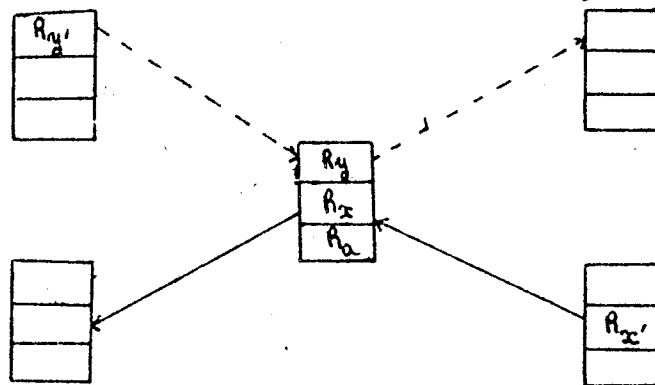
R_x permet de stocker une variable x

R_y permet de stocker une variable y

R_a contient une variable locale qui sera dans la suite une entrée de matrice .

Chaque cellule ^{un} mode de fonctionnement semblable à celui du coeur humain : à tout instant elle absorbe deux variables x , y provenant respectivement des registres R_x et R_y des cellules qui lui sont voisines en entrée, elle effectue une opération élémentaire et l'instant d'après elle envoie ces variables x , y respectivement dans les registres R_x et R_y des cellules qui lui sont voisines en sortie .

Plus précisément, si à l'instant t , la configuration du réseau au voisinage d'une cellule (C) est de la forme



alors, l'état de la cellule (C) à l'instant $t+1$ est défini par

$$R_x := R_{x'}$$

$$R_y := R_{y'} + R_a \cdot R_{x'}$$

$$R_a := R_a$$

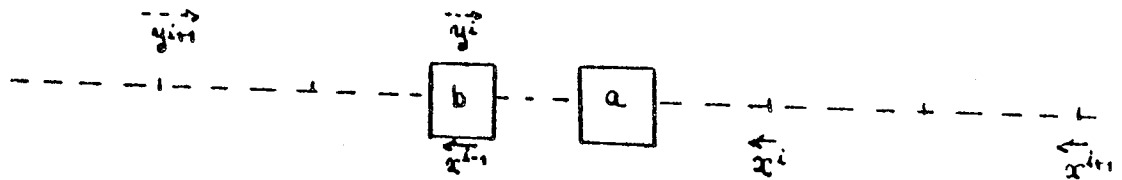
Soit (P) un problème donné. Notons $\phi(k)$ le temps nécessaire à la résolution de (P) sur un réseau comportant k processeurs. Une bonne mesure de l'efficacité du parallélisme est donnée par la quantité

$$e = \frac{\phi(1)}{k \cdot \phi(k)}$$

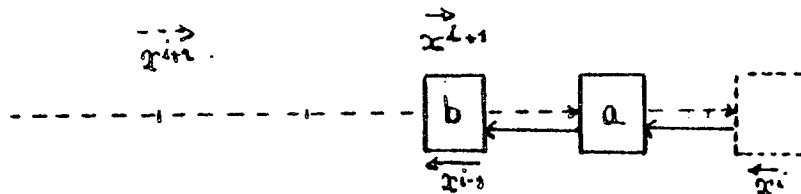
Comme on peut simuler sur un seul processeur en un temps $k \cdot \phi(k)$, un algorithme à k processeurs, on déduit que $e \leq 1$

EXEMPLE [1]

Le calcul de la suite réelle $y^i = ax^i + bx^{i-1}$, a, b réels à partir de la suite de données (x^i) , $i \geq 0$ se modélise ainsi



Pour la suite récurrente réelle $x^{i+1} = ax^i + bx^{i-1}$, avec x^0 et x^1 donnés on a



La cellule en pointillé est un processeur de retard.

Dans les deux cas, l'efficacité est $e = 1/2$ car chaque cellule est active une pulsation sur deux. Cette valeur est optimale car on vérifie aisément qu'il n'est pas possible de définir un algorithme avec un écart d'une unité entre deux itérés successifs.

Le calcul d'une suite a donc en dimension un la même complexité que le calcul récurrent. Nous verrons dans la suite qu'il n'en est pas de même pour les dimensions deux ou plus.

II - CALCUL EN TEMPS REEL D'UNE SUITE VECTORIELLE .

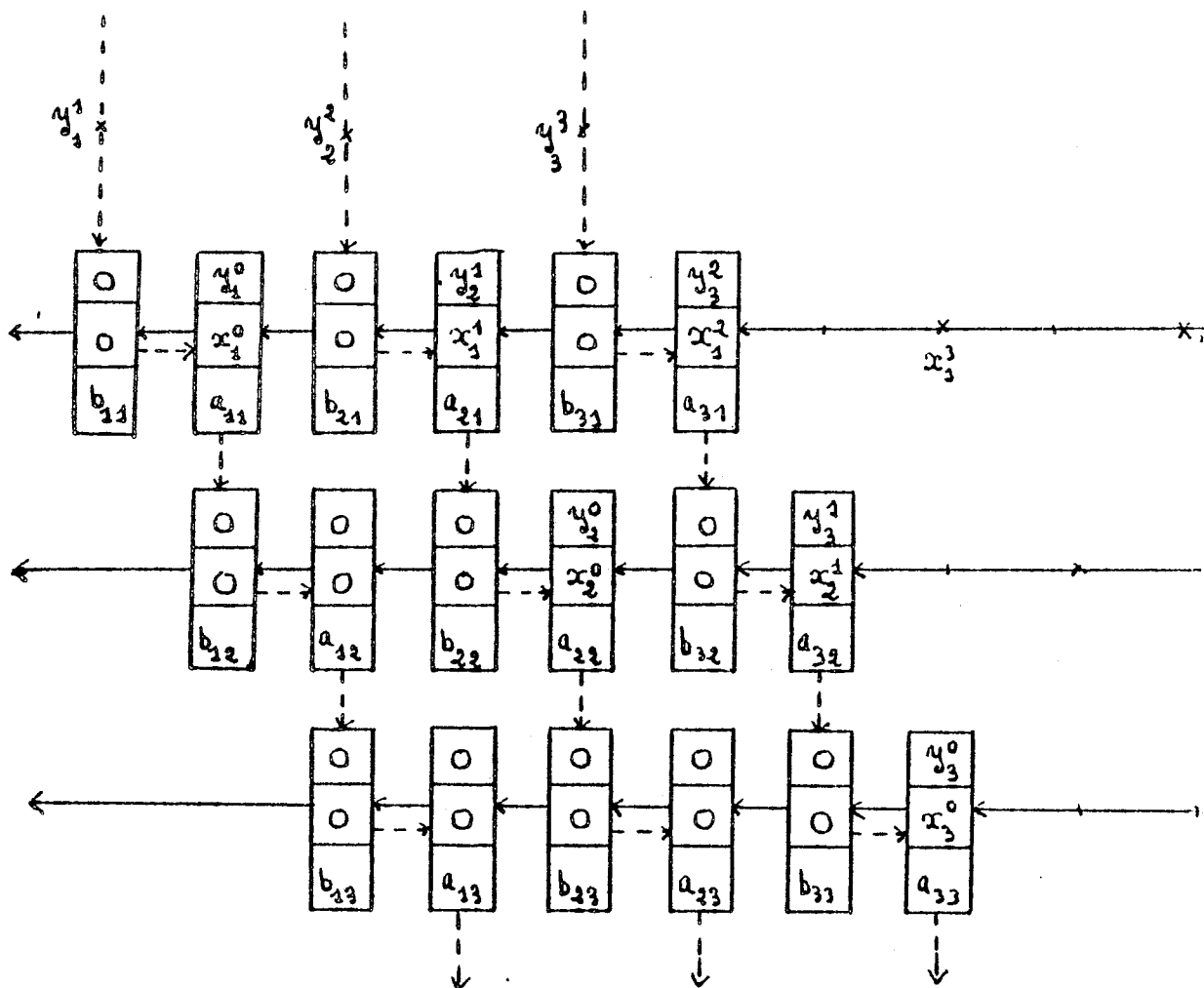
Soit A,B deux matrices carrées d'ordre n . On souhaite donner la configuration d'un réseau de $2n^2$ cellules permettant de calculer à partir d'une suite (x^i) , $i = 0,1,2, \dots$ de données vectorielles à n composantes, la suite (y^i) , $i = 1,2, \dots$ définie par $y^i = Ax^i + Bx^{i-1}$, ce qui en terme de composantes s'écrit :

$$y_j^i = \sum_{k=1}^n a_{jk} x_k^i + \sum_{k=1}^n b_{jk} x_k^{i-1} \quad (i \geq 1, 1 \leq j \leq n) .$$

Un tel réseau est proposé pour $n = 3$ à la figure 1 . Nous avons explicité quelques pulsations consécutives à la figure 2 pour le cas $n = 2$.

En négligeant le temps d'initialisation qui est de l'ordre de $2n$, on voit que l'efficacité de l'algorithme est $e = 1/2$ car chaque cellule est active une pulsation sur deux .

Le schéma précédent s'étend aisément à l'évaluation de la suite $y^{i+p-1} = A^{(p-1)} x^{i+p-1} + \dots + A^{(1)} x^{i+1} + A^{(0)} x^i + b$ où les matrices $A^{(k)}$ sont des matrices carrées d'ordre n et b est un vecteur à n composantes . Pour cela il suffit d'initialiser les coefficients y_j^i à la valeur b_j . Nous donnons un exemple de réseau à la figure 3 . L'efficacité est encore $e = 1/2$.



$$Y^{i+1} = AX^{i+1} + BX^i$$

Figure 1

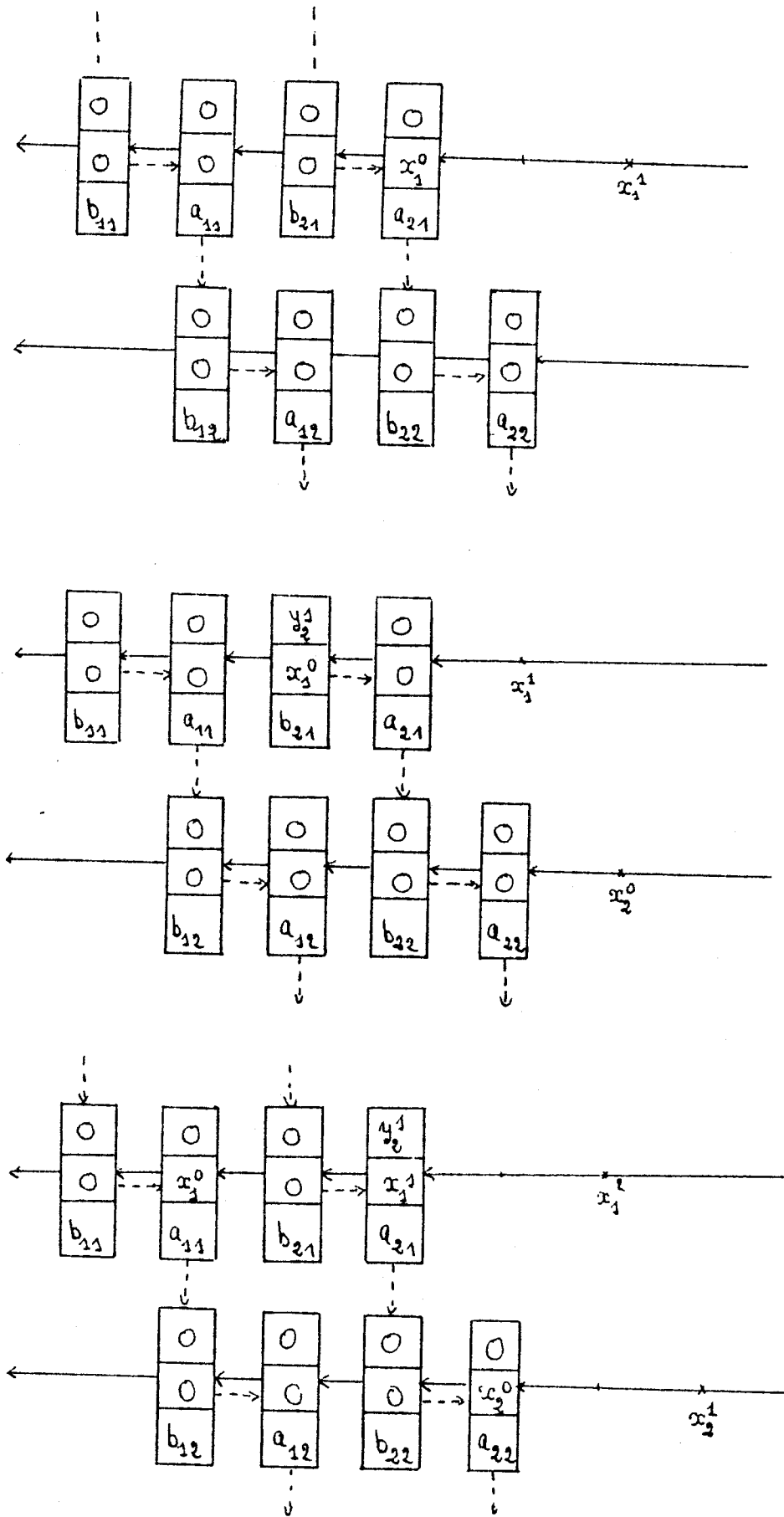


Figure 2a

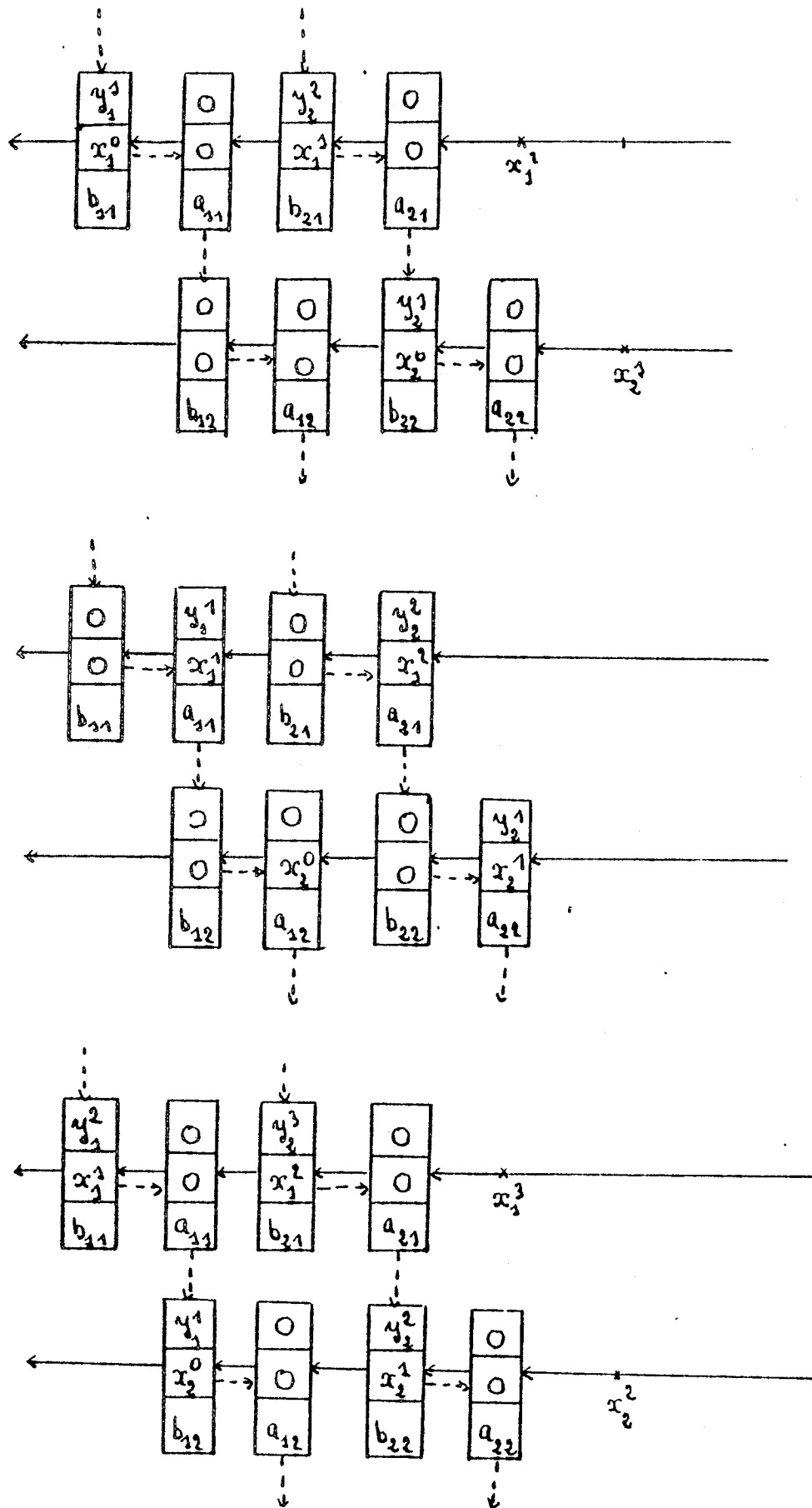
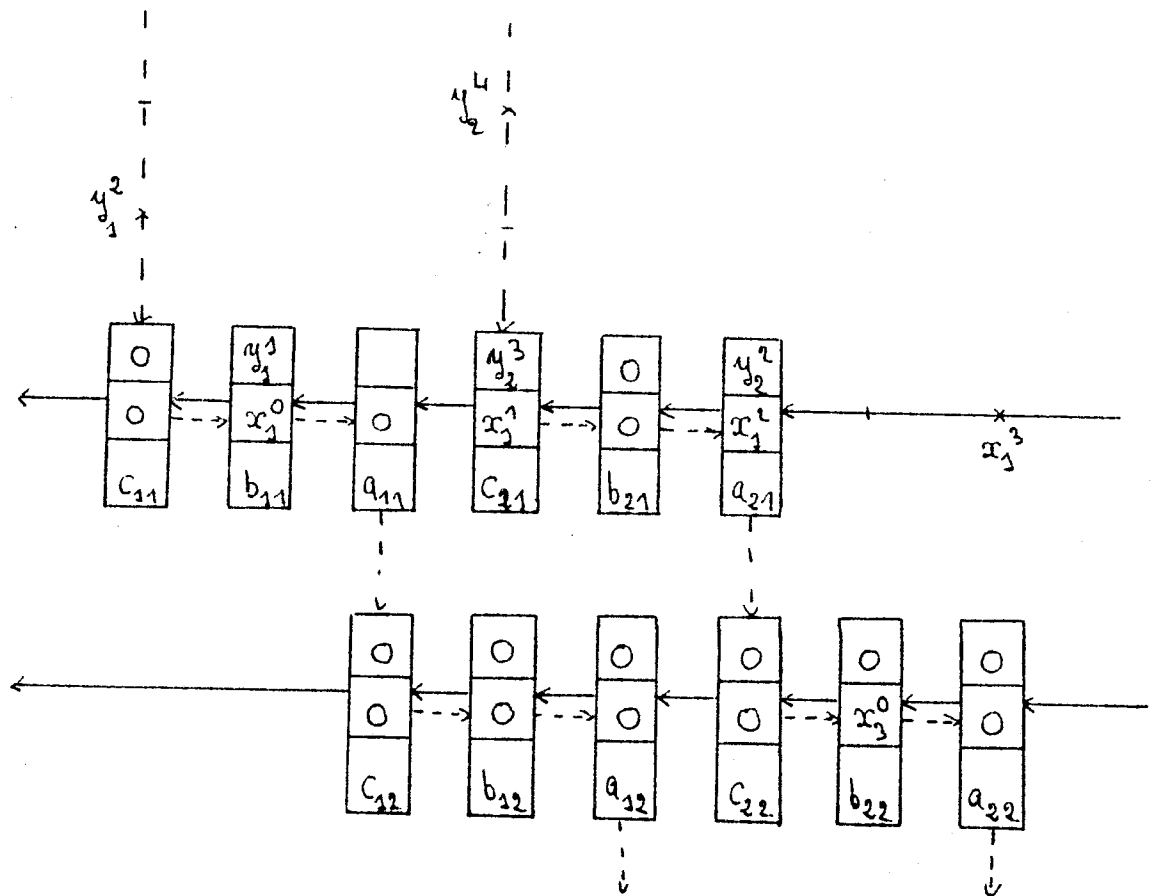


Figure 2b



$$y^{1+2} = Ax^{1+2} + Bx^{1+1} + Cx^1$$

Figure 3

III RECURRENCE LINEAIRE A p TERMES EN DIMENSION n .

$$X^i = AX^{i-1} + BX^{i-2} + b^i, \quad X^0, X^1, b^1 \text{ donnés .}$$

Nous disposons de $2n^2$ cellules . Nous pouvons toujours supposer qu'elles sont disposées selon n lignes comportant chacune $2n$ cellules, et que chaque X_j^i reste sur la ligne j au cours de son calcul . Par contre, nous ne pouvons faire aucune hypothèse sur la trajectoire de X_j^i une fois qu'il est calculé .

Comme il s'agit d'un calcul où les sorties sont délivrées au même rythme que les données, l'écart (dans le temps) entre b_j^i et b_j^{i-1} est le même que celui entre X_j^i et X_j^{i-1} .

LEMME 1

- (i) X_k^h se calcule après tous les $X_j^i, j = 1, \dots, n \quad i \leq h-1$.
- (ii) L'écart d_j entre X_j^i et X_j^{i-1} est indépendant de j .

Démonstration

- (i) Se vérifie aisément par récurrence sur k, h .
- (ii) S'il n'en était pas ainsi, la propriété (i) ci-dessus serait mise en défaut au bout d'un certain temps ■

PROPOSITION 2

Tout algorithme de calcul de $X^i = AX^{i-1} + BX^{i-2} + b^i$ a une efficacité $e \leq 1/(n+1)$.

Démonstration

Soit x_1^i la composante de x^i qui est la première à se calculer, et plaçons-nous à l'instant où elle commence à accumuler sa valeur ; désignons par d l'écart entre x_1^i et x_{j-1}^{i-1} .

- si $d < n$ alors aucune composante de x_{j-2}^{i-2} ou x_{j-1}^{i-1} n'est encore calculée, ce qui est absurde .
- si $d = n$, notons J l'ensemble des indices j tels que x_j^i se calcule en même temps que x_1^i . Les x_j^i $j \in J$ rencontrent forcément des x_{j-2}^{i-2} $j \in J$ car ce sont les seules composantes de x^{i-2} et x^{i-1} à être calculées . En conséquence, x_1^{i-1} rencontre une composante de x^{i-3} . Comme aucun x_j^{i-1} ne sera disponible avant n pulsations , on voit que x_1^i aura à rencontrer au moins $n+1$ composantes de x^{i-2} , ce qui est absurde .

Nous avons ainsi montré que $d \geq n+1$ ce qui entraîne $e \leq 1/(n+1)$ ■

Dans le cas où nous faisons des hypothèses sur l'ordre d'arrivée des b_j^i $1 \leq j \leq n$ dans le réseau, on a la

PROPOSITION 3

- (i) Tout algorithme où les b_j^i entrent en même temps dans le réseau, a une efficacité $e \leq 1/(n + \lceil n/2 \rceil)$ où $\lceil u \rceil$ désigne le plus petit entier supérieur ou égal à u .
- (ii) Tout algorithme où les b_j^i entrent dans le réseau à des instants consécutifs $j = 1, \dots, n$ a une efficacité $e \leq 2/(\sqrt{3}+1)n$.

Démonstration

Supposons qu'à un instant on ait dans le réseau r (resp' r') composantes de x^{i-1} (resp x^i) et notons m (resp m') le nombre de composantes de x^{i-1} (resp x^i) qui rencontrent des composantes de x^{i-3} (resp x^{i-2}) . On vérifie que $m' \geq r' - (n-r)$ et $m \geq r - (n-m')$

L'application de cette propriété permet d'obtenir une minoration M du nombre de coefficients b_{ij} que comporte le réseau et on obtient alors les résultats annoncés en écrivant $M \leq n^2$.

(i) Posons $d = n + d'$, $d' \leq n-1$. On a alors

$$2n(n-d') = M \leq n^2$$

d'où $d' \geq n/2$. Comme d' est entier, on a en fait

$$d' \geq \lceil n/2 \rceil \quad \text{d'où} \quad e \leq 1/d \leq 1/(n + \lceil n/2 \rceil).$$

(ii) Posons $d = 2n-q$.

- Si $q-1 \leq n-q$ alors $q \leq (n+1)/2$

d'où $d \geq (3n-1)/2$ ce qui entraîne

$$e \leq 2/(3n-1) \leq 2/(\sqrt{3}+1)n.$$

- Sinon on a pour les différentes zones du réseau un nombre de coefficients b_{ij} minoré comme indiqué dans le schéma ci-dessous :

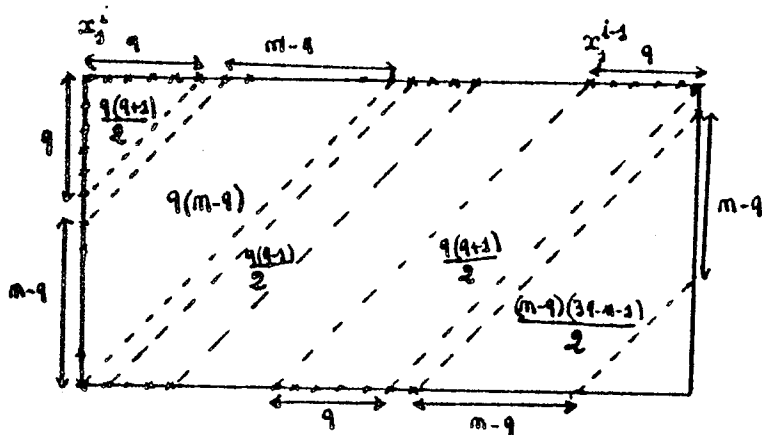


Figure 4

On en déduit que

$$-2q^2 + 2(3n+1)q - (3n^2+n) \leq 0 \quad , \quad q \leq 2n$$

d'où

$$q \leq (3n+1 - \sqrt{3n^2 + 4n + 1})/2$$

On a alors

$$d \geq 2n - (3n+1 - \sqrt{3n^2 + 4n + 1})/2$$

D'où

$$e \leq 1/d \leq 2/(\sqrt{3}+1)n$$

Remarque

La borne $1/(n + \lceil n/2 \rceil)$ obtenue dans le cas où les b_j^1 $1 \leq j \leq n$ entrent dans le réseau en même temps, peut être atteinte. Il suffit pour cela de considérer la généralisation pour n quelconque, de l'algorithme que nous donnons ci-dessous dans le cas $n = 3$,

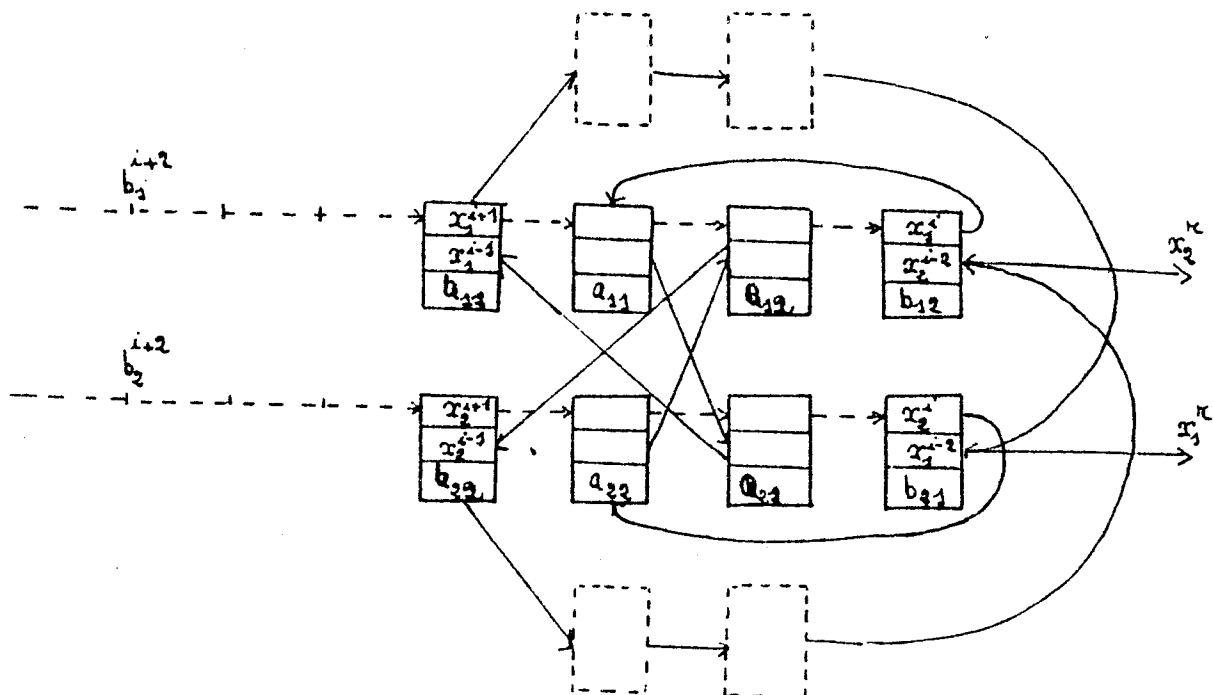


Figure 5

UN MODE D'APPROCHE

Nous allons exhiber maintenant un réseau régulier cylindrique pour le calcul de la suite récurrente

$$x^i = AX^{i-1} + BX^{i-2} + b^i, \quad x^0, x^1, b^1 \text{ données}$$

Pour la commodité de l'exposé, nous allons d'abord présenter un réseau pour le calcul de

$$y^i = AX^i + BX^{i-1} + b^i, \quad x^i, b^i \text{ données}$$

puis nous indiquerons comment ajouter des processeurs de retard pour obtenir un réseau adapté au calcul des suites récurrentes.

Nous adoptons une approche analytique. Plus précisément nous supposons que dans le référentiel ci-dessous

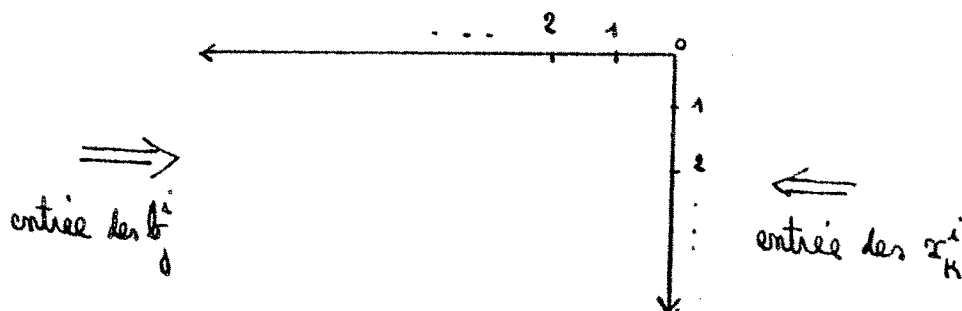


Figure 6.

les coordonnées de y_j^i et x_k^i sont linéaires par rapport à i, j, k, t , où t représente le temps ; nous supposons (sans perte de généralité), que y_j^i a une ordonnée constante et égale à j . On a alors

$$y_j^i(t) = \begin{pmatrix} -t+ai+bj \\ j \end{pmatrix} \quad x_k^i(t) = \begin{pmatrix} t+ci+dk \\ t+k \text{ module } n \end{pmatrix}$$

où $a, b, c, d \in \mathbb{Z}$

Remarquons que a (resp. c) représente le décalage entre y_j^i et y_j^{i+1} (resp. x_j^i et x_{j+1}^i) ; de même b (resp. d) représente le décalage entre y_j^i et y_{j+1}^i (resp. x_j^i et x_{j+1}^i), $1 \leq j \leq n-1$

Comme par définition du modèle, il s'agit d'un calcul où les sorties sont produites au même rythme que les entrées, et comme les y_j^i et x_k^i doivent se croiser, on a nécessairement $a = -c$. Par ailleurs l'efficacité d'un tel algorithme est $e = 1/|a| = 1/|$ et, pour la maximiser, nous devons donc choisir a aussi petit que possible,

Croisement de y_j^i et x_k^i

En notant t_{jk}^i l'instant du croisement, on a

$$\begin{cases} -t_{jk}^i + ai + bj = t_{jk}^i + ci + dk \\ t_{jk}^i + k = j \text{ modulo } n \end{cases}$$

ce qui peut encore s'écrire

$$t_{jk}^i = \frac{(a-c)i + bj - dk}{2} = j - k \text{ modulo } n$$

Comme i, j, k prennent des valeurs arbitraires entre 1 et n , on déduit que

$a - c, b, d$ sont pairs.

Nous prendrons $b = d = 2$.

Ce qui entraîne

$$t_{jk}^i = \frac{a-c}{2} i + j - k = j - k \text{ modulo } n$$

D'où

$$(1) \quad \frac{a-c}{2} = 0 \text{ modulo } n$$

Croisement de y_j^i et x_k^{i-1}

En notant t_{jk}^i l'instant du croisement, on a

$$\begin{cases} -t'_{jk} + ai + bj = t'_{jk} + c(i-1) + dk \\ t'_{jk} \cdot k = j \text{ modulo } n \end{cases}$$

En procédant comme précédemment, on déduit que

$$(2) \quad t'_{jk} = \frac{a-c}{2} i + \frac{c}{2} + j-k = j-k \text{ modulo } n .$$

En combinant les équations (1) et (2) on déduit que

$$c = 0 \text{ modulo } 2n$$

Le meilleur choix possible pour a, c est donc

$$a = -2n \quad c = 2n .$$

Finalement les coordonnées des cellules portant les coefficients a_{jk} et b_{jk} sont :

$$y_j^1(t'_{jk}) = \begin{bmatrix} j+k \\ j \end{bmatrix} \quad y_j^1(t'_{jk}) = \begin{bmatrix} j+k-n \\ j \end{bmatrix}$$

L'algorithme ainsi obtenu est d'efficacité $1/2n$; nous donnons à la figure 7 un exemple pour $n = 3$.

Pour le calcul récurrent, on adopte la méthode utilisée dans [1] pour la dimension un : on calcule les y_j^1 et on les renvoie dans le réseau après les avoir convenablement retardés. Dans le cas présent, on vérifie aisément que la ligne j doit comporter $n-j+1$ processeurs de retard (cf. figure 8).

Nous avons donc exhibé pour le calcul récurrent, un algorithme d'efficacité $1/2n$; comme nous l'avons vu au paragraphe précédent, on peut faire mieux mais le réseau obtenu ici a l'avantage d'admettre une immersion régulière dans un cylindre, c'est-à-dire qu'on peut répartir les cellules sur un cylindre de manière à ce que la distance parcourue par un signal entre deux cellules voisines soit constante. Par ailleurs, notons que, au vu de la proposition 2, il semble difficile de trouver un réseau régulier plus performant.

Enfin, cette étude se généralise aisément pour le calcul de

$$X^i = A^{(0)} X^{i-1} + A^{(1)} X^{i-2} + \dots + A^{(p-1)} X^{i-p} + b^i$$

On obtient un réseau régulier cylindrique d'efficacité $e = 1/2n$
avec pour $a_{jk}^{(r)}$, les coordonnées

$$\begin{bmatrix} j+k+rn \\ j \end{bmatrix} \quad \begin{array}{l} 1 \leq j, k \leq n \\ 0 \leq r \leq p-1 \end{array}$$

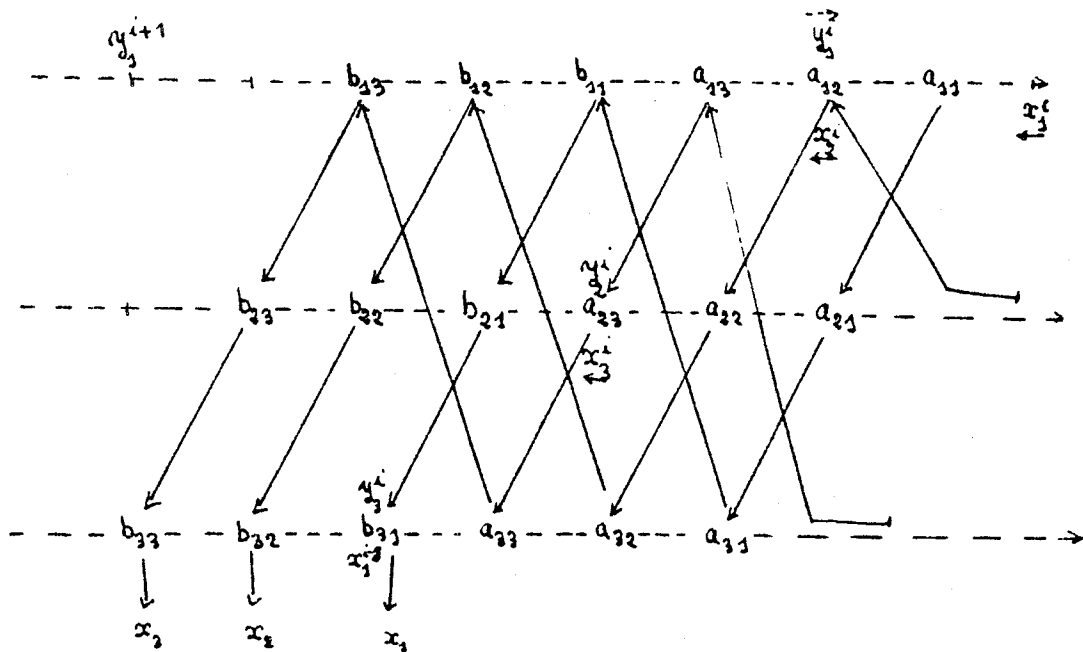


Figure 7

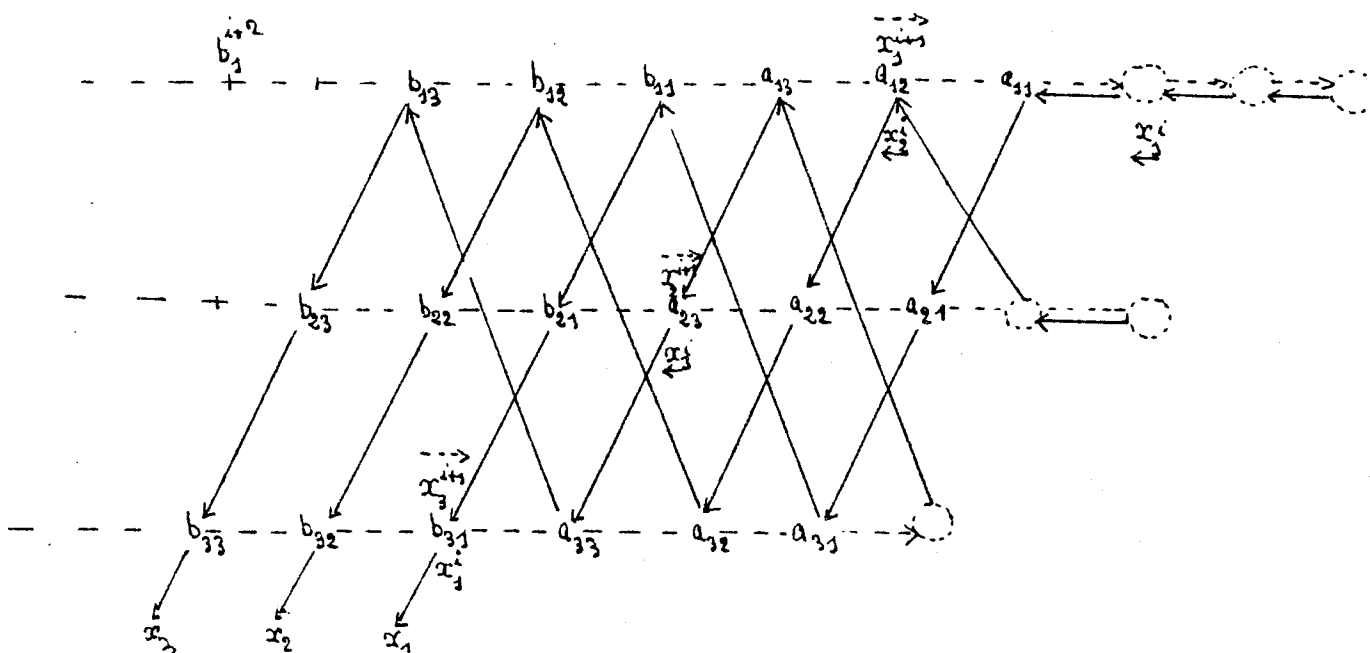


Figure 8

ANNEXE

On s'intéresse ici, au calcul de $y^i = AX^i + BX^{i-1}$ lorsque les matrices A,B comportent un grand nombre de coefficients nuls, et on désire alors réduire au minimum, les cellules correspondant à ces coefficients.

Si A(resp B) est triangulaire inférieure (resp supérieure), on obtient un algorithme d'efficacité $e = 1/2n$, en généralisant la solution exposée ci-dessous dans le cas $n=3$.

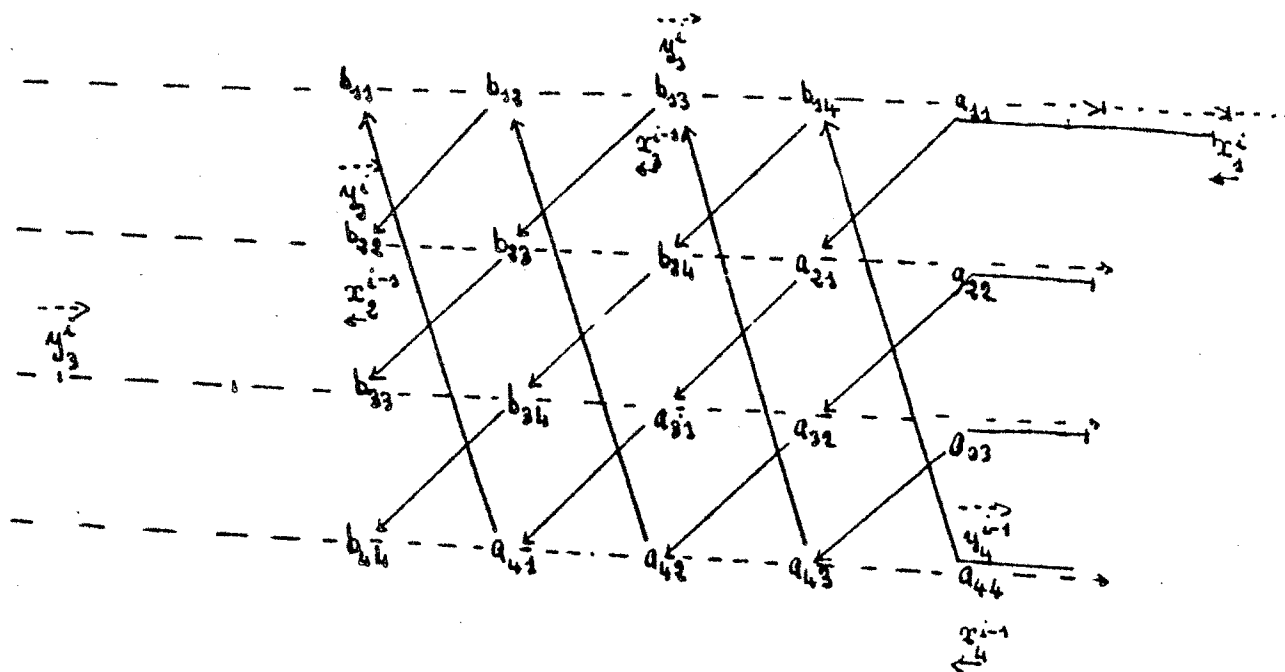
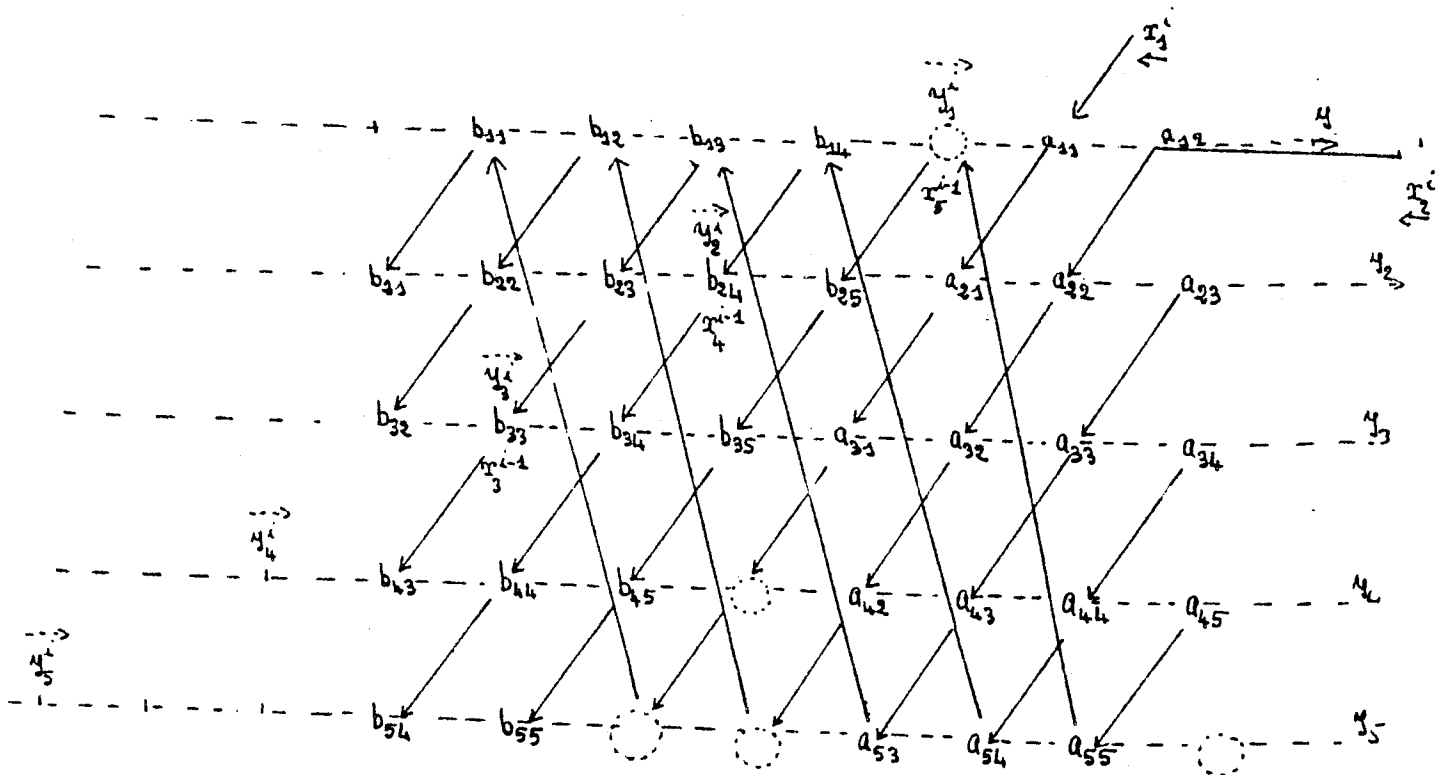


Figure 9

Si A et B ont des structures en bande, on obtient comme dans le cas triangulaire, un algorithme d'efficacité $e = 1/2n$ en généralisant celui que nous exposons ci-dessous :



A structure bande (3.2) ; B structure bande (2.4) ; n=5

Figure 10

Cette solution montre que contrairement au cas précédent (cas triangulaire), on doit recourir à quelques processeurs à coefficients nuls.

REFERENCES

[1] H.T.KUNG

The structure of Parallel Algorithms
Advances in Computers, Vol 19 (1980) p. 65-112

[2] H.T.KUNG et C.E.LEISERSON

Systolic Arrays for VLSI, dans Introduction to VLSI Systems,
C.Mead and L.Conway Ed, Addison-Wesley, 1980, p.271-292.

ANNEXE 2

TITRE : " Systolic algorithms for convolution and
median smoothing "

On détaille ici les résultats présentés aux sous-
chapitres 2.4 et 3.2 .

RESUME

On étudie des réseaux systoliques, c'est-à-dire des réseaux composés de cellules élémentaires, ayant toutes la même structure, et régulièrement agencées, permettant de modéliser un problème de lissage fréquent en traitement du signal : la convolution d'une image bidimensionnelle. Nous proposons par ailleurs une approche permettant le lissage par médiane d'une suite de données.

ABSTRACT

We study systolic systems, i.e. networks of processors which rhythmically compute and pass data through the system. We propose algorithms for the following problems (issued from image and signal processing) :

- 2D(two-dimensional) convolution filtering
- median smoothing on the Von Neumann neighbourhood.

SYSTOLIC ALGORITHMS FOR CONVOLUTION AND MEDIAN SMOOTHING

R.R. n° 295

Y. ROBERT

M. TCHUENTE

Mars 1982

I - INTRODUCTION

The development of very large scale integration (VLSI) technology has revolutionized computer design ; the number and complexity of components that can fit on a chip, has been doubling every one-to-two years. It is now possible to produce relatively inexpensive, highly parallel special purpose computing devices, for the implementation of computationnaly demanding operations. For these reasons, it is now necessary to reconsider algorithms, in order to give them a highly parallel structure which may be adapted to new technologies.

In this paper, we present systolic arrays, i.e. networks consisting essentially of only one type of simple cells which rhythmically compute and pass data through the system, and which are mesh interconnected in a regular way. This model was introduced by H.T.KUNG [4] and it has already been applied to special purpose designs in many application areas [2,3,5,6]. In such networks, I/O constraints are greatly reduced because multiple computations are performed per I/O access ; furthermore, high performance can be obtained by extensive concurrent and pipelined use of the cells in the array.

The first step in designing a systolic algorithm for the computation of a function

$$x = (x_1, \dots, x_m) \mapsto (f_1(x), \dots, f_n(x)),$$

consists of putting any f_i into the form

$$f_i(x) = b_i + \sum_{(j_1, \dots, j_r) \in V_i} g_{i,j_1, \dots, j_r}(x_{j_1}, x_{j_2}, \dots, x_{j_r})$$

one can then look for a network where each cell has $r+1$ input-lines and $r+1$ output-lines.

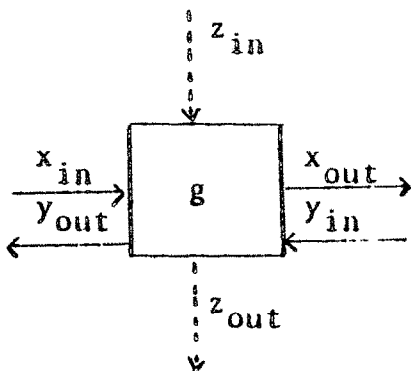
- every data-variable x_j travels from cell to cell without changing its value.
- every result-variable y_i is initialized at b_i ; it travels from cell to cell and, when it meets a r -tuple $(x_{j_1}, x_{j_2}, \dots, x_{j_r}), (j_1, j_2, \dots, j_r) \in V_i$, it performs the partial accumulation.

$$y_i := y_i + g_{i,j_1, \dots, j_r}(x_{j_1}, \dots, x_{j_r})$$

In the sequel, we shall make use of essentially two types of basic cells



$$\begin{aligned} x_{out} &\leftarrow x_{in} \\ y_{out} &\leftarrow f(x_{in}, y_{in}) \end{aligned}$$



$$\begin{aligned} x_{out} &\leftarrow x_{in} \\ y_{out} &\leftarrow y_{in} \\ z_{out} &\leftarrow g(x_{in}, y_{in}, z_{in}) \end{aligned}$$

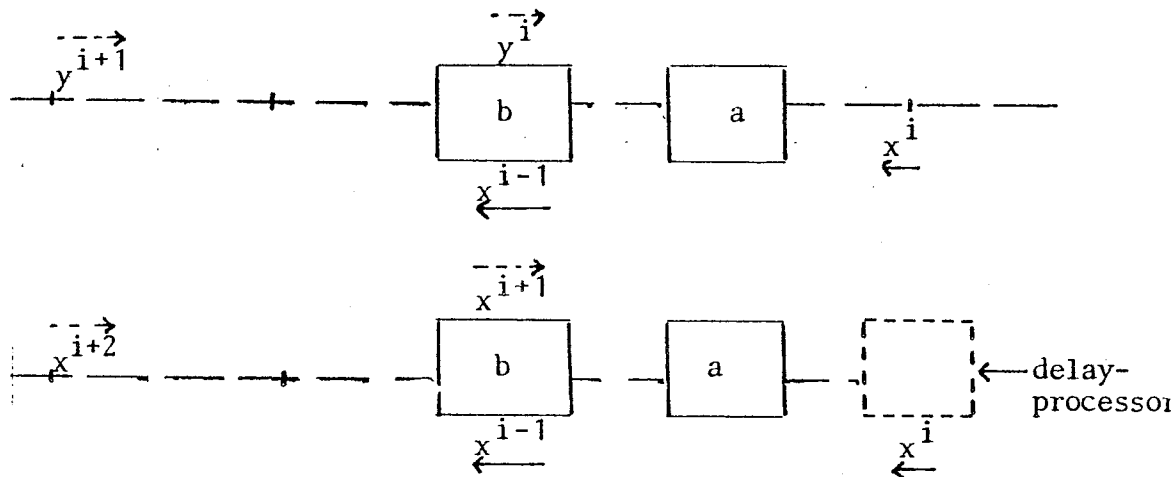
If a network of k cells solves a problem in time t_k , we say that its efficiency is $e = \frac{t_1}{kt_k}$ where t_1 is the time necessary to solve it using a single cell. Clearly, $e \leq 1$ and a parallel algorithm is of maximum efficiency if $e = 1$.

II 2D CONVOLUTION FILTERING .

The algorithms given in H.T.KUNG [4] for the evaluation of one-dimensional real sequences

$$y^i = ax^i + bx^{i-1}$$

or recurrences $x^i = ax^{i-1} + bx^{i-2}$ are pipelined in the following linear arrays :

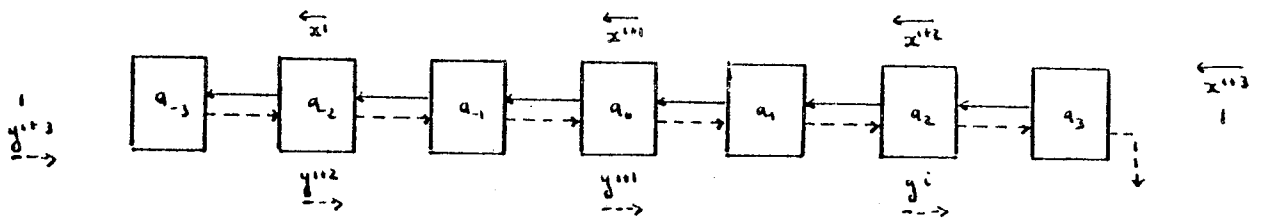


Both algorithms are of efficiency $e = 1/2$. Clearly, 1D convolution (computation of $y^i = a_{-k}x^{i+k} + \dots + a_kx^{i-k}$) and 1D recursive convolution (computation of $y^i = a_{-k}x^{i+k} + \dots + a_kx^{i-k} + b_1y^{i-1} + \dots + b_hy^{i-h}$) can be modeled with these arrays , as shown in figure 1 .

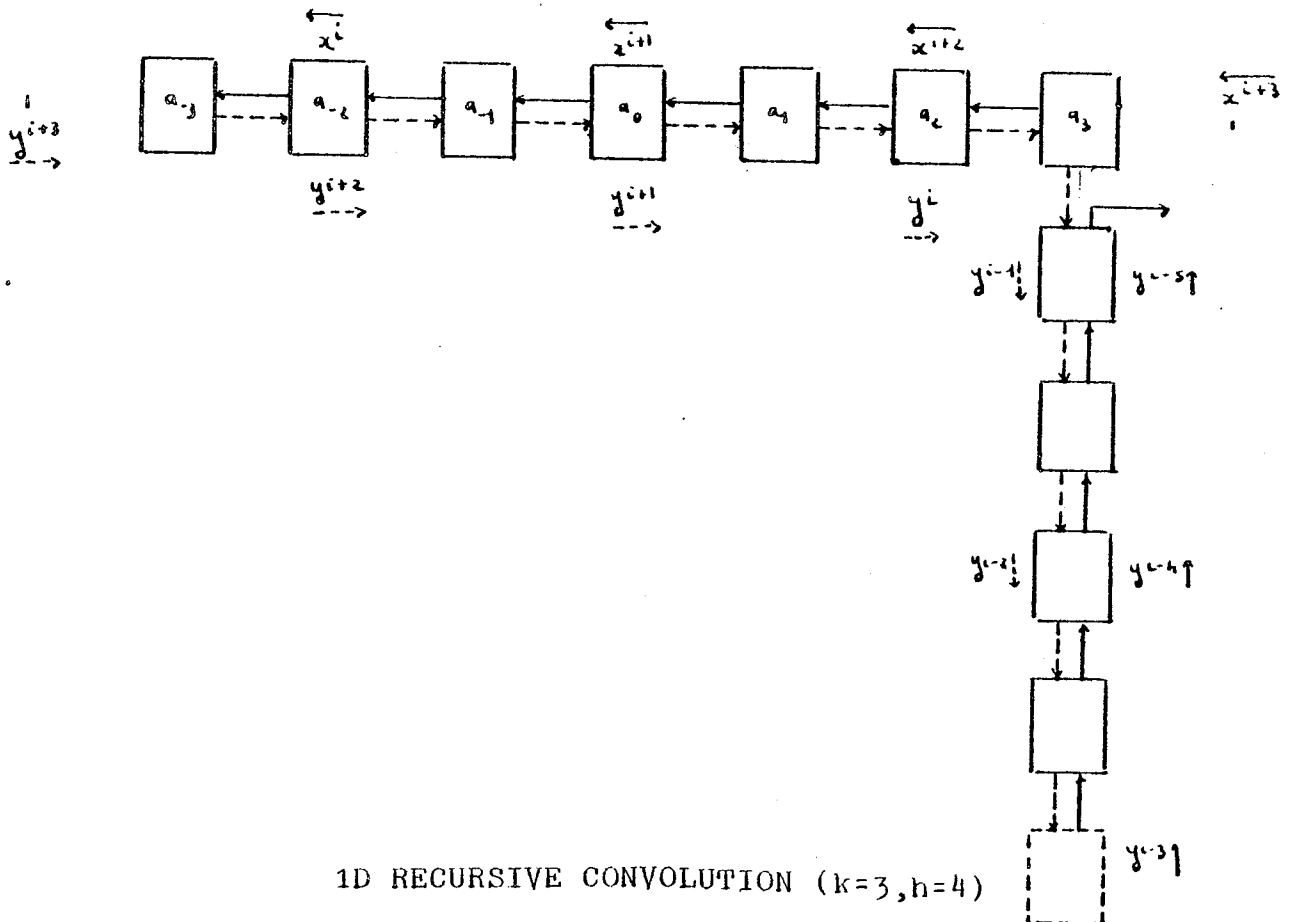
2-1 2D CONVOLUTION

This problem has been solved by H.T.KUNG and S.W.SONG [5] . We give here a simpler solution, using only the basic cells defined in the introduction .

Let us consider n infinite rows of data and a $k \times k$ window with weighting coefficients w_{ij} , $1 \leq i, j \leq k$. We use $k = 3$ for purpose of illustration (see figure 2) .



1D CONVOLUTION ($k=3$)



1D RECURSIVE CONVOLUTION ($k=3, h=4$)

Figure 1

w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}

3x3 window

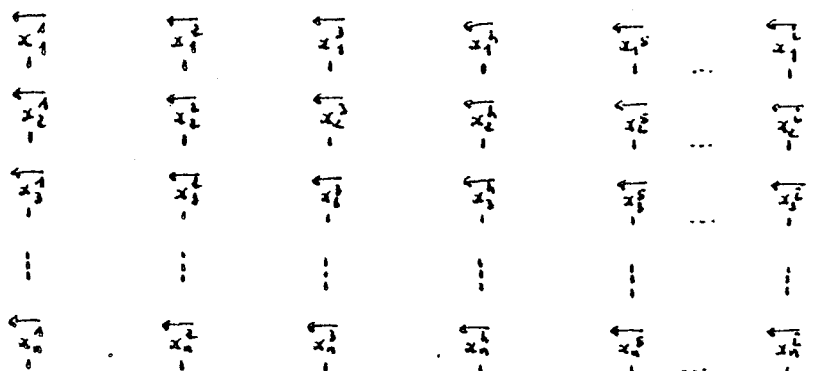


Figure 2

The problem is to compute

$\sum_{i=1}^3 \sum_{j=1}^3 w_{ij} x_{r+i-2}^{s+j-2}$ where $1 \leq r \leq n$ and $s \geq 0$.
Our basic idea is to compute on each row i the three partial weighted sums

$$\sum_{k=1}^3 w_{1,k} x_i^{j-2+k}, \sum_{k=1}^3 w_{2,k} x_i^{j-2+k}, \sum_{k=1}^3 w_{3,k} x_i^{j-2+k}$$

for any three consecutive values x_i^{j-1} , x_i^j and x_i^{j+1}

(since each row, except the first and the last one, appears in three windows); then we add the corresponding partial sums. As illustrated in figure 3, we obtain a regular network of efficiency $e = 1/2$.

2-2) GENERALIZATION

We can notice that our network remains valid if the weighting coefficients w_{ij} vary from row to row. Indeed, changing our notations and setting

$$x^j = \begin{pmatrix} x_1^j \\ \vdots \\ x_n^j \end{pmatrix}$$

We have to compute $y^j = A^{(1)} x^{j-1} + A^{(2)} x^j + A^{(3)} x^{j+1}$ where

$$A^{(k)} = \begin{pmatrix} w_{2,k} & w_{3,k} & & \\ & w_{1,k} & & \\ & & w_{3,k} & \\ & & & w_{1,k} & w_{2,k} \end{pmatrix}$$

2D CONVOLUTION

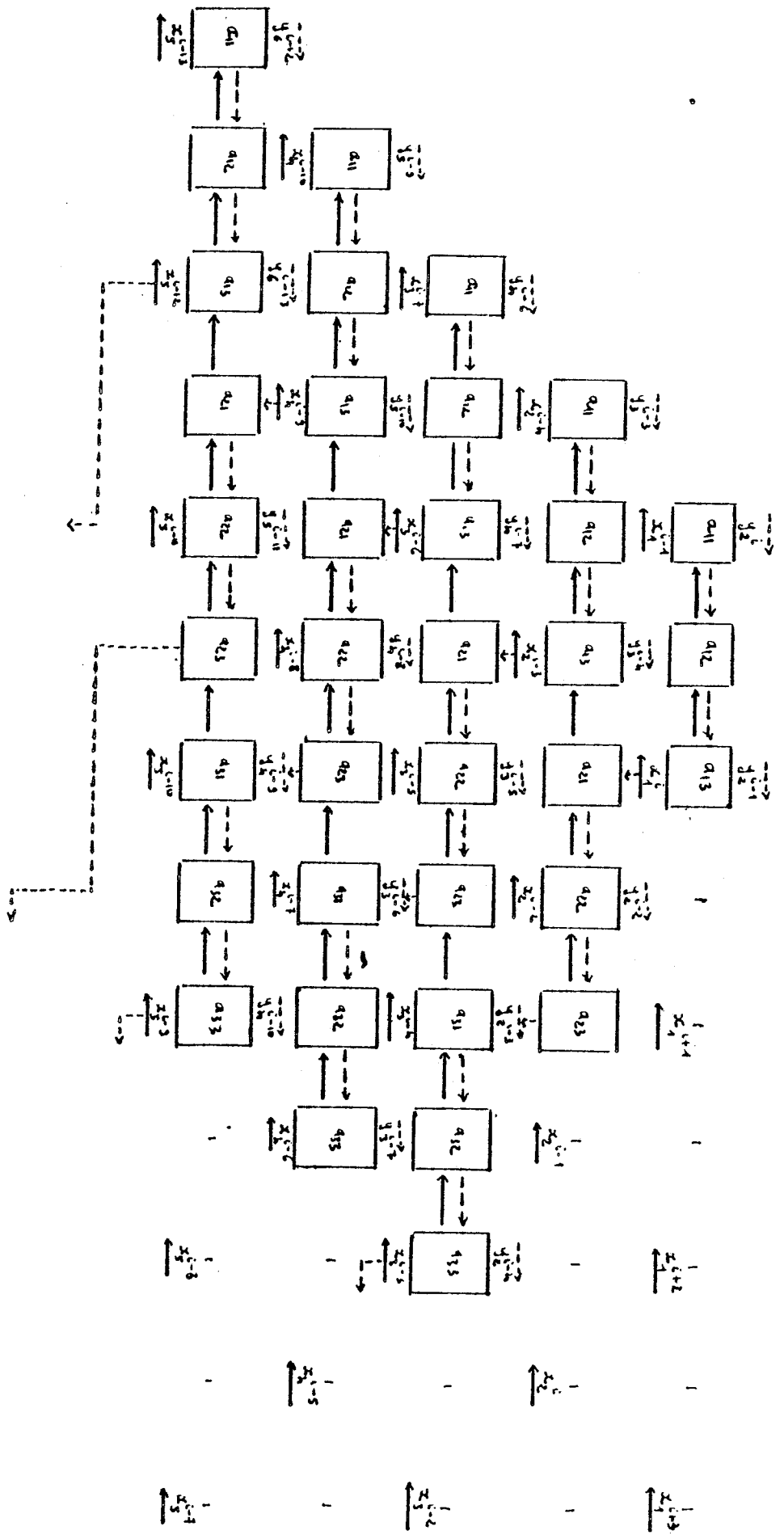


Figure 3

This is a special case of the computation of general n-dimensional sequences

$$Y^j = A^{(p-1)}X^{j+p-1} + \dots + A^{(1)}X^{j+1} + A^{(0)}X^j + b$$

studied in [7] (the $A^{(k)}$, $0 \leq k \leq p-1$ are $n \times n$ matrices and b is a vector of order n) .

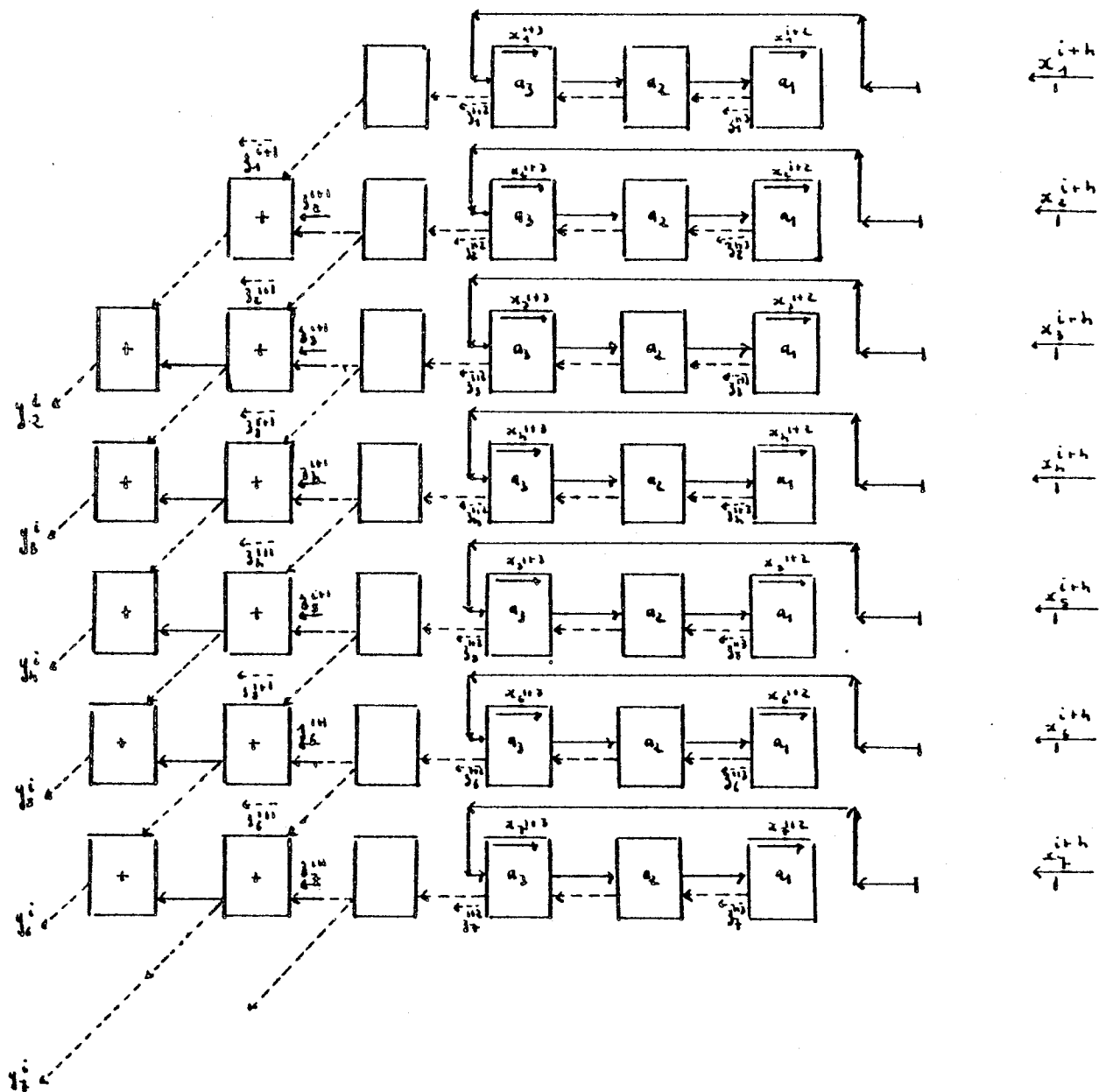
REMARK .

If the weighting coefficients do not depend on the row, i.e. $w_{ij} = a_j$, then we are lead to the cpmputation of

$$Y^j = M(a_1 X^{j-1} + a_2 X^j + a_3 X^{j+1}) \quad \text{where}$$

$$M = \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ & & & & 1 \end{pmatrix}$$

We then obtain the simpler network of figure 4 .



$$y_j^i = z_{j-1}^i + z_j^i + z_{j+1}^i$$

Figure 4

III - MEDIAN SMOOTHING ON THE VON NEUMANN NEIGHBOURHOOD

Median smoothing is a filtering technique widely used in digital signal and image processing, which consists of replacing each sample in a grid, by the median of the

samples within some local neighbourhood. As pointed out in [1], median filtering, contrary to convolution, does not introduce intermediate values not found in the original pattern.

This problem has already been solved by A.L.FISHER [1] but the basic cells of the array were rather complicated. Here we propose algorithms with simpler cells, for the particular cases where the median is taken over three or five values.

Clearly, our algorithm can be applied to the median smoothing of a rectangular pattern, using the VON NEUMANN neighbourhood.

```

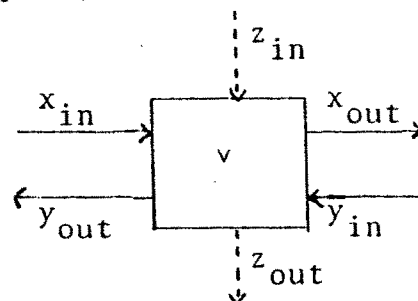
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x
    
```

Our idea consists of expressing the median of $2k-1$ values $a_1, a_2, \dots, a_{2k-1}$, by the well known formula

$$i_1 < i_2 < \dots < i_k \quad (a_{i_1} \wedge a_{i_2} \wedge \dots \wedge a_{i_k})$$

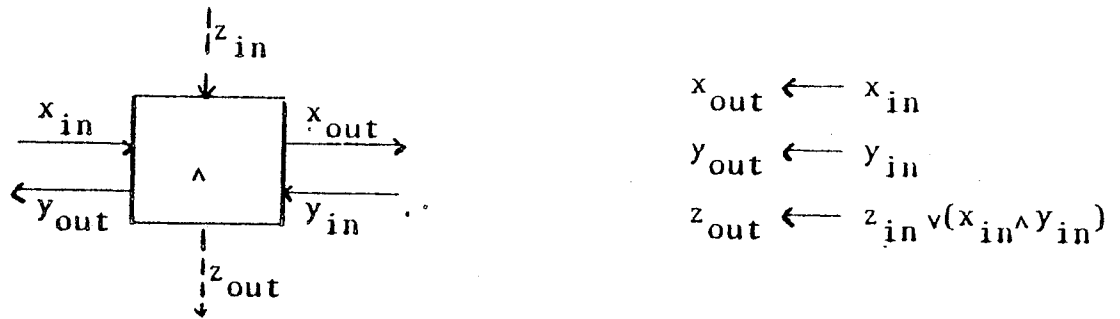
where $\wedge$ (resp $\vee$) is the minimum (resp maximum) operation.

The basic cells used here, have three inputs, three outputs, and are defined as follows :



$$\begin{aligned} x_{out} &\leftarrow x_{in} \\ y_{out} &\leftarrow y_{in} \\ z_{out} &\leftarrow x_{in} \vee y_{in} \vee z_{in} \end{aligned}$$

In the figures, a missing x_{in}, y_{in} or z_{in} is supposed to be constant and equal to $-\infty$.



In the figures, a missing x_{in} , y_{in} , or z_{in} is supposed to be constant and equal respectively to $+\infty$, $+\infty$, and $-\infty$

The networks are presented in figure 5 and figure 6.

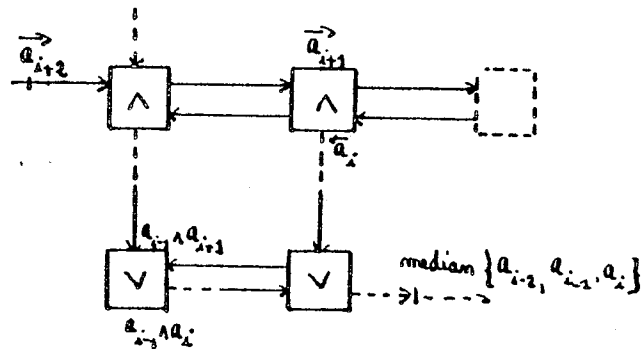


Figure 5

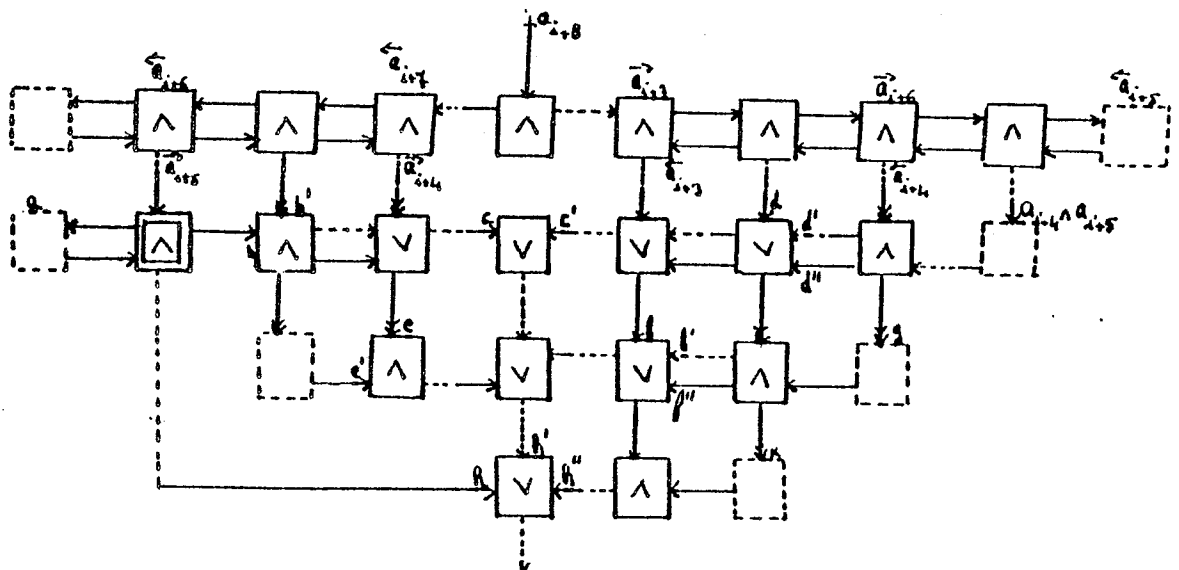
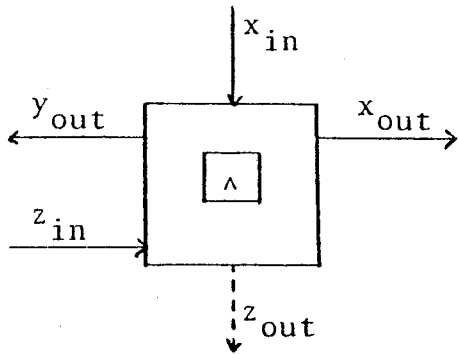


Figure 6

The cell $\boxed{\boxed{\wedge}}$ is special :



$$\begin{aligned} x_{out} &\leftarrow x_{in} \\ y_{out} &\leftarrow x_{in} \\ z_{out} &\leftarrow x_{in} \wedge z_{in} \end{aligned}$$

The symbols $a, b, b', \dots$ of the figure, are defined as follows :

$$a = a_{i+4} \wedge a_{i+5}$$

$$b = a_{i+4} \wedge a_{i+5}$$

$$b' = a_{i+4} \wedge a_{i+6}$$

$$c = (a_{i+3} \wedge a_{i+4} \wedge a_{i+5}) \wedge (a_{i+3} \wedge a_{i+4} \wedge a_{i+6})$$

$$c' = (a_{i+2} \wedge a_{i+3} \wedge a_{i+4}) \wedge (a_{i+2} \wedge a_{i+3} \wedge a_{i+5}) \wedge (a_{i+2} \wedge a_{i+3} \wedge a_{i+6})$$

$$d = a_{i+3} \wedge a_{i+6}$$

$$d' = a_{i+3} \wedge a_{i+4} \wedge a_{i+5}$$

$$d'' = a_{i+3} \wedge a_{i+6}$$

$$e = a_{i+3} \wedge a_{i+6}$$

$$e' = a_{i+3} \wedge a_{i+5}$$

$$f = a_{i+2} \wedge a_{i+6}$$

$$f' = a_{i+2} \wedge a_{i+4} \wedge a_{i+5}$$

$$f'' = a_{i+2} \wedge a_{i+4}$$

$$h = a_{i+3} \wedge a_{i+4} \wedge a_{i+5}$$

$$h' = (a_{i+1} \wedge a_{i+2} \wedge a_{i+3}) \wedge (a_{i+1} \wedge a_{i+2} \wedge a_{i+4}) \wedge (a_{i+1} \wedge a_{i+2} \wedge a_{i+5}) \wedge (a_{i+1} \wedge a_{i+3} \wedge a_{i+4}) \wedge (a_{i+1} \wedge a_{i+3} \wedge a_{i+5})$$

$$h'' = a_{i+1} \wedge a_{i+4} \wedge a_{i+5}$$

$$k = a_{i+2} \wedge a_{i+5}$$

REMARK

We can derive from the two preceeding networks , a recursive computation for the median taken over 7, 9 or more values . Setting for any positive integers n,k

$$F_n(a_1, \dots, a_k) = \bigvee_{i_1 < i_2 < \dots < i_n} a_{i_1} \wedge a_{i_2} \wedge \dots \wedge a_{i_n}$$

which we call the n-function of order k
we can clearly express any n-function of order k+1 as a first degree polynomial in the m-funtions of order k with $m < n$.
For instance,

$$\begin{aligned} F_4(a_1, \dots, a_7) = & F_4(a_1, \dots, a_4) \vee (F_3(a_1, \dots, a_4) \wedge F_1(a_5, \dots, a_7)) \\ & \vee (F_2(a_1, \dots, a_4) \wedge F_2(a_5, \dots, a_7)) \\ & \vee (F_1(a_1, \dots, a_4) \wedge F_3(a_5, \dots, a_7)) \end{aligned}$$

REFERENCES

- [1] FISHER A.L.
Systolic algorithms for running order statistics in
signal and image processing.
Technical Report, Carnegie-Mellon University, 1981
- [2] FOSTER M.J. and KUNG H.T.
Recognize regular languages with programmable building
blocks
in Proceedings of international VLSI conference,
J.P.Gray Ed. Academic Press, August, 1981 p.75-84
- [3] KUNG H.T. and LEISERSON C.E.
Systolic arrays for VLSI
in Introduction to VLSI Systems, C.Mead and L.Conway Ed,
Addison Wesley 1980, p.271-292
- [4] KUNG H.T.
The structure of parallel algorithms
Advances in computers Vol. 19 (1980) p.65-112
- [5] KUNG H.T. and SONG S.W.
A systolic 2.D convolution chip
Technical report, Carnegie-Mellon University, 1981
- [6] PREPARATA F.P. and VUILLEMIN J.
Area-time optimal VLSI networks for multiplying matrices
Information Processing Letters Vol.11 (1980)n°2 p.77-80
- [7] ROBERT Y. and TCHUENTE M.
Organisation du calcul en parallèle de récurrences
vectorielles
Rapport de Recherche I.M.A.G. n°261, Grenoble 1981

ANNEXE 3

TITRE : " Regular incomplete factorizations of real
positive definite matrices "

On donne ici un cadre théorique pour l'étude de la
méthode de factorisation incomplète; l'algorithme
correspondant pour les matrices-bandes a été présenté
au sous-chapitre 3.1 .

Regular Incomplete Factorizations of Real Positive Definite Matrices

Yves Robert
Laboratoire IMAG
B.P. 53 X
38041 Grenoble Cédex, France

Submitted by Richard A. Brualdi

ABSTRACT

We study the classes of matrices which admit a regular incomplete factorization with respect to any graph set. We extend the construction to the class of real positive definite matrices. The convergence of the basic iteration associated with the splittings we have is discussed.

1. INTRODUCTION

NOTATION. Throughout this paper, L (U) is a lower (upper) triangular matrix with unit diagonal, and D is a nonsingular diagonal matrix.

1.1.

Suppose we want to solve the matrix equation $Ax = b$, where A is a large sparse $n \times n$ matrix. Choosing a Gaussian elimination would lead us to a factorization $A = LDU$. But the limitations of the method are the fill-in and the resulting high cost in terms of storage. To avoid these difficulties we may consider an incomplete factorization technique to obtain a splitting $A = LDU - R$, where the fill-in has been controlled by zeroing previously determined coefficients in L and U during the elimination (the numerical power of the method is shown in [1-5]).

1.2.

Let G be a set of pairs of integers (i, j) , with $1 \leq i, j \leq n$ and $i \neq j$. We denote by $\mathcal{S}_n$ [9] the collection of all such sets G .

LINEAR ALGEBRA AND ITS APPLICATIONS

© Elsevier Science Publishing Co., Inc., 1982
52 Vanderbilt Ave., New York, NY 10017

0024-3795/82/040

YVES ROBERT

EXAMPLE. The simplest choice for G is $G = \{(i, j) | i \neq j \text{ and } a_{ij} \neq 0\}$.

DEFINITION. A admits a regular incomplete factorization with respect to G if there exist four matrices L , D , U , and R such that $A = LDU - R$, with

$$l_{ij} = u_{ij} = 0 \quad \text{if } (i, j) \notin G, \quad i \neq j,$$

$$r_{ij} = 0 \quad \text{if } (i, j) \in G \text{ or } i = j.$$

We denote by $\mathcal{G}_n$ the set of $n \times n$ matrices which admit a regular incomplete factorization with respect to any $G \in \mathcal{S}_n$. An element of $\mathcal{G}_n$ is concisely said to be IF-regular.

1.3.

In [5] Meijerink and Van der Vorst show that any M -matrix is IF-regular. In [4] Manteuffel extends this result for the H -matrices; he then constructs a regular incomplete factorization for symmetric positive definite matrices. The basic idea is to let A be symmetric and positive definite and write $A = D - B$, where D is the diagonal of A ; then

$$A(\alpha) = D - \frac{1}{1 + \alpha} B$$

is diagonally dominant (and thus an H -matrix) for α large enough.

1.4.

First we give conditions for a matrix to be IF-regular (these conditions lead us to a new proof of Manteuffel's result). Then we extend the construction to the class of real positive definite matrices (not necessarily symmetric). Finally we study the convergence of the basic iteration associated with the splittings we have.

2. CLASSES OF IF-REGULAR MATRICES

2.1.

Let $A \in \mathbb{C}^{n,n}$ admit a regular incomplete factorization with respect to G :

$$A = LDU - R.$$

REGULAR INCOMPLETE FACTORIZATIONS

The following relations hold [4] for $1 \leq i \leq n$, $i+1 \leq j \leq n$:

$$\begin{aligned} d_i &= a_{ii} - \sum_{k=1}^{i-1} l_{ik} d_k u_{ki}, \\ l_j d_i &= a_{ji} - \sum_{k=1}^{i-1} l_{jk} d_k u_{ki} \quad \text{if } (j, i) \in G, \\ l_j &= 0 \quad \text{if } (j, i) \notin G \\ u_{ij} d_i &= a_{ij} - \sum_{k=1}^{i-1} l_{ik} d_k u_{kj} \quad \text{if } (i, j) \in G, \\ u_{ij} &= 0 \quad \text{if } (i, j) \notin G \end{aligned} \quad (*)$$

For $1 \leq i \leq n$ we define the scheme

$$(H_i) = \text{computation of } \begin{cases} d_i, \\ l_j \text{ and } u_{ij} \end{cases} \text{ with } i+1 \leq j \leq n.$$

The decomposition exists iff [9] the coefficients d_i , $1 \leq i \leq n$, in (*) recursively defined by the scheme (H_i) , $1 \leq i \leq n$, are all nonzero. Using this scheme, we make two remarks:

- (1) If the factorization with respect to G exists, then it is unique.
- (2) If A is symmetric, and if we add the condition $(i, j) \in G \Rightarrow (j, i) \in G$, then $U = L^t$, and $R = LDL^t - A$ is symmetric too.

2.2.

NOTATION. $A \in \mathbb{C}^{n \times n}$ is written

$$A = \begin{pmatrix} a_{11} & b^t \\ c & E \end{pmatrix}, \quad b, c \in \mathbb{C}^{n-1}, \quad E \in \mathbb{C}^{(n-1) \times (n-1)}.$$

We define

$$A' = E - \frac{1}{a_{11}} cb^t;$$

YVES ROBERT

A' is the "Gaussian transform" of A , for one step of Gaussian elimination gives

$$A = \begin{pmatrix} 1 & 0 \\ c/a_{11} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & 0 \\ 0 & A' \end{pmatrix} \begin{pmatrix} 1 & b'/a_{11} \\ 0 & I_{n-1} \end{pmatrix}.$$

THEOREM. Let ω be a class of matrices (of any order) satisfying a given property, and such that

(T1) $A \in \omega \Rightarrow$ the diagonal D of A is nonsingular;

(T2) if $A \in \omega$ and B is obtained from A by zeroing some nondiagonal coefficients, then $B \in \omega$;

(T3) $A \in \omega \Rightarrow A' \in \omega$.

Then any matrix of ω is IF-regular.

Proof. Let us consider $A \in \omega$ of order n and $G \in \mathcal{S}_n$. We set

$$\begin{aligned} A = A_1 &= \begin{pmatrix} a_{11}^{(1)} & b_1^t \\ c_1 & E_1 \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{(1)} & \beta_1^t \\ \gamma_1 & E_1 \end{pmatrix} - \begin{pmatrix} 0 & r_1^t \\ s_1 & 0 \end{pmatrix} = \tilde{A}_1 - R_1 \end{aligned}$$

where β_1 [γ_1] is obtained from b_1 [c_1] by zeroing a_{1j} [a_{j1}] if $(1, j) \notin G$ [$(j, 1) \notin G$]. (T1) and (T2) give $a_{11}^{(1)} \neq 0$ and $\tilde{A}_1 \in \omega$: one step of Gaussian elimination on $\tilde{A}_1$ gives

$$\begin{aligned} \tilde{A}_1 &= \begin{pmatrix} 1 & 0 \\ \frac{1}{a_{11}^{(1)}} \gamma_1 & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11}^{(1)} & 0 \\ 0 & \tilde{A}'_1 \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{a_{11}^{(1)}} \beta_1^t \\ 0 & I_{n-1} \end{pmatrix} \\ &= L_1 \begin{pmatrix} a_{11}^{(1)} & 0 \\ 0 & A_2 \end{pmatrix} U_1. \end{aligned}$$

From (T3), $A_2 \in \omega$, so we decompose A_2 as

$$A_2 = L_2 \begin{pmatrix} a_{22}^{(2)} & 0 \\ 0 & A_3 \end{pmatrix} U_2 - R_2.$$

REGULAR INCOMPLETE FACTORIZATIONS

Filling out matrices of order $n - 1$, we get

$$A = L_1 L_2 \begin{pmatrix} a_{11}^{(1)} & & \\ & a_{22}^{(2)} & \\ & & A_3 \end{pmatrix} U_2 U_1 - (R_1 + R_2)$$

(obviously $L_1 R_2 U_1 = R_2$), and $A_3 \in \omega$.

By iteration we finally have $A = LDU - R$, $L = L_1 \cdots L_n$, $U = U_n \cdots U_1$, $D = \text{diag}(a_{ii}^{(i)})$ and $R = R_1 + \cdots + R_n$; and A is IF-regular. ■

2.3. Diagonally dominant matrices

PROPOSITION [4]. Any diagonally dominant matrix A ($|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$, $1 \leq i \leq n$) is IF-regular.

Proof. Let ω be the class of diagonally dominant matrices. (T1) and (T2) are obvious. Let A be diagonally dominant of order n ; the coefficients of its Gaussian transform A' are given by

$$a'_{ij} = a_{ij} - \frac{1}{a_{ii}} a_{ii} a_{ij}, \quad 2 \leq i, j \leq n.$$

Fixing $i \geq 2$, we have

$$\begin{aligned} \sum_{\substack{j=2 \\ i \neq j}}^n |a'_{ij}| &= \sum_{\substack{j=2 \\ i \neq j}}^n \left| a_{ij} - \frac{a_{ii} a_{ij}}{a_{ii}} \right| \\ &\leq \sum_{\substack{j=2 \\ i \neq j}}^n |a_{ij}| + \left| \frac{a_{ii}}{a_{ii}} \right| \sum_{\substack{j=2 \\ i \neq j}}^n |a_{ij}|; \end{aligned}$$

thus

$$\begin{aligned} \sum_{\substack{j=2 \\ i \neq j}}^n |a'_{ij}| &< (|a_{ii}| - |a_{ii}|) + \left| \frac{a_{ii}}{a_{ii}} \right| (|a_{ii}| - |a_{ii}|) \\ &= |a_{ii}| - \left| \frac{a_{ii} a_{ii}}{a_{ii}} \right|. \end{aligned}$$

YVES ROBERT

or

$$\begin{aligned} \left| a_{ii} - \frac{a_{i1}a_{1i}}{a_{11}} \right| &= \left| a_{ii} - \frac{a_{i1}a_{1i}}{a_{11}} \right| \\ &\leq \left| a_{ii} - \frac{a_{i1}a_{1i}}{a_{11}} \right| = |a'_{ii}|. \end{aligned}$$

So A' is diagonally dominant, $A' \in \omega$, and (T3) is verified. ■

2.4. H -matrices

PROPOSITION. Any H -matrix A (i.e., any A such that B defined by $b_{ii} = |a_{ii}|$ and $b_{ij} = -|a_{ij}|$, $i \neq j$, is an M -matrix) is IF-regular.

Proof. Let ω be the class of H -matrices. (T1) and (T2) follow from usual properties of M -matrices (see [6] for instance). For (T3) we use the following characterization [6]:

(*) A is an H -matrix iff there exists a real nonsingular diagonal matrix Δ such that $A\Delta$ is diagonally dominant.

Setting

$$\Delta = \begin{pmatrix} d_1 & & \\ & \Delta_2 & \\ & & \end{pmatrix}, \quad d_1 \in \mathbb{R}, \quad \Delta_2 \text{ of order } n-1,$$

the Gaussian transformation of $A\Delta$ is $(A\Delta)' = A'\Delta_2$. From the proposition in Section 2.3 and the characterization (*) we see that A' is an H -matrix, thereby establishing (T3). ■

3. EXTENSION TO REAL POSITIVE DEFINITE MATRICES

3.1.

DEFINITION. We denote by $\mathcal{D}^+$ the class of real positive matrices. $A \in \mathcal{D}^+$ iff $(x, Ax) > 0$ for any $x \in \mathbb{R}^n \setminus \{0\}$, where n is the order of A .

REGULAR INCOMPLETE FACTORIZATIONS

REMARKS (7).

- (i) $A \in \mathcal{D}^+ \Rightarrow$ all the principal minors of A are > 0 ; the converse is false.
- (ii) If A is symmetric, then $A \in \mathcal{D}^+ \Leftrightarrow$ all the principal minors of A are > 0 .

LEMMA. $A \in \mathcal{D}^+ \Rightarrow A' \in \mathcal{D}^+$, where A' is the Gaussian transform of A .

Proof. Let n be the order of $A \in \mathcal{D}^+$. We write

$$A = \begin{pmatrix} a_{11} & b' \\ c & E \end{pmatrix}, \quad A' = E - \frac{1}{a_{11}} cb'.$$

For any $X \in \mathbb{R}^{n-1} \setminus \{0\}$ and any real x ,

$$(x, X') A \begin{pmatrix} x \\ X \end{pmatrix} > 0,$$

or equivalently, $x(a_{11}x + b'X) + X'(cx + EX) > 0$. Choosing $x = -(1/a_{11})b'X$, we get

$$X' \left(E - \frac{1}{a_{11}} cb' \right) X > 0,$$

thus $A' \in \mathcal{D}^+$. ■

The lemma shows (T3) is true for $\mathcal{D}^+$. (T1) is clearly verified. But the choice of

$$A = \begin{pmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{pmatrix} \quad \text{and} \quad G = \{(1,3), (3,1)\}$$

provides a counterexample to (T2) ($A \in \mathcal{D}^+$ —use the remark (ii)—but the determinant of

$$B = \begin{pmatrix} 4 & 0 & 3 \\ 0 & 4 & 3 \\ 3 & 3 & 4 \end{pmatrix}$$

is -8).

YVES ROBERT

3.2.

We extend the definition of an incomplete factorization $A = LDU - R$ by suppressing the condition $\text{diag } R = 0$. We then have the following theorem:

THEOREM. *Let $A \in \mathcal{D}^+$ of order n and $G \in \mathcal{S}_n$. Then A admits a regular incomplete factorization with respect to G : $A = LDU - R$. Furthermore, $D > 0$ and R is semipositive definite ($(x, Rx) \geq 0$ for any $x \in \mathbb{R}^n$).*

Proof. Divided into four parts.

First part. A 2×2 matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is semipositive definite iff $a \geq 0$, $d \geq 0$, $4ad - (b+c)^2 \geq 0$, since

$$\begin{aligned} (x_1, x_2) \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= ax_1^2 + (b+c)x_1x_2 + dx_2^2 \\ &= a \left(x_1 + \frac{b+c}{2a}x_2 \right)^2 + \left(d - \frac{(b+c)^2}{4a} \right) x_2^2 \quad \text{if } a \neq 0 \end{aligned}$$

(and if $a = 0$, the result is obvious).

Second part: construction of R_1 . We define R_1 by setting

$$r_{i1} = -a_{i1} \quad \text{if } (i, 1) \notin G \quad i \neq 1,$$

$$r_{1j} = -a_{1j} \quad \text{if } (1, j) \notin G \quad j \neq 1;$$

$$r_{i1} = 0 \quad \text{if } (i, 1) \in G$$

$$r_{1j} = 0 \quad \text{if } (1, j) \in G$$

$$r_{11} = \sum_{2 \leq j \leq n} \frac{1}{2} (|r_{1j}| + |r_{j1}|),$$

$$r_{jj} = \frac{1}{2} (|r_{j1}| + |r_{1j}|), \quad j \neq 1;$$

REGULAR INCOMPLETE FACTORIZATIONS

and all the other coefficients of R_1 equal to zero. The configuration of R_1 is

$$\begin{pmatrix} x & x & x & x \\ x & x & & \\ x & & x & \\ x & & & x \end{pmatrix}.$$

R_1 may be viewed as the sum of 2×2 matrices $\begin{pmatrix} \frac{1}{2}(|r_{1i}| + |r_{i1}|) & r_{ii} \\ r_{ii} & \frac{1}{2}(|r_{1i}| + |r_{i1}|) \end{pmatrix}$.

The first part shows R_1 is semipositive definite.

Third part: First step of the elimination process: $A + R_1 \in \mathcal{D}^+$. One step of Gaussian elimination gives

$$A + R_1 = L_1 \begin{pmatrix} a_{11} + r_{11} & 0 \\ 0 & A_2 \end{pmatrix} U_1$$

with $a_{11} + r_{11} > 0$ and $A_2 \in \mathcal{D}^+$ (from the lemma in Section 3.1).

Fourth part: Iteration of the process. The same iteration as in the theorem of Section 2.2 leads to $A + R = LDU$, where $D > 0$, and $R = R_1 + \dots + R_n$ is semipositive definite, which is the required incomplete factorization of A . $\blacksquare$

REMARK. This factorization is clearly not unique: we may compute the diagonal of R by setting $r_{ii} = \alpha_i(|r_{i1}| + |r_{1i}|)$ and $r_{ij} = \beta_i(|r_{i1}| + |r_{1i}|)$, with α_i, β_i real numbers such that $4\alpha_i\beta_i \geq 1$ (remember R must be semipositive definite).

3.3. Process of computation

Let $A = LDU + R$ be the incomplete factorization of the matrix $A \in \mathcal{D}^+$ of order n with respect to some $G \in \mathcal{S}_n$. We use the recursive scheme (H_i) , $1 \leq i \leq n$, of Section 2.1 to compute the coefficients of L , D , and U as follows.

(1) Suppose the coefficients related to (H_k) , $k \leq i-1$, are computed. For $j \geq i+1$ we have

$$a_{ji} - \sum_{k=1}^{i-1} l_{jk} d_k u_{ki} + r_{ji} = l_{ji} d_i, \quad (E)$$

$$a_{ij} - \sum_{k=1}^{i-1} l_{ik} d_k u_{kj} + r_{ij} = u_{ij} d_i. \quad (E')$$

YVES ROBERT

(2) Compute

$$r_{ij} = \begin{cases} \sum_{k=1}^{i-1} l_{ik} d_k u_{kj} - a_{ij} & \text{if } (i, j) \notin G, \\ 0 & \text{otherwise,} \end{cases}$$

$$r_{ji} = \begin{cases} \sum_{k=1}^{i-1} l_{jk} d_k u_{ki} - a_{ji} & \text{if } (j, i) \notin G, \\ 0 & \text{otherwise.} \end{cases}$$

(3) Compute $r_{ii} = \sum_{j=1}^i \frac{1}{2}(|r_{ij}| + |r_{ji}|)$ and $d_i = a_{ii} + r_{ii} - \sum_{k=1}^{i-1} l_{ik} d_k u_{ki}$.

(4) Compute l_{ji} and u_{ij} using respectively (E) and (E') if $(j, i) \in G$ or $(i, j) \in G$

Thus we compute the coefficients through the successive configurations

$$\begin{pmatrix} x & x & x & x & x \\ x & & & & \\ x & & & & \\ x & & & & \\ x & & & & \end{pmatrix}, \begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & & & \\ x & x & & & \\ x & x & & & \end{pmatrix}, \dots$$

REMARK. If A is symmetric, and we add the condition $(i, j) \in G \Rightarrow (j, i) \in G$, then R is symmetric too, and $U = L^t$.

3.4

EXAMPLE.

$$A = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

is symmetric, its eigenvalues are 2, 4, 4, 6, and thus $A \in \mathcal{P}^+$. The choice of

$$G = \{(1, 3), (1, 4), (2, 3), (3, 4), (3, 1), (4, 1), (3, 2), (4, 3)\}$$

REGULAR INCOMPLETE FACTORIZATIONS

leads to the factorization

$$\begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ -\frac{1}{5} & 0 & 1 & \\ 0 & 0 & -\frac{5}{19} & 1 \end{pmatrix} \begin{pmatrix} 5 & & & \\ & 6 & & \\ & & \frac{19}{5} & \\ & & & \frac{80}{19} \end{pmatrix} \begin{pmatrix} 1 & 0 & -\frac{1}{5} & 0 \\ & 1 & 0 & 0 \\ & & 1 & -\frac{5}{19} \\ & & & 1 \end{pmatrix} \\ - \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

The eigenvalues of $A + R$ are $5, 6, (9 \pm \sqrt{5})/2$ ($A + R$ has a condition number 1.7 times as good as A).

4. CONVERGENCE OF THE BASIC ITERATION ASSOCIATED WITH THE INCOMPLETE FACTORIZATIONS

4.1.

Suppose we want to solve the linear system $Ax = b$ by using a splitting $A = M - N$ (A and M are nonsingular). The basic iteration associated with this splitting is defined as follows:

- (1) x^0 is an arbitrary vector.
- (2) x^{i+1} is obtained from x^i by solving $Mx^{i+1} = Nx^i + b$.

It converges to the solution of $Ax = b$ (for any x^0) iff [8] $\rho(M^{-1}N) < 1$ (we set $\rho(X) = \max_i |\lambda_i|$, λ_i the eigenvalues of the matrix X).

4.2. H-matrices

PROPOSITION. Let A be an H-matrix of order n , and $A = LDU - R$ the incomplete factorization of A with respect to $G \in \mathcal{S}_n$. The basic iteration associated with this splitting converges.

Proof. The M-matrix B associated with A (see Section 2.4) admits [5] a regular incomplete factorization with respect to G

$$B = \hat{L}\hat{D}\hat{U} - \hat{R} \quad \text{such that} \quad \rho((\hat{L}\hat{D}\hat{U})^{-1}\hat{R}) < 1.$$

YVES ROBERT

Using the recursive scheme (H_i) , $1 \leq i \leq n$, of Section 2.1, we see easily that $|L| \leq \hat{L}$, $|U| \leq \hat{U}$, $|D| \geq \hat{D}$, and $|R| \leq \hat{R}$. Since $L^{-1} = 2I - L$ and $U^{-1} = 2I - U$, we have $|(LDU)^{-1}R| \leq |(\hat{L}\hat{D}\hat{U})^{-1}\hat{R}|$, and the Perron-Frobenius theorem enables us to conclude that $p((LDU)^{-1}R) < 1$. ■

4.3. Positive definite matrices

PROPOSITION. Let $A \in \mathcal{D}^+$ be symmetric of order n , and $A = LDL' - R$ be the incomplete factorization of A with respect to some $G \in \mathcal{S}_n$. The basic iteration associated with this splitting converges.

Proof. Let λ be an eigenvalue of $(LDL')^{-1}R$, and x an eigenvector associated with λ . λ is real, since LDL' is symmetric positive definite and R is symmetric. Then

$$\begin{aligned} 0 \leq (x, Rx) &= \lambda(x, (A + R)x) \Rightarrow \lambda \geq 0, \\ 0 < (x, Ax) &= (1 - \lambda)(x, (A + R)x) \Rightarrow 1 - \lambda > 0; \end{aligned}$$

therefore we do have $p((LDL')^{-1}R) < 1$. ■

REMARK. If $A \in \mathcal{D}^+$ is nonsymmetric, $A = LDU - R$, the eigenvalues γ of $(LDU)^{-1}R$ may have a modulus greater than 1: the choice of

$$A = \begin{pmatrix} 2 & -10 \\ 10 & 2 \end{pmatrix}$$

($A \in \mathcal{D}^+$ from 3.2) and $G = \emptyset$ gives

$$L = U = I, \quad D = \begin{pmatrix} 12 & 0 \\ 0 & 12 \end{pmatrix}, \quad \text{and} \quad R = \begin{pmatrix} 10 & 10 \\ -10 & 10 \end{pmatrix},$$

so that $p(D^{-1}R) = \frac{5}{6}|1 \pm i| = 5\sqrt{2}/6 > 1$.

4.4. Conclusion

For H -matrices and symmetric positive definite matrices the basic iteration associated with the incomplete factorization converges for *any* graph set. For nonsymmetric positive definite matrices this is not the case; the question of finding *some* graph set leading to convergence remains open.

I am greatly indebted to the referee for his helpful suggestions.

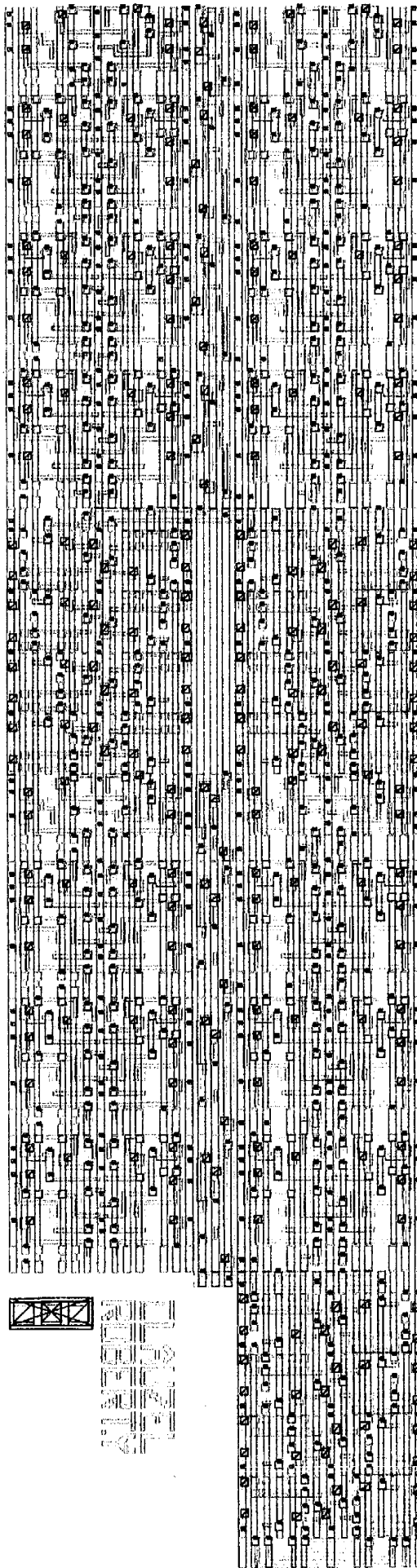
REGULAR INCOMPLETE FACTORIZATIONS

REFERENCES

- 1 R. Glowinski, J. Periaux, and O. Pironneau, An efficient preconditioning scheme for iterative numerical solutions of partial differential equations, *Appl. Math. Modelling* 4, June 1980.
- 2 A. Greenbaum, Comparison of splittings used with the conjugate gradient method, *Numer. Math.* 33:181-194 (1979).
- 3 D. S. Kershaw, The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations, *J. Comput. Phys.* 26:43-65 (1978).
- 4 T. A. Manteuffel, An incomplete factorization technique for positive definite linear systems, *Math. Comp.* 34:473-497 (1980).
- 5 J. A. Meijerink and H. A. Van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix, *Math. Comp.* 31:148-162 (1977).
- 6 F. Robert, *Matrices Nonnégatives et Normes Vectorielles*, Cours de D.E.A., Grenoble, 1973.
- 7 R. Reman, *Cours d'Algèbre Linéaire C3 Analyse Numérique*, Polycopié, Orsay, 1970.
- 8 R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J. 1962.
- 9 R. S. Varga, E. B. Saff, and V. Mehrmann, Incomplete factorizations of matrices and connections with H -matrices, *SIAM J. Numer. Anal.* 17(6):767-793 (1980).

Received 11 March 1991; revised 9 April 1992





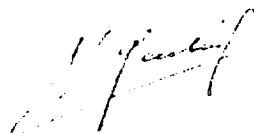
LIBRARY

Dernière page d'une thèse

VU

Grenoble, le 25 Mars 1922

Le Président de la thèse

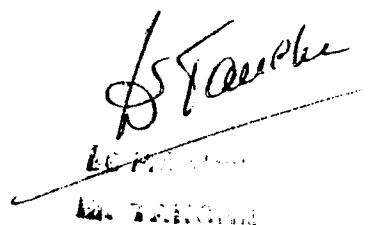
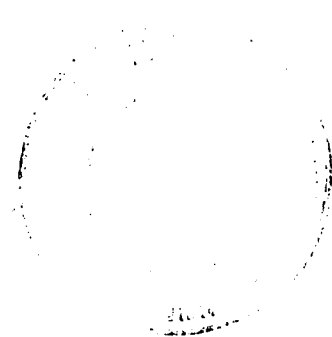


N. GASTINEL

Vu, et permis d'imprimer,

Grenoble, le

Le Président de l'Université Scientifique et Médicale



LE PRÉSIDENT
L. TARDIEU

Résumé :

On présente le modèle systolique introduit par H.T.KUNG pour la conception d'algorithmes VLSI. On propose alors quelques nouveaux algorithmes systoliques pour le calcul scientifique , notamment

- l'évaluation de récurrences vectorielles
- le lissage par médiane
- la factorisation incomplète de matrices-bandes .

Pour l'algorithme de la médiane, on simule son exécution systolique sur micro-ordinateur, et on dessine les masques d'un circuit intégré réalisant cet algorithme .

Mots-clés :

algorithme, systolique, calcul parallèle, VLSI