Étude de la propagation des erreurs de calcul dans deux méthodes classiques de résolution de l'équation de la chaleur

Bernard Liot

N° d'ordre

T H E S E

présentée à la Faculté des Sciences
de l'Université de Grenoble

pour obtenir

le titre de Docteur de Troisième Cycle

"MATHEMATIQUES APPLIQUEES"

par

Bernard LIOT

"ETUDE DE LA PROPAGATION DES ERREURS DE CALCUL DANS DEUX METHODES CLASSIQUES DE RESOLUTION DE L'EQUATION DE LA CHALEUR"

Thèse soutenue le 23 octobre 1964 devant la Commission d'Examen

MM. KUNTZMANN Président
GASTINEL)
VAUQUOIS) Membres

FACULTE DES SCIENCES

L I S T E D E S P R O F E S S E U R S

DOYENS HONORAIRES

M. FORTRAT P.

M. MORET L.

DOYEN

M. WEIL L.

PROFESSEURS TITULAIRES

MM.	NEEL L.	MAGNETISME ET PHYSIQUE DU SOLIDE
	DORIER A.	ZOOLOGIE
	HEILMANN R.	CHIMIE ORGANIQUE
	KRAVTCHENKO J.	MECANIQUE RATIONNELLE
	CHABAUTY C.	CALCUL DIFFERENTIEL ET INTEGRAL
	PARDE M.	POTAMOLOGIE
	BENOIT J.	RADIOELECTRICITE
	CHENE M.	CHIMIE PAPETIERE
	BESSON J.	ELECTROCHIMIE
	WEIL L.	THERMODYNAMIQUE
	FELICI N.	ELECTROSTATIQUE
	KUNTZMANN J.	MATHEMATIQUES APPLIQUEES
	BARBIER R.	GEOLOGIE APPLIQUEE
	SANTON L.	MECANIQUE DES FLUIDES
	OZENDA P.	BOTANIQUE
	FALLOT M.	PHYSIQUE INDUSTRIELLE
	GALVANI O.	MATHEMATIQUES
	MOUSSA A.	CHIMIE NUCLEAIRE
	TRAYNARD P.	CHIMIE
	SOUTIF M.	PHYSIQUE
	CRAYA A.	HYDRODYNAMIQUE
	REULOS R.	THEORIE DES CHAMPS
	AYANT Y.	PHYSIQUE APPROFONDIE
	GALLISSOT F.	MATHEMATIQUES APPLIQUEES
Melle	LUTZ E.	MATHEMATIQUES
MM.	BLAMBERT M.	MATHEMATIQUES
	BOUCHEZ R.	PHYSIQUE NUCLEAIRE
	ILLIBOUTRY L.	GEOPHYSIQUE
	MICHEL R.	GEOLOGIE ET MINERALOGIE
	BONNIER E.	ELECTROCHIMIE
	DESSAUX G.	PHYSIQUE ANIMALE
	PILLET E.	ELECTROCHIMIE
	DEBELMAS J.	GEOLOGIE
	GERBER R.	MATHEMATIQUES
	PAUTHENET R.	ELECTROTECHNIQUE
	VAUQUOIS B.	MATHEMATIQUES APPLIQUEES
	SILBER R.	MECANIQUE DES FLUIDES
	MOUSSIEGT J.	ELECTRONIQUE
	BARBIER J. C.	PHYSIQUE
	KPSZUL J. L.	MATHEMATIQUES
	BUYLE-BODIN M.	ELECTRONIQUE

PROFESSEURS SANS CHAIRE

M.	LACASE A.	THERMODYNAMIQUE
Mme	KOFLER L.	BOTANIQUE
MM.	DREYFUS B.	THERMODYNAMIQUE
	VAILLANT F.	ZOOLOGIE ET HYDROBIOLOGIE
	GIRAUD P.	GEOLOGIE
	GIDON P.	GEOLOGIE ET MINERALOGIE
	ARNAUD P.	CHIMIE
	PERRET R.	SERVOMECANISMES
Mme	LUMER L.	MATHEMATIQUES
Mme	BARBIER M.J.	ELECTROCHIMIE
Mme	SOUTIF J.	PHYSIQUE
MM.	BRISSONNEAU P.	PHYSIQUE
	COHEN J.	ELECTROCHIMIE
	DEPASSEL R.	MECANIQUE
	GASTINEL N.	MATHEMATIQUES APPLIQUEES

PROFESSEURS ASSOCIES

MM.	LUMER G.	MATHEMATIQUES
	HIGUCHI	BIOSYNTHESE DE LA CELLULOSE
	WAGNER	BOTANIQUE

MAITRES DE CONFERENCES

MM.	ROBERT A.	CHIMIE PAPETIERE
	ANGLES D'AURIAC	MECANIQUE DES FLUIDES
	BIAREZ J. P.	MECANIQUE PHYSIQUE
	COUMES A.	ELECTRONIQUE
	DODU J.	MECANIQUE DES FLUIDES
	DUCROS P.	MINERALOGIE ET CRISTALLOGRAPHIE
	CLENAT P.	CHIMIE
	HACQUES G.	CALCUL NUMERIQUE
	LANCIA R.	PHYSIQUE AUTOMATIQUE
	PEBAY-PEROULA	PHYSIQUE
	KAHANE	PHYSIQUE GENERALE
	DOLIQUE	ELECTRONIQUE
Mme	KAHANE J.	PHYSIQUE
MM.	DEGRANGE C.	ZOOLOGIE
	GAGNAIRE D.	CHIMIE PAPETIERE
	RASSAT A.	CHIMIE SYSTEMATIQUE
	KLEIN J.	MATHEMATIQUES
	BETHOUX P.	MATHEMATIQUES APPLIQUEES
	POULOUJADOFF M.	ELECTROTECHNIQUE
	DEPOMMIER P.	PHYSIQUE NUCLEAIRE
	DEPORTES C.	CHIMIE
	BARRA J.	MATHEMATIQUES APPLIQUEES
Mme	BOUCHE L.	MATHEMATIQUES
MM.	PERRIAUX J.	GEOLOGIE
	SARROT-REYNAULD	GEOLOGIE
	CAUQUIS G.	CHIMIE GENERALE
	LABBE A.	BOTANIQUE
	BONNET G.	PHYSIQUE GENERALE
	BARNOUD F.	BIOSYNTHESE DE LA CELLULOSE
Mme	BONNIER M. J.	CHIMIE

MAITRES DE CONFERENCES ASSOCIES

MM.	ISHIKAWA Y.	MAGNETISME
	QUATTROPANI	THERMODYNAMIQUE

Je remercie Monsieur le Professeur KUNTZMANN, Directeur de l'Institut de Mathématiques Appliquées de Grenoble, de me faire l'honneur de présider le Jury de cette thèse.

J'exprime ma profonde reconnaissance à Monsieur le Professeur GASTINEL qui a dirigé ce travail : après en avoir suggéré l'idée, il en a suivi pas à pas l'élaboration et en a permis l'achèvement par ses conseils et ses encouragements variés.

Je remercie Monsieur le Professeur VAUQUOIS de bien vouloi faire partie du Jury de cette thèse.

Je remercie tous les membres du Laboratoire de Calcul de l'Université de Grenoble qui m'ont aidé dans ce travail, particulièrement Monsieur AUROUX pour ses conseils en programmation.

Je remercie Mademoiselle PICCO et Monsieur MOUNET qui ont apporté tous leurs soins à la présentation matérielle.

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

I - Equation de la chaleur et méthodes de différences

L'équation de la chaleur se présente sous la forme générale de l'équation aux dérivées partielles suivante :

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{n} \left(\mu_i \cdot \frac{\partial^2 u}{\partial x_i^2} \right)$$

où $u\ (x_1, x_2, \ldots, x_n, t)$ est la fonction inconnue.
On distingue la variable t qui désigne presque toujours le temps et les variables $x_1, x_2, \ldots x_n$ appelées variables d'espace.

On peut associer à l'équation ci-dessus la condition initiale :

$u\ (x_1, \ldots, x_n, t_0)$ = fonction donnée pour $-\infty < x_i < +\infty$ et $i = 1, 2, \ldots n$; la solution est alors déterminée pour $t > t_0$ et $-\infty < x_i < +\infty$.

On peut aussi se borner à un domaine fini : $a_i < x_i < b_i$; dans ce cas, il faut en outre se donner des "conditions de bord" pour $x_i = a_i$ et $x_i = b_i$; la forme la plus simple de ces conditions de bord consiste à se donner les valeurs $u(a_1, a_2, \ldots, a_n, t)$ et $u\ (b_1, b_2, \ldots, b_n, t)$ de la fonction u sur les "bords" pour $t > t_0$.

Cette équation est la plus habituelle des équations aux dérivées partielles de type parabolique.

Nous étudierons seulement le cas d'une seule variable d'espace :

$$\frac{\partial u}{\partial t} = \mu \cdot \frac{\partial^2 u}{\partial x^2}$$

équation qui se ramène à :

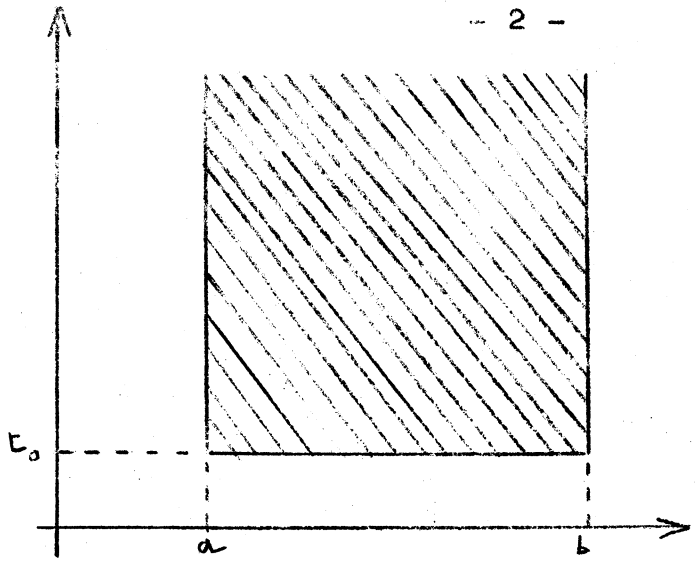
$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

On suppose que sont connus :

$u(x,t_0)$ pour $a < x < b$

$u(a,t)$ pour $t > t_0$

$u(b,t)$ pour $t > t_0$



Cela revient à chercher en tout point intérieur au domaine hachuré sur la figure ci-dessus la fonction $u(x,t)$ qui satisfasse à l'équation (1) et aux conditions (2).

Si les fonctions données : $f(x) = u(x,t_0)$, $g(t) = u(a,t)$ et $h(t) = u(b,t)$ sont suffisamment simples, on peut trouver une solution explicite de ce problème au moyen des séries de Fourier (*)

En pratique, on calcule souvent la solution par des méthodes numériques de discrétisation sans passer par les séries de Fourier.

Le principe de toute une catégorie de telles méthodes consiste à remplacer l'équation aux dérivées partielles par une équation aux différences et à calculer de proche en proche la solution de cette équation aux différences aux noeuds d'un quadrillage recouvrant le domaine, en partant des valeurs initiales et en utilisant les conditions de bord

La particularité de chacune de ces méthodes réside dans le type de schéma aux différences choisi.

(*) (5) page 99

II - Erreurs dans ces méthodes

Pour un problème donné, une fois la méthode choisie, il y a lieu de distinguer 3 solutions, ou plutôt 2 solutions et un résultat.

- la solution exacte de l'équation aux dérivées partielles que nous appellerons $u(x,t)$ et qui est définie en tout point intérieur au domaine.

- la solution exacte de l'équation aux différences que nous appellerons $U(x,t)$ et qui n'est définie qu'aux noeuds du quadrillage.

- le résultat obtenu effectivement par la résolution de l'équation aux différences, résultat que nous appellerons $\overline{U}(x,t)$ et qui, lui aussi, n'est défini qu'aux noeuds du quadrillage.

Ces trois ensembles de valeurs ne coïncident évidemment pas et, en chaque noeud du quadrillage, on peut définir :

- $U(x,t) - u(x,t)$ ou erreur de méthode qui dépend de la méthode choisie et des "mailles" du quadrillage : une méthode est convergente si cette erreur tend vers zéro lorsque l'on fait tendre vers zéro la longueur et la largeur des mailles.

- $\overline{U}(x,t) - U(x,t)$ ou erreur de calcul due en particulier aux erreurs d'arrondi qui s'accumulent : cette erreur de calcul dépend en général du volume de calcul effectué depuis le niveau initial et ne tend pas du tout vers 0 quand on "resserre" le quadrillage.

- $\overline{U}(x,t) - u(x,t)$ que l'on pourrait appeler erreur globale, la plus intéressante, mais la plus difficile à prévoir : elle est évidemment combinaison des deux autres erreurs.

On a essayé d'étudier l'erreur de calcul dans deux méthodes de résolution de l'équation de la chaleur :

- la méthode explicite la plus simple
- la méthode de Crank et Nicolson.

III - Méthodes d'étude de l'erreur d'arrondi

Pour étudier l'influence des erreurs de calcul sur le déroulement des méthodes, on a fait des hypothèses sur l'erreur supplémentaire introduite à chaque fois que l'on effectue l'une des quatre opérations arithmétiques

- pour le calcul en point décimal flottant, on a considéré que cette erreur élémentaire est une erreur relative dont l'ordre de grandeur est constant quelle que soit l'opération et quels que soient les opérandes.

- pour le calcul en point décimal fixe, on a considéré que cette erreur élémentaire est négligeable pour les additions et les soustractions et qu'elle a pour les multiplications et les divisions la forme d'une erreur absolue dont l'ordre de grandeur est constant et indépendant des opérandes.

En s'appuyant sur ces hypothèses simplificatrices, on a modifié les formules des deux méthodes de façon à tenir compte de ces erreurs élémentaires. L'objet de ce travail est d'essayer d'établir comment l'introduction de ces erreurs élémentaires modifie les résultats finaux des algorithmes.

D'autre part on a effectué une étude expérimentale pour tenter de vérifier par des passages sur calculatrice les conclusions ainsi obtenues. Pour cela on a écrit deux sous-programmes FORTRAN-MAP et deux procédures ALGOL ; deux appels de l'un de ces sous-programmes ou procédures par un programme ont pour effet de transformer toutes les instructions commandant une opération arithmétique flottante dans une partie déterminée du programme appelant et de faire exécuter les opérations correspondantes avec une erreur d'arrondi supérieur à la normale et déterminée à volonté par l'argument d'appel ou le paramètre effectif. Le premier de ces sous-programmes ou procédures simule ainsi des opérations arithmétiques flottantes effectuées avec une erreur d'arrondi grossie et de type flottant ; le second transforme les opérations flottantes en opérations fixes simulées, avec une erreur d'arrondi élémentaire de type fixe. L'utilisation de ces procédures a permis d'étudier

expérimentalement les rapports entre les erreurs d'arrondi élémentaires et les erreurs sur les résultats finaux.

C H A P I T R E I

ERREURS DE CALCUL DANS LA METHODE EXPLICITE

I - Principe de la méthode explicite

(*)

Le principe de la méthode explicite la plus simple consiste à remplacer l'équation aux dérivées partielles :

$$(1) \qquad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

par l'équation aux différences suivante :

$$(2) \qquad \frac{U(x,t+\Delta t) - U(x,t)}{\Delta t} = \frac{U(x-\Delta x,t) - 2.U(x,t) + U(x+\Delta x,t)}{(\Delta x)^2}$$

Δt et Δx étant les accroissements en t et en x.

Ce remplacement se justifie par les développements limités :

$$u(x,t+\Delta t) = u(x,t) + \Delta t.\left(\frac{\partial u}{\partial t}\right)_{(x,t)} + \frac{(\Delta t)^2}{2}.\left(\frac{\partial^2 u}{\partial t^2}\right)_{(x,t+\theta.\Delta t)}$$

$$u(x-\Delta x,t) + u(x+\Delta x,t) = 2u(x,t) + (\Delta x)^2.\left(\frac{\partial^2 u}{\partial x^2}\right)_{x,t)} + \frac{(\Delta x)^4}{12}.\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\vartheta.\Delta x,t)}$$

$0 \leq \theta \leq 1 \quad ; \quad -1 \leq \vartheta \leq +1$

d'où l'on tire :

$$(3) \qquad \left(\frac{\partial u}{\partial t}\right)_{(x\,t)} = \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + \frac{\Delta t}{2}.\left(\frac{\partial^2 u}{\partial t^2}\right)_{(x,t+\theta.\Delta t)}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t)} = \frac{u(x-\Delta x,t) - 2.u(x,t) + u(x+\Delta x,t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}.\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\vartheta.\Delta x,t)}$$

On voit que l'équation aux différences (2) exprime l'équation aux dérivées partielles (1) au point (x,t) en utilisant les expressions approchées que l'on peut tirer de (3)

(*) Sources pour cette méthode :

5 pages 92 et 100
8 page 91
2 page 7

On voit que l'erreur de méthode est en Δt et $(\Delta x)^2$ et que la méthode est convergente pourvu que l'on suppose bornées dans le domaine étudié les dérivées partielles de la fonction solution

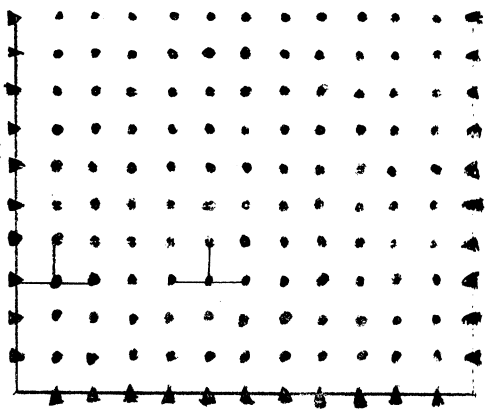
$$\left(\frac{\partial^2 u}{\partial^2}\right) \text{ et } \left(\frac{\partial^4 u}{\partial x^4}\right)$$

De (2) on tire l'expression

$$(4) \qquad U(x,t+\Delta t) = \frac{\Delta t}{(\Delta x)^2}\cdot\left[U(x-\Delta x,t)+\left(-2+\frac{(\Delta x)^2}{\Delta t}\right)\cdot U(x,t)+U(x+\Delta x,t)\right]$$

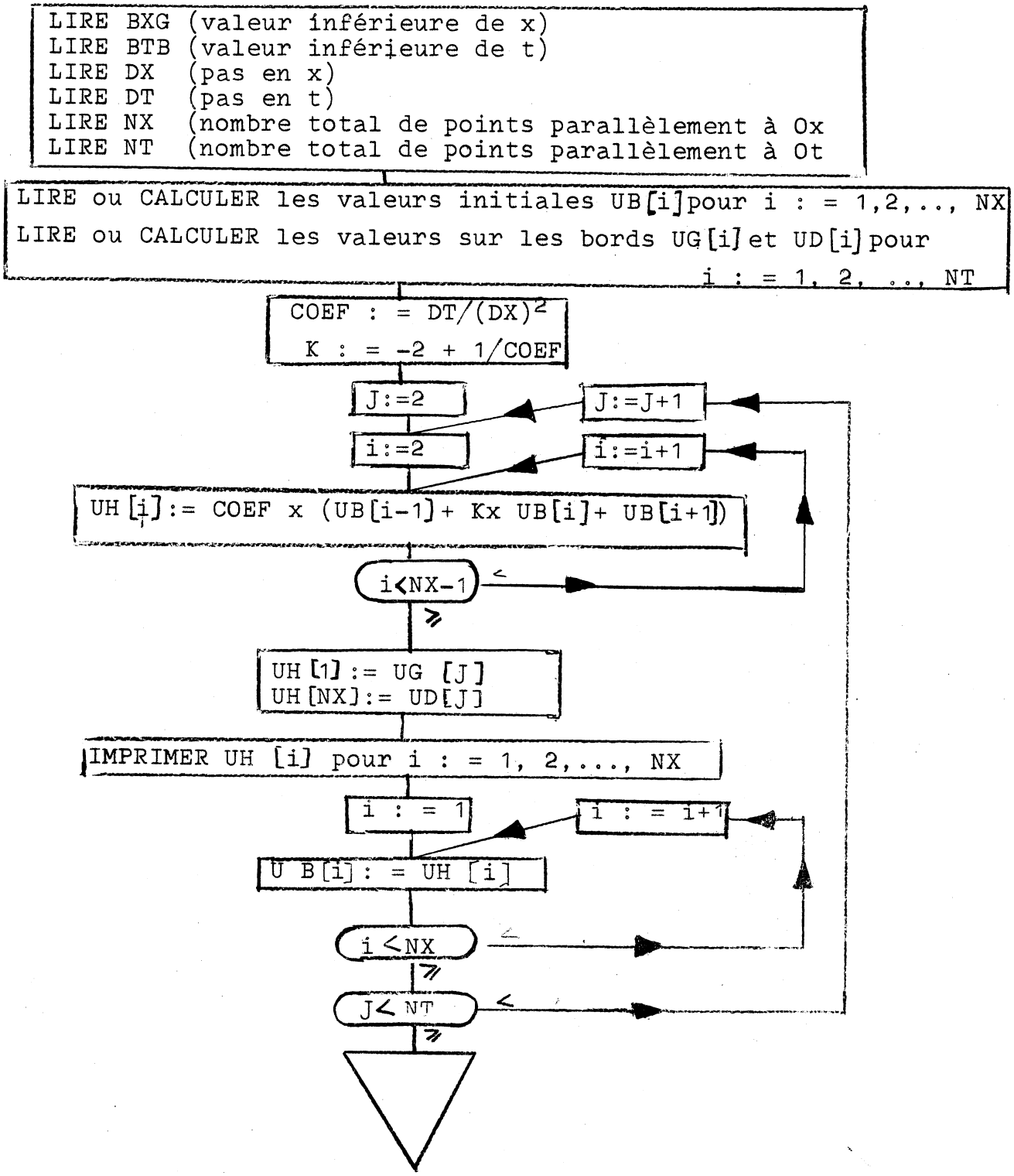
Ce qui exprime explicitement (d'où le nom de la méthode) chaque valeur de U au niveau $t+\Delta t$ en fonction de 3 valeurs de U au niveau t : si l'on discrétise un domaine borné en x, cela permet de progresser aussi loin que l'on veut en t pourvu que l'on connaisse les valeurs de U sur les bords.

Si l'on recouvre le domaine d'un quadrillage parallèle aux axes Ox et Ot, de maille Δx et Δt, la formule (4) définit un algorithme très simple permettant de calculer les valeurs de U aux noeuds du maillage en partant des valeurs initiales et en utilisant les valeurs sur les "bords" (*)



(*) Dans le programme Algol, les procédures d'entrée-sortie sont de l'un des types utilisables avec le compilateur ALGOL de Grenoble pour IBM 7044. Les procédures concernant les valeurs initiales et sur les bords ont un corps de procédure laissé en blanc : ce peut être soit une lecture de valeurs soit une tabulation de fonction.

ORGANIGRAMME DE LA METHODE EXPLICITE



PROGRAMME ALGOL DE LA METHODE EXPLICITE

```
DEBUT
     REEL  BXG, BTB, DX, DT, SIGMA, AMBDA, T, CA ;
     ENTIER NX, NT, I, P;
     ME : MØDELE (' (5 E 1 2. 8, 2 I 3) ') ;
     MSS: MØDELE (' (8H1METHØDE, 1X, 9HEXPLICITE//4HOBXG,4X, 1H=,1X,
          E16.8/1X,3HBTB,4X,1H=,1XE16.8/1X2HDX,5X,1H=,1X,E16.8/
          1X,2HDT.5X,1H=,1X,E16.8/1X,2HNX,5X,1H=,I4/1X,2HNT,
          5X,1H=,I4)') ;
     MSS 2 : MØDELE ('(6HOAMBDA,2X,1H=,1X,E16.8)') ;
     MS    :  MØDELE ('(1OHORESULTATS,1X,6HNIVEAU,I4,5X,2H(T,1X,
              1H=,E16.8,1H)/(6E16.8))') ;
     ENTREE (5,ME,1,BXG,1,BTB,1,DX,1,DT,1,SIGMA,1,NX,1,NT) ;
     COMMENTAIRE SIGMA INUTILE MAIS MEMES DONNEES QUE CN ;
     SØRTIE (6,MS,1,BXG,1,BTB,1,DX,1,DT,1,NX,1,NT) ;
     DEBUT
          TABLEAU UB, UH [1 : NX ], UG, UD [1 : NT ] ;
          REEL PRØCEDURE VALINI (I,X) ; VALEUR X, I;
               REEL X ; ENTIER I ,
               DEBUT
               FIN ;
          REEL PRØCEDURE VALBØRGAU  (I,X) ; VALEUR  X, I ;
               REEL X ; ENTIER I ;
               DEBUT
               FIN ;
          REEL PRØCEDURE VALBØRDRØI (I,X) ; VALEUR X, I ;
               REEL  X ; ENTIER  I ;
               DEBUT
               FIN ;
FRØNTIERES :
          PØUR I := 1 PAS 1 JUSQUA NX FAIRE
               UB [I]: = VALINI (I, BXG + DX x(I-1)) ;
```

```
        PØUR I  = 1 PAS 1 JUSQUA NT FAIRE
        DEBUT
            T : = BTB + DT x ( I-1) ;
            UG[ I ] : = VALBØRGAU (I,T) ;
            UD[ I ] : = VALBØRDRØI (I,T) ;
        FIN
        P : = 1 ; SØRTIE (6,MS,1,P,1,BTB,NX,UB[ 1 ] ) ;
CØNSTANTE :
        AMBDA : = DT/DX↑2 ; CA : = - 2 + 1 / AMBDA;
        SØRTIE (6,MSS2,1,AMBDA) ;
ALGØRITHME :
        PØUR P : = 2 PAS 1 JUSQUA NT FAIRE
        DEBUT
            PØUR I : = 2 PAS 1 JUSQUA NX - 1 FAIRE
                UH[ I] := AMBDA x (UB[ I-1] + CAxUB [I] +UB[ I+1] ) ;
            UH [1] : = UG [P] ; UH [NX] := UD [P] ; T :=BTB+DT*(P-1)
            SØRTIE (6, MS, 1, P, 1, T, NX, UH [1] ) ;
            POUR I : = 1 PAS 1 JUSQUA NX FAIRE UB[ I ] := UH [ I ];
        FIN
    FIN
FIN ;
```

II - Condition de stabilité de la méthode explicite

On dit qu'une équation aux différences est stable si une petite modification des valeurs initiales a sur les résultats des niveaux suivants en t une influence qui reste bornée quand t augmente indéfiniment.

Cette question de stabilité peut s'étudier de façon matricielle (*)

(*) [5] pages 93 et 99 le fait par l'intermédiaire des solutions explicites formelles des équations aux différences.
[10] chapitre 8 utilise les matrices mais la façon de procéder n'est pas exactement la même que celle employée ici.
Dans les travaux effectués à Nancy sous la direction de Mr LEGRAS (en particulier [3]), les matrices n'interviennent pas non plus de la même façon.

Soit U_p le vecteur dont les composantes sont les valeurs de U aux noeuds strictement intérieurs du $p^{ième}$ niveau en t au-dessus du niveau initial. Soit NX le nombre total de noeuds sur une parallèle à l'axe des x. U_p est un vecteur de R^{NX-2} et l'algorithme de la méthode explicite peut s'exprimer ainsi

$$U_{p+1} = A . U_p + V_p$$

$$U_p = A^p . U_0 + \sum_{j=1}^{p} A^{j-1} . V_{p-j}$$

La matrice A, de dimension (NX-2, NX-2), dépend uniquement de la méthode et du quadrillage ; ici elle est tri-diagonale

1)
$$A = K . \begin{bmatrix} K' & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & K' \end{bmatrix}$$

Avec $K = \dfrac{\Delta t}{(\Delta x)^2}$ et $K' = -2 + \dfrac{1}{K}$

U_0 et V_p sont des vecteurs de R^{NX-2}

U_0 exprime les conditions initiales.

V_p exprime les conditions de bord au niveau p :

2)
$$V_p = \begin{vmatrix} \alpha_p \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \beta_p \end{vmatrix}$$

Avec α_p et β_p = valeurs de U sur les bords

Soit U^*_p, le vecteur analogue à U_p mais obtenu après une légère modification des valeurs initiales. Soit

3)
$$E_p = U^*_p - U_p$$

La stabilité peut alors s'exprimer par le fait que E_p reste borné en norme quand p augmente indéfiniment

On a

$$U_{p+1} = A.\ U_p + V_p$$

$$U^*_{p+1} = A.\ U^*_p + V_p$$

La matrice A et le vecteur V_p sont les mêmes dans les 2 cas ; donc

$$E_{p+1} = A.E_p$$

(4)

$$E_p = A^p.\ E_0$$

E_0 exprimant les modifications de valeurs initiales.

La formule (4) montre donc qu'une condition nécessaire et suffisante de stabilité est que toutes les valeurs propres de la matrice A aient un module inférieur à 1.

On sait qu'une matrice carrée d'ordre n de la forme :

$$\begin{bmatrix} 1-2K & K & 1 & & 0 \\ K & 1-2K & K & \ddots & \\ & \ddots & \ddots & \ddots & 1 \\ & & \ddots & \ddots & K \\ 0 & & 1 & K & 1-2K \end{bmatrix}$$

possède n valeurs propres de la forme

(5) $$\lambda'_\ell = K + 2\cos\frac{\ell\pi}{n+1} \quad \text{avec } \ell = 1, 2, \ldots, n$$

La matrice A a donc Nx -2 valeurs propres :

(6) $$\lambda_\ell = K.(K' + 2\cos\frac{\ell\pi}{NX-1}) = 1 - 2K\,(1-\cos\frac{\ell\pi}{NX-1})$$

La condition nécessaire et suffisante de stabilité est donc :

$$|\lambda_\ell| \quad 1 \text{ pour } \ell = 1, 2, \ldots, NX-2$$

On a

$$|\lambda_\ell| < 4K-1 \text{ pour } \ell = 1,2,..,NX-2$$

et

$$K = \frac{\Delta t}{(\Delta x)^2}$$

Donc :

Théorème :

Une condition nécessaire et suffisante de stabilité de la méthode explicite est : $\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$

III - Erreurs de calcul dans la méthode explicite quand on calcule en flottant

On reprend les notations du § II.

L'algorithme peut s'exprimer par la relation de récurrence :

$$U_{p+1} = A . U_p + V_p$$

$$U_p = A^p . U_0 + \sum_{j=1}^{p} A^{j-1} V_{p-j}$$

Le passage du niveau p au niveau (p+1) exige les calculs suivants (avec des notations évidentes) :

$[K . (K' . u_{p1} + U_{p2} + \alpha_p)]$ pour la 1ère composante de U_{p+1}

$[K . (K' . u_{p(NX-2)} + U_{p(NX-3)} + \beta_p)]$ pour la $(Nx-2)^{ième}$ composante de U_{p+1}

$[K . (u_{p(i-1)} + K' . u_{pi} + u_{p(i+1)})]$ pour la $i^{ème}$ composante $i=2,3,\ldots, Nx-3$

Si l'on calcule en flottant, on peut considérer que l'ordre de grandeur de l'erreur relative commise sur chaque opération arithmétique élémentaire est constant. On peut donc appeler ε l'ordre de grandeur de l'erreur relative commise sur chaque calcul du type $[a . (b+c.d+e)]$. On peut alors considérer que le calcul des diverses composantes de U_{p+1}, dont on a vu le détail ci-dessus, produit la même erreur relative.

Si donc on appelle $\overline{U_p}$ le vecteur analogue à U_p, mais dans lequel on tient compte des erreurs de calcul, on a

$$\overline{U}_{p+1} = (1+\varepsilon) . (A . \overline{U_p} + V_p)$$

$$\overline{U_p} = (1+\varepsilon)^p . A^p . U_0 + \sum_{j=1}^{p} (1+\varepsilon)^j . A^{j-1} . V_{p-j}$$

D'où :

Théorème

Dans la résolution de l'équation de la chaleur par la méthode explicite, si l'on calcule en flottant,
si l'on commet sur chaque calcul du type $[a.(b+c.d+e)]$ une erreur relative de l'ordre de ε,
si l'on appelle U_p le vecteur des résultats théoriques de l'équa- aux différences aux noeuds du $p^{ième}$ niveau en t,
si l'on appelle $\overline{U}_p$ le vecteur annalogue à U_p mais réellement obtenu, si U_0 est le vecteur des conditions initiales on a

$$U_{P+1} = A.\,U_P + V_P \quad ; \quad U_P = A^P.\,U_0 + \sum_{j=1}^{P} A^{j-1}.\,V_{P-j}$$

$$\overline{U}_{P+1} = (1+\varepsilon).(A.\overline{U}_P + V_P) \quad ; \quad U_P = (1+\varepsilon)^P.A^P.U_0 + \sum_{j=1}^{P} (1+\varepsilon)^j A^{j-1} V_{P-j}$$

la matrice A et le vecteur V_p étant définis en (1) et (2) du § II.

Pour tirer les conclusions pratiques de ce théorème, on peut remarquer que :

- le terme $(1+\varepsilon)^P.A^P.U_0$ qui remplace le terme $A^P.U_0$ introduit une erreur relative proportionnelle à l'erreur relative élémentaire et au nombre de pas en t effectués depuis le niveau initial.

- si les conditions de bord sont nulles, les vecteurs V_P sont nuls et aucune erreur n'est due à leur calcul

- si les conditions de bord ne sont pas nulles, mais si le nombre de points est grand dans la direction de l'axe des x, le terme $\sum_{j=1}^{P} (1+\varepsilon)^j.A^{j-1}.V_{P-j}$ qui remplace le terme $\sum_{j=1}^{P} A^{j-1}.V_{P-j}$ introduit une erreur relative qui n'est pas strictement proportionnelle comme celle ci-dessus mais qui est négligeable puisque les vecteurs V_p n'ont que leur première et leur dernière composantes non nulles.

D'où :

Corollaire pratique

Dans la résolution de l'équation de la chaleur par la méthode explicite, si l'on calcule en flottant,
l'erreur de calcul est une erreur relative proportionnelle à l'erreur relative élémentaire ;
si, de plus, les conditions de bord sont nulles,
ou si le nombre de noeuds dans la direction de l'axe des x est grand, cette erreur relative moyenne est aussi proportionnelle au nombre de pas intermédiaires en t calculés depuis le niveau initial.

IV - Erreurs de calcul dans la méthode explicite quand on calcule en fixe.

Les types de calcul sont évidemment les mêmes que ceux exposés en (1) du paragraphe III dont on reprend les notations.

Mais, dans le calcul en fixe, on suppose que les erreurs élémentaires sur les additions et les soustractions sont négligeables et que les erreurs élémentaires sur les multiplications sont des erreurs absolues dont l'ordre de grandeur est constant et égal à ε. En suivant cette hypothèse on voit qu'il s'introduit dans le calcul de chaque composante de U_{p+1} en fonction de U_p une erreur absolue $(1+K).\varepsilon$

Par conséquent, on a

$$\bar{U}_{p+1} = A.\bar{U}_p + V_p + (1+K).\varepsilon.Z$$

(1)

$$\bar{U}_p = A^p.U_0 + \sum_{j=1}^{p} A^{j-1}.V_{p-j} + (1+K).\varepsilon.\sum_{j=1}^{p} A^{j-1}.Z$$

D'où :

Théorème

Dans la résolution de l'équation de la chaleur par la méthode explicite, si l'on calcule en fixe,
si l'on commet sur chaque multiplication une erreur absolue de l'ordre de ε,

si l'on appelle U_p le vecteur des résultats théoriques de l'équation aux différences aux noeuds du $p^{ième}$ niveau en t.
si l'on appelle $\bar{U}_p$ le vecteur analogue à U_p mais réellement obtenu,
si U_0 est le vecteur des conditions initiales.
on a :

$$U_{p+1} = A.U_p + V_p \qquad U_p = A^p.U_0 + \sum_{j=1}^{p} A^{j-1}.V_{p-j}$$

$$\bar{U}_{p+1} = A.\bar{U}_p + V_p + \varepsilon(1+K).Z \qquad \bar{U}_p = A^p.U_0 + \sum_{j=1}^{p} A^{j-1}V_{p-j} + \varepsilon(1+K)\sum_{j=1}^{p} A^{j-1}.Z$$

la matrice A et le vecteur V_p étant définis en (1) et (2) du §II et Z en (1) du §IV

$\varepsilon(1+K).\sum_{j+1}^{p} A^{j-1}.Z$ On remarque que le seul terme d'erreur est Donc si l'on appelle E_p le vecteur d'erreur du niveau p, on a :

$$E_{p\ 1} = E_p + \varepsilon.(1+K).A^{p-1}.Z$$

les scalaires ε et K et le vecteur Z sont constants au cours du calcul ; d'autre part, dans les conditions d'utilisation de la méthode explicite, on suppose satisfaite la condition de stabilité et les valeurs propres de la matrice A ont donc un module non supérieur à 1 ; on peut donc considérer que la moyenne des composantes du vecteur $\varepsilon.(1+K).A^{p-1}.Z$ reste constante quand p augmente.
D'où :

<u>Corollaire pratique</u>

Dans la résolution de <u>l'équation de la chaleur</u> par la <u>méthode explicite</u>, si l'on calcule en <u>fixe</u>,
l'erreur de calcul moyenne d'un certain niveau en t
est une erreur absolue qui est une fonction linéaire par rapport à l'erreur absolue commise sur chaque multiplication et par rapport au nombre de pas en t effectués depuis le niveau initial

V - Influence des erreurs de calcul sur la stabilité de la méthode explicite

Dans l'étude de la stabilité de la méthode explicite faite plus haut (*), on a supposé une petite modification des valeurs initiales (ou des valeurs d'un certain niveau en t) et on a étudié les répercussions que cette petite modification entraîne sur les résultats des niveaux suivants en t, mais ceci en supposant que les calculs des niveaux suivants sont effectués sans erreur. Dans la réalité, on effectue des erreurs de calcul à chaque niveau. On va reprendre l'étude de la stabilité en tenant compte de ces erreurs de calcul sur les niveaux ultérieurs

Au paragraphe II on a raisonné sur la relation de récurrence :

(1) $$U_{p+1} = A.U_p + V_p$$

et on a montré qu'une condition nécessaire et suffisante de stabilité théorique est que toutes les valeurs propres de la matrice A soient inférieurs en module à 1.

On a vu d'autre part (théorèmes des paragraphes III et IV) que l'effet des erreurs de calcul sur le déroulement de la méthode explicite pouvait s'exprimer par le remplacement de la relation (1) par les relations (2) ou (3) selon que l'on calcule en flottant ou en fixe :

(2) $$\overline{U_{p+1}} = (1+\varepsilon).(A.\overline{U_p} + V_p)$$

(3) $$\overline{U_{p+1}} = A.\overline{U_p} + V_p + (1+K).\varepsilon.Z$$

(*) paragraphe II page 11

On peut appliquer aux relations (2) et (3) un raisonnement analogue à celui appliqué à la relation (1) et l'on en déduit que, lorsque l'on tient compte des erreurs de calcul, la stabilité de la méthode explicite est liée aux valeurs propres des matrices $(1+\varepsilon).A$ et A selon que l'on calcule en flottant ou en fixe.

On voit tout de suite que, lorsque l'on calcule en fixe, la condition de stabilité n'est pas affectée par les erreurs de calculs puisque la matrice dont il s'agit est la même que dans la relation de récurrence théorique.

Dans le cas du calcul en flottant, la condition nécessaire et suffisante de stabilité est :

$$|\text{Valeurs propres de } [(1+\varepsilon).A]| \leq 1$$

c'est-à-dire, en tenant compte de la formule 6 du § II

$$|1+\varepsilon| . |1-2K.(1-\cos\frac{l\pi}{NX-1})| \leq 1 \text{ pour } l = 1,2,..,NX-2$$

alors que la condition sans tenir compte des erreurs de calcul est :

$$|1 - 2K.(1-\cos\frac{l\pi}{NX-1})| \leq 1 \text{ pour } l = 1,2,...,NX-2$$

A partir de cette dernière condition, on arrive à la condition

$$K \leq \frac{1}{2}$$

en prenant 2 comme valeur maximum de $(1-\cos\frac{l\pi}{u+1})$.

Devant cette approximation, la modification apportée par $(1+\varepsilon)$ est négligeable et la condition de stabilité n'est donc pas modifiée

Théorème

Dans la résolution de l'équation de la chaleur par la méthode explicite, que l'on calcule en flottant ou en fixe, les erreurs de calcul ne modifient pas la condition de stabilité

$$\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

VI - Etude expérimentale des erreurs de calcul dans la méthode explicite

On doit vérifier expérimentalement les conclusions auxquelles on est arrivé dans les paragraphes précédents

Pour cela, on a écrit des programmes exécutant la méthode explicite simultannément avec la précision maximum (*) et avec des erreurs d'arrondi grossies (**). Ces programmes calculent de plus à chaque noeud :

- la différence entre la solution en précision diminuée de l'équation aux différences et la solution analytique de l'

(1) équation aux dérivées partielles (solution connue dans les exemples choisis) : ce qui donne l'erreur globale telle qu'on l'a définie dans l'introduction

- la différence entre la solution en précision diminuée et la solution en précision maximum de l'équation aux différences : ce qui donne l'erreur de calcul telle qu'on l'a définie dans l'introduction, pourvu que l'on considère comme

(2) négligeables les erreurs de calcul qui s'introduisent même quand on utilise la précision maximum : étant données les précisions diminuées que l'on a prises, cette approximation semble légitime.

- la différence entre la solution en précision maximum de l'équation aux différences et la solution analytique : ce

(3) qui donne l'erreur de la méthode telle qu'on l'a définie dans l'introduction, pourvu que l'on fasse la même approximation que ci-dessus.

Les augmentations d'erreurs d'arrondi ont été faites successivement en flottant et en simulation de fixe (**).

(*) Comme précision maximum, on a pris la simple précision de la machine IBM 7044, en point décimal flottant : 27 chiffres binaires significatifs, ce qui correspond approximativement à 8 chiffres décimaux significatifs.

(**) par les sous-programmes TRONC et TROFXS : cf l'introduction et le chapitre 4.

Les différences ont été calculées comme des erreurs absolues pour le cas "fixe" et comme des erreurs relatives pour le cas flottant. Dans les deux cas, on s'est intéressé à l'erreur moyenne à chaque niveau en t, et à son évolution quand t augmente. On n'a pas interprêté les erreurs de méthode.

VII - Etude expérimentale des erreurs de calcul en flottant dans la méthode explicite

Pour le calcul en flottant, on a voulu vérifier les trois conclusions suivantes, auxquelles on a abouti dans les paragraphes 3 et 5 :

A - l'erreur de calcul moyenne est une erreur relative proportionelle au nombre de pas intermédiaires en t calculés depuis le niveau initial

B - l'erreur de calcul relative sur les résultats est proportionnelle à l'erreur relative élémentaire commise sur chaque opération.

C - les erreurs de calcul ne modifient pas de façon appréciable la condition de stabilité $\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$

La conclusion A semble vérifiée par l'examen des courbes portées au bas des graphiques 1 à 10 ; ces courbes donnent l'erreur relative moyenne de calcul d'un certain niveau en t en fonction du nombre de pas en t calculés depuis le début. Ce sont bien approximativement des droites ; cette linéarité peut s'apprécier par la disposition des points tracés dans la partie supérieure des graphiques : ceux-ci indiquent pour chaque valeur de t la pente de la courbe située à la partie inférieure : la linéarité parfaite correspondrait à une valeur constante : la plus ou moins grande dispersion de ces points par rapport à la valeur moyenne indique le caractère plus ou moins approximatif de la linéarité.

Sur le tableau de la page 23 on voit qu'on a fait varier 3 paramètres : (*)
- d'une part pour une erreur relative élémentaire en 10^{-6} et pour la fonction solution $e^{-t}.\sin x$ on a pris diverses valeurs de $\Delta t / (\Delta x)^2$
- d'autre part pour $\Delta t / (\Delta x)^2 = 0,5$ et toujours pour la fonction $e^{-t} \sin (x)$, on a pris des erreurs élémentaires en 10^{-7}, 10^{-5} 10^{-4}, 10^{-3}
- enfin, pour $\Delta t / (\Delta x)^2 = 0.5$ et pour une erreur élémentaire en 10^{-6}, on a pris la fonction $1000\ e^{-t}\sin x$ afin de vérifier que les erreurs sur les résultats sont bien des erreurs relatives indépendantes de l'ordre de grandeur des valeurs de la fonction solution.

La conclusion B semble vérifiée par l'examen des pentes moyennes de chacune des courbes décrites ci-dessus (dernière ligne du tableau de la page 23)
On voit que l'ordre de grandeur de la pente moyenne de chaque courbe ne dépend ni de la valeur de $\Delta t / (\Delta x)^2$ ni de l'ordre de grandeur des fonctions solutions mais seulement de l'ordre de grandeur de l'erreur relative élémentaire : c'est donc bien que l'erreur relative sur les résultats est proportionnelle à l'erreur relative élémentaire.

La conclusion C semble vérifiée par l'examen des graphiques 11 à 15. Ces graphiques sont établis d'une façon analogue aux graphiques 1 à 10 mais ils représentent cette fois-ci l'erreur relative globale. (*)

On trouve à la page 24 un tableau résumant les expériences qui sont à la source des graphiques 11 à 15.
On voit sur ces 5 graphiques qu'il y a stabilité tant que $\Delta t / (\Delta x)^2 \leq 0,5$ (courbes 11, 12 et 13) et instabilité dès que cette condition n'est plus vérifiée (courbes 14 et 15).

(*) On a tracé les graphiques 1 à 10 et 11 à 15, mais pour des raisons matérielles, on n'a reproduit que les graphiques 5, 7, 13 et 14.

- TABLEAU DES PARAMETRES DES GRAPHIQUES 1 à 10 -

N° de graphique	1	2	3	4	5	6	7	8	9	10
Date de passage en machine	25/2/64	25/2/64	25/2/64	26/2/64	26/2/64	13/3/64	13/3/64	16/3/64	5/5/64	5/5/64
Fonction solution analytique	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$10.e^{-t}.\sin x$
Valeur inférieure de x	0	0	0	0	0	0	0	0	0	0
Valeur inférieure de t	0	0	0	0	0	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,1.10^{-4}$	$0,2.10^{-4}$	$0,3.10^{-4}$	$0,4.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$
Nombre total de noeuds // à Ox	101	101	101	101	101	101	101	101	101	101
Nombre total de noeuds // à Ot	201	201	201	201	201	201	201	201	201	201
Nombre de bits binaires tronquées	7	7	7	7	7	3	10	13	17	7
Nombre de chiffres décimaux sign. exacts pour chaque opération élémentaire	5	5	5	5	5	6	4	3	2	5
Ordre de grandeur de l'erreur relative élémentaire	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-5}	10^{-4}	10^{-3}	10^{-6}
Valeur de $\frac{\Delta t}{(\Delta x)^2}$	0,1	0,2	0,3	0,4	0,5	0,5	0,5	0,5	0,5	0,5
Pente moyenne	$0,6.10^{-6}$	$0,48.10^{-6}$	$0,55.10^{-6}$	$0,47.10^{-6}$	$0,26.10^{-6}$	$0,13.10^{-7}$	$0,22.10^{-5}$	$0,15.10^{-4}$	$0,37.10^{-3}$	$0,26.10^{-6}$

TABLEAU DES PARAMETRES DES GRAPHIQUES 11 A 15.

	STABILITE			INSTABILITE	
N° de graphique	11	12	13	14	15
Date de passage en machine	25/2/64	2/3/64	26/2/64	2/3/64	26/2/64
Fonction solution analytique	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$
Valeur inférieure de x	0	0	0	0	0
Valeur inférieure de t	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,1.10^{-4}$	$0,49.10^{-4}$	$0,5.10^{-4}$	$0,51.10^{-4}$	$0,6.10^{-4}$
Nombre total de noeuds sur // à Ox	101	101	101	101	101
Nombre total de noeuds sur // à Ct	201	201	201	201	201
Ordre de grandeur de l'erreur relative élémentaire	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Valeur de $\Delta t / (\Delta x)^2$	0,1	0,49	0,5	0,51	0,6

D'autres expériences avec des erreurs élémentaires plus importantes ont confirmé ces résultats.

VIII - Etude expérimentale des erreurs de calcul en fixe dans la méthode explicite

Pour le calcul en fixe, on a voulu vérifier les trois conclusions suivantes, auxquelles on a abouti plus haut (*)

D - l'erreur de calcul moyenne d'un certain niveau en t est une erreur absolue proportionnelle au nombre de pas intermédiaires en t calculés depuis le niveau initial

E - l'erreur de calcul absolue sur les résultats est proportionnelle à l'erreur absolue élémentaire commise sur chaque opération.

F - les erreurs de calcul ne modifient pas de façon appréciable la condition de stabilité. $\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$

La conclusion D semble vérifiée par l'examen des courbes portées au bas des graphiques 21 à 30 (**) ces courbes donnent l'erreur absolue moyenne d'un certain niveau en t en fonction du nombre de pas en t calculés depuis le début. Ce sont bien approximativement des droites ; cette linéarité peut s'apprécier par la disposition des points tracés sur la partie supérieure des graphiques de la même façon que pour les graphiques 1 à 10 (***)

Sur cette linéarité, il faut noter 2 cas particuliers :

- Pour une grande erreur absolue élémentaire (en 10^{-3}) le "départ" de la droite se fait avec un certain retard (graphique 29)

(*) Corollaire pratique p. 17 et théorème p. 19

(**) On a tracé les graphiques 21 à 30 mais pour des raisons matérielles, on n'a reproduit ici que les graphiques 25 et 27.

(***) cf page 21.

- la linéarité de la courbe 30 n'est pas très satisfaisante mais cela peut venir de ce que cette courbe exprime l'erreur de calcul d'une façon moins précise que les autres, l'approximation expliquée en 2 du § VI est moins valable dans le cas de la courbe 30 : en effet la différence absolue entre la solution en précision maximum et la solution exacte de l'équation aux différences provient d'erreurs de calcul en flottant et l'ordre de grandeur de la fonction solution est 10^3 fois plus grand qu'ailleurs ; or on a vu que, dans le calcul en flottant, c'est l'erreur de calcul relative qui ne dépend pas de l'ordre de grandeur de la fonction solution; donc ici cette différence absolue entre la solution en précision maximum et la solution exacte n'est plus négligeable et l'approximation expliquée en (2) du § VI est moins valable.

Sur le tableau de la page 28 on voit qu'on a fait varier trois paramètres

- d'une part pour une erreur absolue élémentaire en 10^{-6} et pour la fonction solution $e^{-t}\sin x$, on a pris diverses valeurs de $\Delta t / (\Delta x)^2$.
- d'autre part, pour $\Delta t/(\Delta x)^2 = 0,5$ et toujours pour la fonction $e^{-t}\sin x$, on a pris des erreurs absolues élémentaires en 10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}.
- enfin, pour $\Delta t / (\Delta x)^2 = 0,5$ et pour une erreur absolue élémentaire en 10^{-6}, on a pris $10^3 . e^{-t}.\sin x$ pour vérifier l'influence de l'ordre de grandeur de la fonction solution.

La conclusion E semble vérifiée par l'examen des pentes moyennes de chacune des courbes qu'on vient de citer (dernière ligne du tableau de la page 28).

On constate des anomalies pour les courbes 29 et 30, mais ces anomalies s'expliquent par les remarques faites ci-dessus à propos de ces mêmes courbes.

En dehors de ces deux cas, on constate que l'ordre de grandeur de la pente moyenne de chaque courbe est proportionnel à l'ordre de grandeur de l'erreur absolue élémentaire : c'est donc bien que l'erreur absolue sur les résultats est proportionnelle à l'erreur absolue élémentaire.

La conclusion F semble vérifiée par l'examen des graphiques 31 à 35. Ces courbes sont analogues aux courbes du bas des graphiques 21 à 30 mais ils représentent cette fois-ci l'erreur absolue globale.

On trouve à la page 29 un tableau résumant les expériences qui sont à la source des graphiques 31 à 35.

On voit sur ces 5 graphiques qu'il y a stabilité tant que $\Delta t / (\Delta x)^2 \leqslant 0,5$ (courbes 31 et 32) et instabilité dès que cette condition n'est plus vérifiée (courbes 33, 34 et 35). D'autres expériences avec des erreurs élémentaires plus importantes ont confirmé ces résultats.

(*) On a tracé les graphiques 31 à 35 mais pour des raisons matérielles, on n'a reproduit ici que les graphiques 32 et 34

TABLEAU DES PARAMETRES RELATIFS AUX COURBES 21 à 30

N° de graphique	21	22	23	24	25	26	27	28	29	30
Date de passage en machine	5/5/64	4/5/64	4/5/64	4/5/64	4/5/64	5/5/64	4/5/64	6/4/64	5/5/64	28/4/64
Fonction solution analytique	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$3.e^{-t}.\sin x$
Valeur inférieure de x	0	0	0	0	0	0	0	0	0	0
Valeur inférieure de t	0	0	0	0	0	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,1.10^{-4}$	$0,2.10^{-4}$	$0,3.10^{-4}$	$0,4.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$
Nombre total de noeuds // à Ox	101	101	101	101	101	101	101	101	101	101
Nombre total de noeuds // à Ot	201	201	201	201	201	201	201	201	201	201
Nombre de bits binaires après la virgule	20	20	20	20	20	23	17	13	10	20
Nombre de chiffres décimaux exacts après la virgule pour chaque opération.	5	5	5	5	5	6	4	3	2	5
Ordre de grandeur de l'erreur absolue élémentaire	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-5}	10^{-4}	10^{-3}	10^{-6}
Valeur de $\Delta t / (\Delta x)^2$	0,1	0,2	0,3	0,4	0,5	0,5	0,5	0,5	0,5	0,5
Pente moyenne	$0,50.10^{-6}$	$0,52.10^{-6}$	$0,39.10^{-6}$	$0,5.10^{-6}$	$0,25.10^{-6}$	$0,3.10^{-7}$	$0,16.10^{-5}$	$0,2.10^{-4}$	$0,20.10^{-4}$	$0,24.10^{-7}$

TABLEAU DES PARAMETRES DES EXPERIENCES 31 A 35

	STABILITE		INSTABILITE		
N° de graphique	31	32	33	34	35
Date de passage en machine	5/5/64	4/5/64	12/5/64	12/5/64	12/5/64
Fonction solution analytique	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$	$e^{-t}\sin x$
Valeur inférieure de x	0	0	0	0	0
Valeur inférieure de t	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,1.10^{-4}$	$0,5.10^{-4}$	$0,51.10^{-4}$	$0,52.10^{-4}$	$0,55.10^{-4}$
Nombre total de noeuds à Ox	101	101	101	101	101
Nombre total de noeuds à Ot	201	201	201	201	201
Ordre de grandeur de l'erreur absolue élémentaire	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Valeur de $\Delta t / (\Delta x)^2$	0,1	0,5	0,51	0,52	0,55

C H A P I T R E II

EXPOSE DE LA METHODE DE CRANK ET NICOLSON

I Principe de la méthode de Crank et Nicolson

(*) Le principe de la méthode de Crank et Nicolson consiste à remplacer l'équation aux dérivées partielles :

$$(1) \qquad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

par l'équation aux différences suivantes :

$$(2) \qquad \frac{U(x,t+\Delta t)-U(x,t)}{\Delta t} = (1-\sigma).\frac{U(x-\Delta x,t)-2.U(x,t)+U(x+\Delta x,t)}{(\Delta x)^2} + \sigma.\frac{U(x-\Delta x,t+\Delta t)-2\,U(x,t+\Delta t)+U(x+\Delta x,t+\Delta t)}{(\Delta x)^2}$$

$$0 \leq \sigma \leq 1$$

(pour $\sigma = 0$, on retrouve la méthode explicite)

Ce remplacement se justifie par des développemen limités en supposant l'existence et la continuité des dérivées partielles de la fonctior solution qui interviennent dans ces développements.

On a :

$$(3) \qquad \left(\frac{\partial u}{\partial t}\right)_{(x,t)} = \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} + \frac{\Delta t}{2}.\frac{\partial^2 u}{\partial t^2}_{(x,t+\theta_1.\Delta t)}$$

avec $0 \leq \theta_1 \leq 1$

(*) Sources pour cette méthode :
- 5 pages 102 et 119
- 8 pages 16 et 91
- 2 page 30

(4) $$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t)} = \frac{u(x-\Delta x,t)-2.u(x,t)+u(x+\Delta x,t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}.\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_2.\Delta x,\,t)}$$

avec $-1 \leq \theta_2 \leq +1$

(5) $$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t+\Delta t)} = \frac{u(x-\Delta x,t+\Delta t)-2.u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}.\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_3.\Delta x,\,t+\Delta t)}$$

avec $-1 \leq \theta_3 \leq +1$

(6) $$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t+\theta_4.\Delta t)} = (1-\sigma).\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t)} + \sigma.\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t+\Delta t)} + \Delta t.\left[(1-\sigma).\theta_4\left(\frac{\partial^3 u}{\partial x^2.\partial t}\right)_{(x,t+\theta_5.\Delta t)} - \sigma(1-\theta_4).\left(\frac{\partial^3 u}{\partial x^2.\partial t}\right)_{(x,t+\theta_6.\Delta t)}\right]$$

avec $0 \leq \theta_4 \leq 1$ et $0 \leq \theta_5 \leq \theta_4 \leq \theta_6 \leq 1$

En combinant ces trois relations, on obtient

(7) $$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t+\theta_4.\Delta t)} = (1-\sigma).\frac{u(x-\Delta x,t)-2u(x,t)+u(x+\Delta x,t)}{(\Delta x)^2} + \sigma.\frac{u(x-\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{(\Delta x)^2} + \frac{(\Delta x)^2}{12}.\left[(1-\sigma).\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_2.\Delta x,\,t)} + \sigma.\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_3.\Delta x,\,t+\Delta t)}\right] + \Delta t.\left[(1-\sigma).\theta_4.\left(\frac{\partial^3 u}{\partial x^2.\partial t}\right)_{(x,t+\theta_5.\Delta t)} - \sigma(1-\theta_4).\left(\frac{\partial^3 u}{\partial x^2.\partial t}\right)_{(x,t+\theta_6.\Delta t)}\right]$$

On voit que l'équation aux différences (2) exprime l'équation aux dérivées partielles (1) en utilisant les expressions approchées que l'on peut tirer de (3) et (7).

On voit aussi que l'erreur de méthode est dans le cas général ($0 \leq \sigma \leq 1$) en Δt et $(\Delta x)^2$. La méthode est convergente pourvu que l'on suppose bornées dans le domaine étudié les dérivées partielles de la fonction solution

$$\left(\frac{\partial^2 u}{\partial t^2}\right), \left(\frac{\partial^4 u}{\partial x^4}\right) \text{ et } \left(\frac{\partial^3 u}{\partial x^2 . \partial t}\right)$$

Le plus souvent, on prend $\sigma = \frac{1}{2}$. Dans ce cas, on peut écrire à la place de (3) :

$$\left(\frac{\partial u}{\partial t}\right)_{(x,\, t+\frac{\Delta t}{2})} = \frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} + \frac{(\Delta t)^2}{24} \cdot \left(\frac{\partial^3 u}{\partial t^3}\right)_{(x,\, t+\theta_7 . \Delta t)} \tag{8}$$

avec $0 \leq \theta_7 \leq 1$

et, à la place de (6) :

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,\, t+\frac{\Delta t}{2})} = \frac{1}{2} \cdot \left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,t)} + \frac{1}{2} \cdot \left(\frac{\partial^2 u}{\partial x^2}\right)_{(x,\, t+\Delta t)} + \frac{(\Delta t)^2}{8} \cdot \left(\frac{\partial^4 u}{\partial t^4}\right)_{(x,\, t+\theta_8 . \Delta t)} \tag{9}$$

avec $0 \leq \theta_8 \leq 1$

En rapprochant (9) de (4) et (5), on obtient à la place de (7)

(10)
$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{x,t+\frac{\Delta t}{2}} = \frac{u(x-\Delta x,t)-2.u(x,t)+u(x+\Delta x,t)}{2.(\Delta x)^2}$$
$$+ \frac{u(x-\Delta x,t+\Delta t)+u(x+\Delta x,t+\Delta t)}{2.(\Delta x)^2}$$
$$+ \frac{(\Delta x)^2}{24}\cdot\left[\left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_2.\Delta x,t)} + \left(\frac{\partial^4 u}{\partial x^4}\right)_{(x+\theta_3.\Delta x,t+\Delta t)}\right]$$
$$+ \frac{\Delta t)^2}{8}\cdot\left(\frac{\partial^4 u}{\partial x^2.\partial t^2}\right)_{(x,t+\theta_8.\Delta t)}$$

On voit que, dans le cas particulier où $\sigma=\frac{1}{2}$, l'équation aux différences (2) exprime l'équation aux dérivées partielles (1) au point $(x,t+\frac{\Delta t}{2})$ en utilisant les expressions approchées que l'on peut tirer de (8) et (10).

On voit également que, dans le cas où $\sigma = \frac{1}{2}$, l'erreur de méthode est en $(\Delta t)^2$ et $(\Delta x)^2$. La méthode est convergente pourvu que l'on suppose bornées dans le domaine étudié les dérivées partielles de la fonction solution

$$\left(\frac{\partial^3 u}{\partial t^3}\right), \quad \left(\frac{\partial^4 u}{\partial x^4}\right) \text{ et } \left(\frac{\partial^4 u}{\partial x^2.\partial t^2}\right)$$

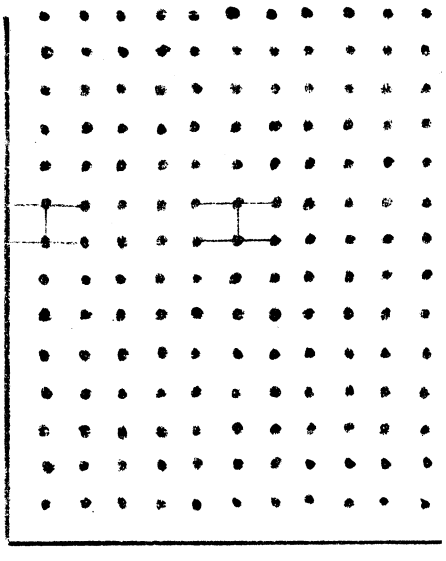
Dans (2), on peut grouper les termes du niveau t et les termes du niveau $(t+\Delta t)$: on obtient alors la relation

11)
$$U(x-\Delta x,t+\Delta t)-\left(2+\frac{1}{\sigma\lambda}\right).U(x,t+\Delta t)+U(x+\Delta x,t+\Delta t)$$
$$= \frac{\sigma-1}{\sigma}\cdot\left[U(x-\Delta x,t)+\left(-2+\frac{1}{(1-\sigma)\lambda}\right).U(x,t)+U(x+\Delta x,t)\right]$$
avec $\lambda = \frac{\Delta t}{(\Delta x)^2}$

Contrairement à ce qui se produit dans la méthode explicite, cette relation ne fournit pas une expression explicite de chaque valeur de U au niveau $(t + \Delta t)$ en fonction de plusieurs valeurs de U du niveau t. Simplement cette relation relie les valeurs de la solution en trois points du niveau $(t + \Delta t)$ et en trois points du niveau t : c'est pourquoi on dit que la méthode de Crank et Nicolson est une méthode implicite.

Si l'on discrétise un domaine borné en x et si l'on écrit la relation (11) pour tous les points strictement intérieurs et situés sur un certain niveau en t, on obtient un système d'équations linéaires où les inconnues sont les valeurs de U au niveau $(t + \Delta t)$ et où les valeurs sur les "bords" du domaine et les valeurs au niveau t interviennent dans les seconds membres : la résolution de ce système linéaire permet de passer du niveau t au niveau $(t + \Delta t)$: on peut ainsi progresser aussi loin que l'on veut en t pourvu que l'on connaisse les valeurs de U sur les "bords".

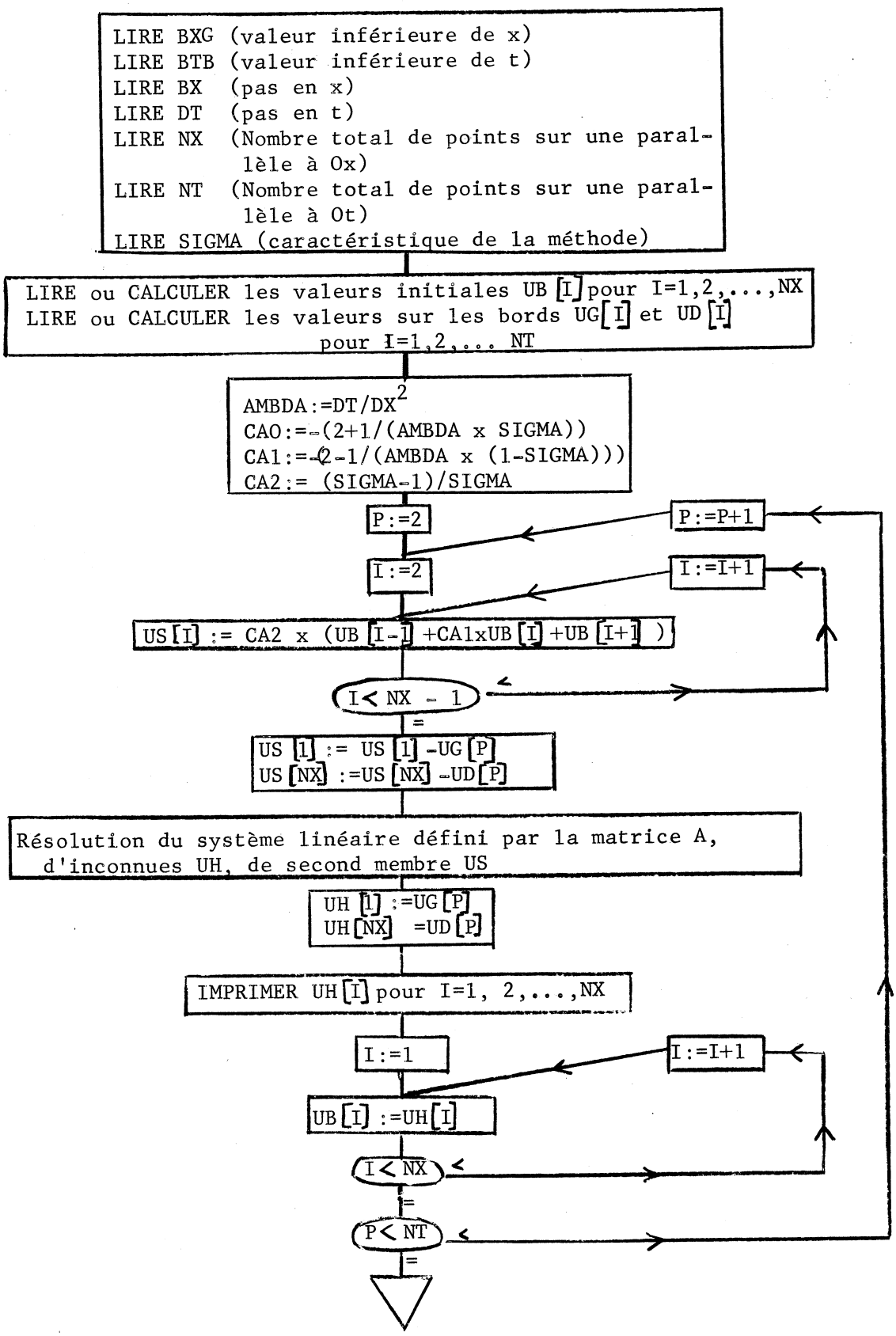
Si l'on recouvre le domaine à étudier d'un quadrillage parallèle aux axes $0x$ et $0t$, de maille Δx et Δt, la résolution successive des systèmes linéaires formés à l'aide de la relation (11) forme un algorithme permettant calculer les valeurs de U aux noeuds du maillage en partant des valeurs initiales et en utilisant les valeurs sur les bords



II- Mise en oeuvre de la méthode de Crank et Nicolson

On a écrit l'organigramme de cette méthode de Crank et Nicolson et un programme ALGOL l'éxécutant ; les valeurs initiales et sur les bords sont soit données discrètement, soit données par des fonctions ; c'est pourquoi, dans le programme ALGOL, on amis seulement DEBUT et FIN pour les corps des trois procédures correspondantes qui peuvent être prévues soit pour exécuter de simples lectures, soit pour tabuler une fonction. On a de même simplement mis DEBUT et FIN pour le corps de la procédure de résolution des systèmes linéaires. Les procédures d'entrée-sortie sont les "procédures étendues" du compilateur AIGOL de Grenoble pour I.B.M. 7044.

ORGANIGRAMME DE LA METHODE DE CRANK ET NICOLSON



Programme algol de la méthode de Crank et Nicolson

```
DEBUT
LECTURES :
   REEL BXG,BTB,DX,DT,SIGMA,CAO,CA1,CA2,AMBDA,T ;
   ENTIER NX,NT,I,P ;
ME:MODELE ('(5E12.8,2I3)') ;
MSS:MODELE ('(8H1METHODE,1X,2HDE,1X,5HCRANK,2X,2HET,2X,8HNICOLSON/
   /4HOBXG,4X,1H=,1X,E16.8/1X,3HBTB,4X,1H=,1X,E16.8/1X,2HDX,5X,1H=,
   1X,E16.8/1X,2HDT,5X,1H=,1X,E16.8/1X,5HSIGMA,2X,1H=,1X,E16.8/1X,
   2HNX,5X,1H=,I4/1X,2HNT,5X,1H=,I4)') ;
MSS2:MODELE ('(6HOAMBDA,2X,1H=,1X,E16.8/1X,3HCAO,4X,1H=,1X,E16.8/1X,
   3HCA1,4X,1H=,1X,E16.8/1X,3HCA2,4X,1H=,1X,E16.8)') ;
MS:MODELE ('(1OHORESULTATS,1X,6HNIVEAU,I4,5X,2H(T,1X,1H=,E16.8,1H)/
   (6E16.8))') ;
   ENTREE (5,ME,1,BXG,1,BTB,1,DX,1,DT,1,SIGMA,1,NX,1,NT ) ;
   SORTIE (6,MSS,1,BXG,1,BTB,1,DX,1,DT,1,SIGMA,1,NX,1,NT) ;
   DEBUT
      TABLEAU UB,UH,US [1:NX] , UG,UD [1:NT] ;
PROCEDURES :
      REEL PROCEDURE VALINI (X,I)   VALEUR X,I ;
         REEL X ; ENTIER I ;
         DEBUT
         FIN ;
      REEL PROCEDURE VALBORGAU (X,I) ; VALEUR X, I ;
         REEL X ; ENTIER I ;
         DEBUT
         FIN ;
      REEL PROCEDURE VALBORDROI (X,I) ; VALEUR X,I ;
         REEL X ; ENTIER I ;
         DEBUT
         FIN ;
```

(suite du programme ALGOL de la méthode de Crank et Nicolson)

```
        PROCEDURE TRISPE (C,N,X,B) ; VALEUR C,N,B ;
          REEL  C, ENTIER N ; TABLEAU X,B [1:N ] ;
          COMMENTAIRE RESOUT SYST. LIN. D'ORDRE N-2 ;
          DEBUT
          FIN ;
FRONTIERES :
   POUR I:=1 PAS 1 JUSQUA NX FAIRE
        UB [I] := VALINI (BXG+DXx(I-1),I) ;
   POUR I:=1 PAS 1 JUSQUA NT FAIRE
   DEBUT
        T =BTB + DT x (I-1) ;
        UG [I] := VALBORGAU (T,I) ;
        UD [I] := VALBORDROI (T,I) ;
   FIN
   P:=1
   SORTIE (6,MS,1,P,1,BTB,NX,UB [ 1] ) ;
CONSTANTES :
   AMBDA:=DT/DX ↑ 2 ;
   CA0:=-(2+1/(AMBDAxSIGMA)) ;
   CA1:=-(2-1/(AMBDAx(1-SIGMA))) ;
   CA2:=(SIGMA-1)/SIGMA ;
   SORTIE (6,MSS2,1,AMBDA,1,CA0,1,CA1,1,CA2) ;
ALGORITHME :
   POUR P:=2 PAS 1 JUSQUA NT FAIRE
   DEBUT
          T:=BTB+ D Tx(P-1) ;
          POUR I:=2 PAS 1 JUSQUA NX-1 FAIRE
               US [I] :=CA2x(UB[I-1] +CA1xUB[I]+UB[I+1]) ;
          US[1] :=US [1] -UG[P] ; US[NX]:=US[NX] -UD [P] ;
          TRISPE (CA0,NX,UH,US) ;
          UH[1]:= UG[P] ; UH[NX] := UD[P] ;
          SORTIE (6,MS,1,P,1,T,NX,UH[1]) ;
          POUR I:=1 PAS 1 JUSQUA NX FAIRE
               UB[I] := UH[I] ;
     FIN
  FIN
FIN ;
```

Cette méthode de Crank et Nicolson peut s'exprimer au moyen de matrices d'une manière analogue à la méthode explicite (*).

Soit U_p le vecteur dont les composantes sont les valeurs de U aux noeuds strictement intérieurs du $p^{ième}$ niveau en t au-dessus du niveau initial. Soit NX le nombre total de noeuds sur une parallèle à l'axe des x. U_p est un vecteur de R^{NX-2}.
L'algorithme peut s'exprimer ainsi :

(1) $$A.U_{p+1} = B.U_p + V_p$$

(2) $$U_{p+1} = A^{-1}.B.U_p + A^{-1}.V_p$$

(3) $$U_p = (A^{-1}.B)^P.\ U_0 + \sum_{j=1}^{P} (A^{-1}.B)^{j-1}.\ A^{-1}.\ V_{p-j}$$

Les matrices A et B sont de dimensions (NX-2,NX-2) et ne dépendent que de la méthode et du quadrillage. Ici elles sont tridiagonales :

$$A = \begin{pmatrix} K & 1 & & & 0 \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & K \end{pmatrix} \quad ; \quad B = K' \cdot \begin{pmatrix} K' & 1 & & & 0 \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & 1 & K' \end{pmatrix}$$

$$K = -\left(2 + \frac{1}{\lambda.\sigma}\right) \quad ; \quad K' = -\left(2 - \frac{1}{(1-\sigma).\lambda}\right) \quad ; \quad K'' = \frac{\sigma - 1}{\sigma}$$

$$\lambda = \frac{\Delta t}{(\Delta x)^2}$$

(*) cf page 12

U_0 et V_p sont des vecteurs de R^{NX-2}

U_0 exprime les conditions initiales

V_p fait intervenir les conditions de bord aux niveaux p et p+1 ;

$$V_p = \begin{vmatrix} v_{p,1} \\ o \\ o \\ \cdot \\ \cdot \\ \cdot \\ o \\ o \\ v_{p,NX-2} \end{vmatrix}$$

$v_{p,1} = - ug\,[p+1] + K''.ug\,[p]$

$v_{p,NX-2} = - ud\,[p+1] + K''.ug\,[p]$

$ug\,[p]$ et $ud\,[p]$ = valeurs imposées sur les bords

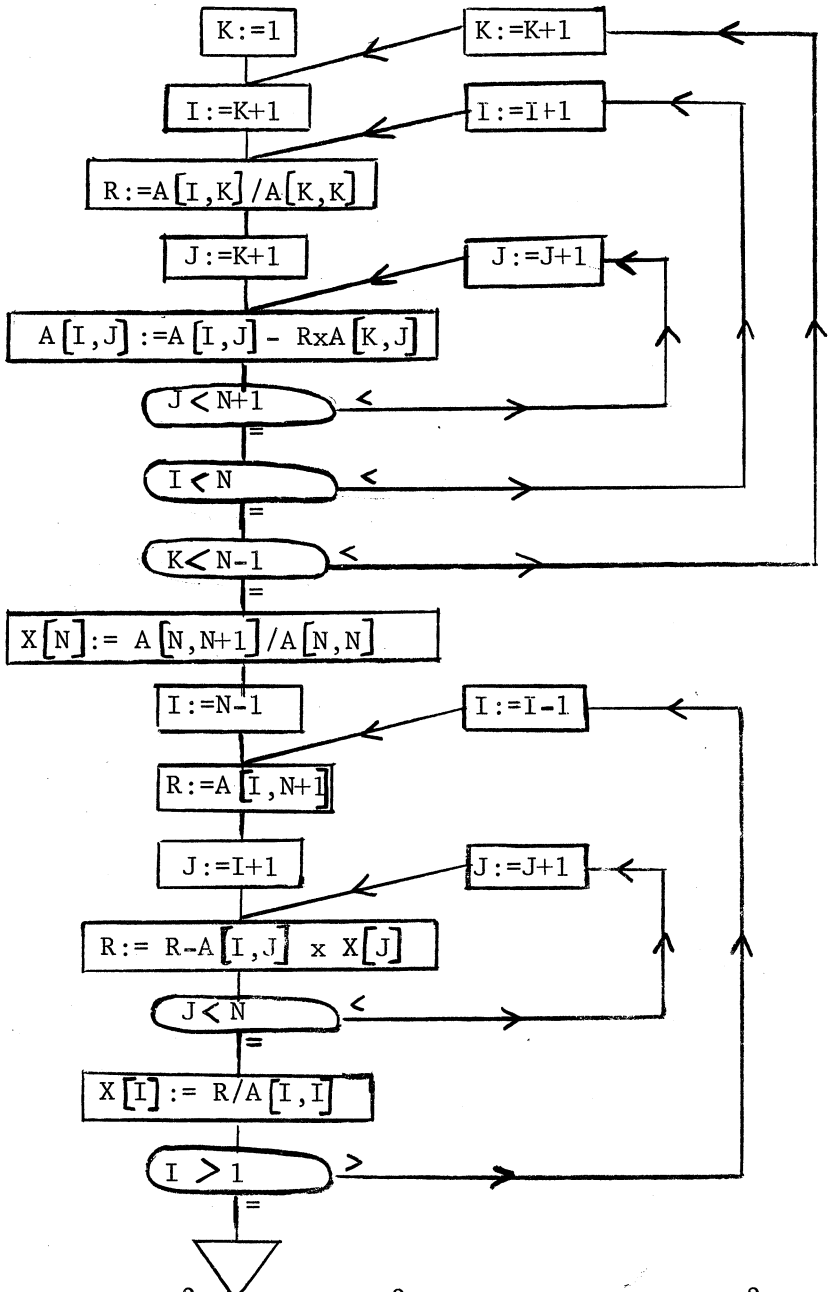
Le passage de la forme (1) à la forme (2) suppose au point de vue numérique la résolution d'un système de (NX-2) équations linéaires à (NX-2) inconnues. La forme (3) suppose les résolutions successives depuis le début de p tels systèmes linéaires ayant tous la même matrice A au premier membre et des colonnes seconds membres différentes. Dans la résolution de ces systèmes linéaires, on tient évidemment compte de la forme tridiagonale de la matrice A du premier membre ; mais on peut employer diverses méthodes. On en étudiera ici 2, qui sont toutes 2 des particularisations de la méthode directe de simple élimination de Gauss l'une dans la variante avec division par les pivots, l'autre dans la variante qui évite ces divisions par les pivots.

III-Utilisation dans la méthode de Crank et Nicolson de l'élimination avec divisions.

L'algorithme général de cette méthode de résolution des systèmes linaires s'exprime par la relation suivante : (*)

$$a_{ij}^{(K+1)} = a_{ij}^{(K)} - \frac{a_{ij}^{(K)}}{a_{KK}^{(K)}} \cdot a_{Kj}^{(K)} \quad ; K=1 ;\ldots,n-1 ; i=n+1,\ldots,n; j=K+1,\ldots,n+1$$

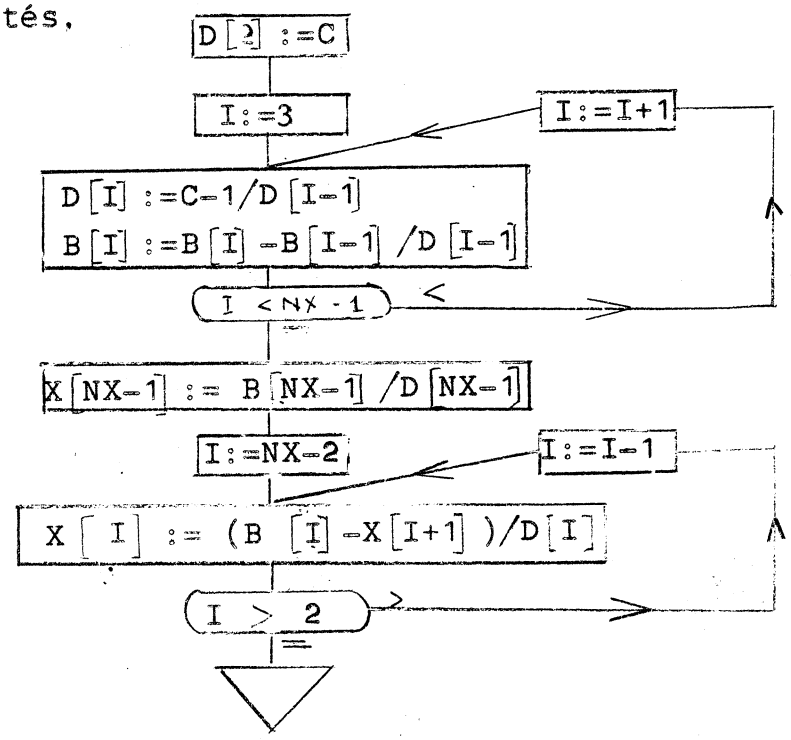
L'organigramme général de cette méthode est le suivant :



On voit que cela nécessite environ n^2 mémoires, $\frac{n^3}{3}$ multiplications et $\frac{n^2}{2}$ divisions

(*) Sources pour la méthode d'élimination avec divisions : (7). IV 9 sq et p. V 34
(4). p 63 sq

Si l'on tient compte de la forme particulière des systèmes rencontrés dans la méthode de Crank et Nicolson, on voit que, pour chaque pas en K, il suffit de transformer la ligne (K+1) et, dans cette ligne, en dehors de l'élément (K+1,K) qui s'annule, seuls sont transformés les eléments (K+1 K+1) et (K+1, n+1), l'élément (K+1,K+2) restant égal à 1 sans calcul. De plus, dans notre cas, on résoud des systèmes de dimensions NX-2 en travaillant sur des tableaux de dimension NX. Voici un organigramme tenant compte de ces particularités.



On voit qu'on utilise 3n mémoires au lieu de n (n+2) et qu'on exécute 3 n divisions au lieu de $\frac{n^2}{2}$ divisions et $\frac{n^3}{3}$ multiplications.

Voici une procédure ALGOL appliquant cette organigramme et utilisable par le programme de la méthode de Crank et Nicolson :

```
PROCEDURE TRISPE (C,N,X,B) ; VALEUR C,N,B ;
REEL C ; ENTIER N ; TABLEAU X,B [1:N] ;
COMMENTAIRE RESOUT SYST. LIN. TRID. D'ORDRE N-2 ;
DEBUT
        TABLEAU  D [ 2 : N-1 ] ; ENTIER I ;
        D  [2] : = C ;
        POUR I:= 3 PAS 1 JUSQUA N-1 FAIRE
        DEBUT
                D[I]:= C - 1/D [I-1] ;
                B[I]:= B [I] - B [I-1] /D[I-1] ;
        FIN ;
        X[N-1] := B [N-1] /D [N-1] ;
        POUR I:=N-2 PAS -1 JUSQUA 2 FAIRE
                X [I] := (B [I] - X [I+1] ) / D [I];
FIN ;
```

IV - Utilisation dans la méthode de Crank et Nicolson de l'élimination sans division

L'algorithme général de cette méthode de résolution des systèmes linéaires s'exprime par la relation suivante : (*)

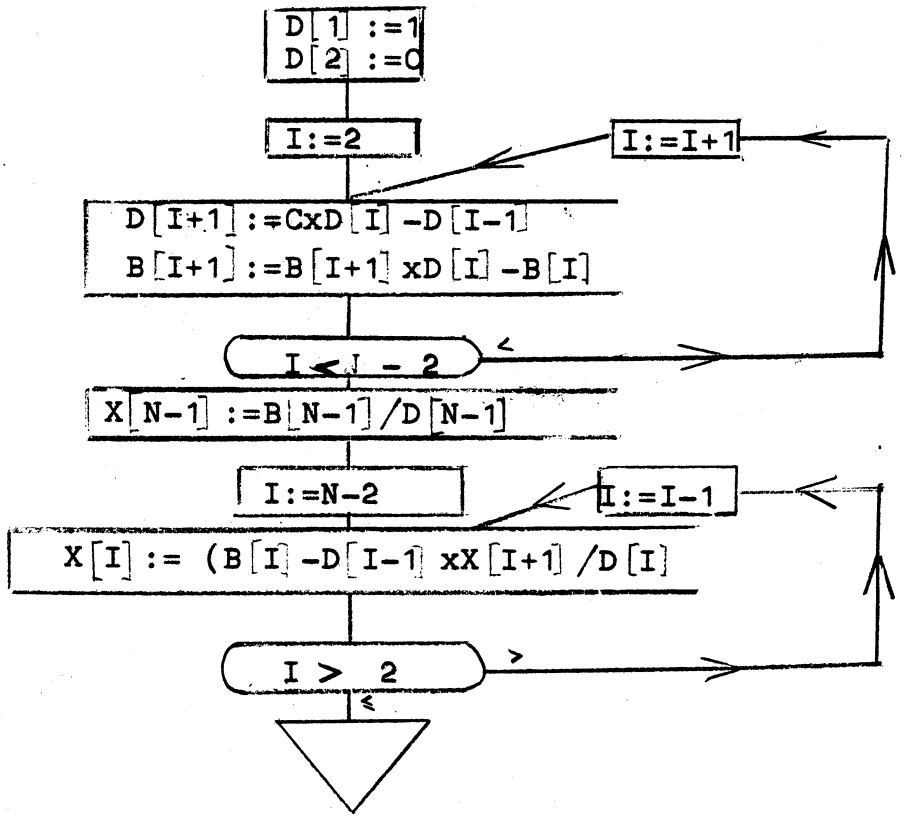
$$a_{ij}^{(K+1)} = a_{ij}^{(K)} \cdot a_{KK}^{(K)} - a_{Kj}^{(K)} \cdot a_{iK}^{(K)} \quad ; \quad K=1,..,n-1 \; ; \; i=K+1,..,n \; ; \; j=K+1,..,n+1$$

L'organigramme général de cette méthode est analogue à celui de la méthode avec divisions, les seules différences étant :
- la suppression du calcul de R
- le remplacement de l'ordre A[I,J]:= A [I, J] -RxA [K, J]
par l'ordre A[I, J] := A [I, J]x A[K,K]- A[K, J]x A [i,K]

(*) Source pour cette méthode d'élimination sans division : (7).IV.30

Cela nécessite environ n^2 mémoires, $\frac{2}{3} n^3$ multiplications et $\frac{n^2}{2}$ divisions.

Si l'on particularise cette méthode à notre problème, on obtient :



On voit qu'on utilise 3 n mémoires et que l'on exécute 3 n multiplications et n divisions.

Voici une procédure algol exécutant cette particularisation de la méthode sans division et utilisable par le programme de la méthode de Crank et Nicolson

```
PROCEDURE TRISPE (C,N,X,B) ; VALEUR C,N,B ;
REEL C ; ENTIER N ; TABLEAU X, B[1:N] ;
COMMENTAIRE RESOUT SYST. LIN. TRID. D'ORDRE N-2 ;
DEBUT
        TABLEAU  D [1:N-1] ; ENTIER  I ;
        D[1] :=1 ; D[2] := C ;
        POUR I:=2 PAS 1 JUSQUA N-2 FAIRE
        DEBUT
             D[I+1] :=CxD[I]-D[I-1] ;
             B[I+1] :=B[I+1]xD[I]-B[I] ;
        FIN
        X[N-1] :=B[N-1]/ D[N-1] ;
        POUR I:= N-2 PAS -1 JUSQUA 2 FAIRE
             X [I] := (B [I] - D [I-1] x X[I+1] ) / D [I] ;
FIN ;
```

V-Stabilité de la méthode de Crank et Nicolson

On rappelle qu'une équation aux différences est stable si une petite modification des valeurs initiales a sur les résultats des niveaux suivants en t une influence qui reste bornée quand t augmente indéfiniment.

On a vu (*) que les résultats de la méthode de Crank et Nicolson pour le $p^{ième}$ niveau en t au-dessus du niveau initial satisfont à

$$U_p = (A^{-1}.B)^p.U_0 + \sum_{j=1}^{p} (A^{-1}.B)^{j-1}.A^{-1}.V_{p-j}$$

Ces matrices A et B étant de dimension (NX-2, NX-2) :

(*) page 40

$$A = \begin{bmatrix} K & 1 & & \\ 1 & \ddots & & \\ & & \ddots & 1 \\ & & 1 & K \end{bmatrix} \quad ; \qquad B = K'' \cdot \begin{bmatrix} K' & 1 & & \\ 1 & \ddots & & \\ & & \ddots & 1 \\ & & 1 & K' \end{bmatrix}$$

K, K' et K" étant définis plus haut.

Soit U^*_p le vecteur analogue à U_p mais obtenu après une légère modification des valeurs initiales.
Soit :

$$E_p = U^*_p - U_p$$

La stabilité peut alors s'exprimer comme étant le fait que E_p reste borné quand p augmente indéfiniment.
On a

$$U_p = (A^{-1}.B)^p.U_0 + \sum_{j=1}^{p} (A^{-1}.B)^{j-1}.A^{-1}.V_{p-j}$$

$$U^*_p = (A^{-1}.B)^p.U^*_0 + \sum_{j=1}^{p} (A^{-1}.B)^{j-1}.A^{-1}.V_{p-j}$$

Les matrices A et B et les vecteurs V_{p-j} sont les mêmes dans les deux cas : donc

$$U^*_p - U_p = (A^{-1}.B)^p.\ (U^*_0 - U_0)$$

$$E_p = (A^{-1}.B)^p.E_0$$

E_0 exprimant les modifications initiales ;
Donc :

Lemme 1

Une condition nécessaire et suffisante de stabilité de la méthode de Crank et Nicolson est que toutes les valeurs propres de la matrice $(A^{-1}.B)$ aient un module ≤ 1

$$A = \begin{bmatrix} K & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K \end{bmatrix} ; \quad B = K'' \cdot \begin{bmatrix} K' & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K' \end{bmatrix}$$

A et B de type (NX-2,NX-2)

Soit à chercher les valeurs et les vecteurs propres de matrices de la forme :

$$\begin{bmatrix} c & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & c \end{bmatrix}$$

Soit λ_p une telle valeur propre et soit $w_{p,i}$ la $i^{ème}$ composante du vecteur propre correspondant. Trouver les valeurs et les vecteurs propres revient à trouver des λ_p et des $w_{p,i}$ tels que :

(1) $$w_{p,i-1} + (c-\lambda_p) \cdot w_{p,i} + w_{p,i+1} = 0$$

(2) $$w_{p,0} = w_{p,n+1} = 0$$

où n est l'ordre de la matrice.

L'équation (1) représente une récurrence linéaire homogène d'ordre 2. Son équation caractéristique est :

(3) $$r^2 + (c - \lambda_p).r + 1 = 0$$

et sa solution générale est de la forme :

(4) $$w_{p,u} = \mu_1 . r_1^u + \mu_2 . r_2^u$$

où r_1 et r_2 sont les racines de (3)
Si r_1 et r_2 sont réels, on ne pourra satisfaire (2).
Supposons donc λ tel que les racines de (3) soient non réelles
Le produit de ces racines est égal à 1 d'après (3)
Donc ces racines sont de la forme :

(5) $$r_1 = e^{i\theta} \quad ; \quad r_2 = e^{-i\theta}$$

et l'on a

(6) $$w_{p,n} = \mu_1 e^{in\theta} + \mu_2 e^{-in\theta}$$

On trouve θ en reportant (5) dans (3) :

$$e^{2i\theta} + (c - \lambda_p) . e^{i\theta} + 1 = 0$$

c'est-à-dire

(7) $$-(c - \lambda) = 2 \cos \theta$$
$$\theta = \pm \text{Arc cos} \left(\frac{\lambda - c}{2} \right)$$

μ_1 et μ_2 étant des constantes complexes, on peut avoir comme solution particulière de (1) :

$$w_{p,n} = \mu_1 e^{ni\theta} + \mu_2 e^{-ni\theta} = \sin(n\theta)$$

(2) est alors vérifié si l'on prend

$$\theta = \frac{p\pi}{n+1} \text{ avec } p \text{ entier } 0 < p \leq n$$

λ est alors déterminé par (7).

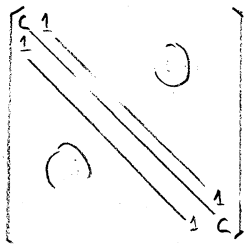
Finalement la solution de (1) et (2) est :

$$\lambda_p = c + 2\cos\frac{p\pi}{n+1} \qquad w_{n,p} = \sin\frac{np\pi}{n+1} \qquad p=1,2,\ldots,n)$$

Donc

Lemme 2

Une matrice carrée d'ordre n, tridiagonale de la forme



possède n valeurs propres :

$$\lambda_p = c + 2\cos\frac{p\pi}{n+1}$$

auxquelles correspondent n vecteurs propres

$$W_p = \begin{vmatrix} w_{p,1} \\ w_{p,2} \\ \vdots \\ w_{p,n} \end{vmatrix} \quad \text{avec } w_{p,j} = \sin\left(\frac{j\,p\pi}{n+1}\right)$$

On a vu au lemme 1 que la stabilité de la méthode de Crank et Nicolson est liée aux valeurs propres de certaines matrices. On a vu au lemme 2 la forme des valeurs et vecteurs propres des matrices de ce type. On va rapprocher ces 2 résultats :

Les matrices A et B définies au lemme 1 et attachées à la méthode de Crank et Nicolson sont du type étudié au lemme 2. Donc les matrices A et B ont des valeurs et des vecteurs propres

du type étudié au lemme 2. Ces valeurs propres sont différentes pour A et B mais les vecteurs propres sont les mêmes pour A et B.
D'autre part A et A^{-1} ont mêmes vecteurs propres et des valeurs propres inverses. Donc les matrices A, A^{-1}, B et $(A^{-1}.B)$ ont mêmes vecteurs propres.

Donc, si μ_{A_p}, μ_{B_p} et μ_p désignent les valeurs propres correspondant au vecteur propre W_p respectivement pour les matrices A, B, et $(A^{-1}.B)$, on a

$$\mu_p = \frac{\mu_{B_p}}{\mu_{A_p}}$$

Or, d'après les lemmes 1 et 2 :

$$\mu_{A_p} = K + 2 \cos\left(\frac{p\pi}{NX-1}\right)$$

$$\mu_{B_p} = K''.\left(K' + 2 \cos\left(\frac{p\pi}{NX-1}\right)\right)$$

Si l'on prend les valeurs de K, K' et K'' définies plus haut (*) on a

$$\mu_{A_p} = \frac{1}{\sigma.\lambda}\left[-1 - 2\sigma.\lambda.\left(1-\cos\frac{p\pi}{NX-1}\right)\right]$$

$$\mu_{B_p} = \frac{1}{\sigma.\lambda}\left[-1 + 2.\lambda.(1-\sigma).\left(1-\cos\frac{p\pi}{NX-1}\right)\right]$$

Donc :

$$\mu_p = \frac{1 - 2.\lambda.(1-\sigma).\left(1-\cos\frac{p\pi}{NX-1}\right)}{1 + 2\sigma.\lambda\left(1-\cos\frac{p\pi}{NX-1}\right)}$$

On peut donc transformer le lemme 1 en :

(*) page 40

Lemme 3

Une condition nécessaire et suffisante de stabilité de la méthode de Crank et Nicolson est que l'on ait pour p = 1,2,...,NX-2

$$|\mu_p| \leq 1$$

avec

$$\mu_p = \frac{1-2.\lambda.(1-\sigma).\left(1-\cos\frac{p\pi}{NX-1}\right)}{1+2.\sigma.\lambda.\left(1-\cos\frac{p\pi}{NX-1}\right)}$$

On cherche donc à quelles conditions est vérifié

$$\frac{|1-2.\lambda.(1-\sigma)\,Q_p|}{|1+2.\sigma.\lambda.Q_p|} \leq 1 \quad \text{avec } Q_p = 1-2\cos\frac{p\pi}{NX-1}$$

On remarque que

$$0 < Q_p < 2 \quad \text{et} \quad |1+2.\sigma.\lambda.Q_p| = 1+2.\sigma.\lambda Q_p$$

a) Si $2\lambda(1-\sigma).Q_p < 1$

$$|1-2\lambda(1-\sigma).Q_p| = 1-2\lambda.(1-\sigma).Q_p$$

et la condition s'exprime :

$$1-2\lambda.(1-\sigma).Q_p - 1 - 2\sigma.\lambda.Q_p \leq 0$$

$$-2.\lambda.Q_p \leq 0$$

ce qui est toujours satisfait puisque λ et Q_p sont positifs

b) Si $2.\lambda.(1-\sigma).Q_p > 1$

$$|1-2.\lambda.(1-\sigma).Q_p| = -1+2.\lambda.(1-\sigma).Q_p$$

et la condition s'exprime :

$$2.\lambda(1-\sigma).Q_p - 1 - (1 - 2.\sigma.\lambda Q_p) \leq 0$$

$$\lambda.Q_p.(1 - 2\sigma) - 1 \leq 0$$

comme $0 < Q_p \leq 2$ cela revient à

- ou bien $\sigma \geq \frac{1}{2}$ et λ quelconque
- ou bien $\sigma < \frac{1}{2}$ et $\lambda \leq \frac{1}{2\,(1-\sigma)}$

On retrouve ainsi le résultat classique. (*)

Théorème

Dans la méthode de Crank et Nicolson,
si $\sigma \geq \frac{1}{2}$, il y a stabilité quel que soit $\frac{\Delta t}{(\Delta x)^2}$

si $\sigma < \frac{1}{2}$ la condition de stabilité est $\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2\,(1-\sigma)}$

(*) cf par exemple (5) page 122

C H A P I T R E 3

ERREURS DE CALCUL DANS LA METHODE DE CRANK ET NICOLSON

I - Erreurs de calcul en flottant dans la résolution de systèmes linéaires par l'élimination avec divisions

Si l'on résout le système linéaire AX = Y en calculant en point décimal flottant et en utilisant la méthode générale de simple élimination de Gauss, dans sa version avec divisions, la solution effectivement obtenue peut être considérée comme la solution exacte du système (*)

(1) $$(A + \delta A) . X = Y + \delta Y$$

avec

(2) $$\delta A = \sum_{k=1}^{n-1} (C^{(k)} + \delta_1 A^{(k)})$$

(3) $$\delta Y = (\sum_{k=1}^{n-1} \delta_1 Y^{(k)}) + \delta_2 Y + \delta_3 Y$$

(4) $$C^{(k)} = \begin{bmatrix} 0 & & 0 \\ & \overset{k}{\downarrow} & \\ 0 \cdots & 0 & \cdots 0 \leftarrow k \\ & \varepsilon^{(k)}_{k+2} \cdot \bar{a}^{(k)}_{k+1,k} & \\ 0 & \varepsilon^{(k)}_{k+2} \cdot \bar{a}^{(k)}_{k+2,k} & 0 \\ & \vdots & \\ & \varepsilon^{(k)}_{n} \cdot \bar{a}_{n,k} & \end{bmatrix} \quad ; \quad \delta_1 A^{(k)} = \begin{bmatrix} 0 & & 0 \\ & \overset{k}{\downarrow} & \\ 0 \cdots & 0 & \cdots 0 \leftarrow k \\ 0 & \eta^{(k)}_{ij} \cdot \bar{a}^{(k+1)}_{ij} & 0 \end{bmatrix}$$

(5) $$\delta_1 Y = \begin{bmatrix} 0 \\ \rho^{(k)}_{k+1} \cdot \overline{y}^{(k+1)}_{k+1} \\ \rho^{(k)}_{k+2} \cdot \overline{y}^{(k+1)}_{k+2} \\ \vdots \\ \rho^{(k)}_{n} \cdot \overline{y}^{(k+1)}_{n} \end{bmatrix} \quad ; \quad \delta_2 Y = \begin{bmatrix} \gamma_1 \cdot \bar{a}^{(1)}_{11} \cdot \bar{\xi}_1 \\ \gamma_2 \cdot \bar{a}^{(2)}_{22} \cdot \bar{\xi}_2 \\ \vdots \\ \gamma_n \cdot \bar{a}^{(n)}_{nn} \cdot \bar{\xi}_n \end{bmatrix} \quad ; \quad \delta_3 Y = \begin{bmatrix} \mu_{12} \cdot \bar{s}_{12} + \mu_{13} \bar{s}_{13} \cdots \mu_{1n} \cdot \bar{s}_{1n} \\ \mu_{23} \bar{s}_{23} \; \mu_{24} \bar{s}_{24} \cdots \mu_{2n} \cdot \bar{s}_{2n} \\ \vdots \\ \mu_{n-1,n} \cdot \bar{s}_{n-1,n} \\ 0 \end{bmatrix}$$

(6) $a^{(k)}_{ij}$ même notation que dans description de l'algorithme (**)

$\rho^{(k)}_{k+p}$ erreur relative dans le calcul $\bar{a}^{(k)}_{k+p} = \bar{a}^{(k)}_{k+p} / \bar{a}^{(k)}_{kk}$ + erreur

---

(*) Mr Gastinel l'a montré : (6) théorème VII page 71 et explicitations pages 69 et 70.

(**) page 42

$\eta_{ij}^{(k)}$ = erreur relative dans le calcul $\bar{a}_{ij}^{(k+1)} = \bar{a}_{ij}^{(k)} - \bar{\rho}_i^{(k)} \bar{a}_{kj}^{(k)}$ + erreur

$p_i^{(k)}$ = erreur relative dans le calcul $\bar{y}_i^{(k+1)} = \bar{y}_i^{(k)} - \bar{\rho}_i^{(k)} \cdot \bar{y}_k^{(k)}$ + erreur

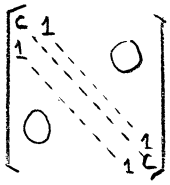
S_{ik} = sommes partielles dans le retour AR: $\bar{S}_{i,n+1} = \bar{y}_i^{(i)}$; $\bar{S}_{i,k} = \bar{S}_{i,k+1} - \bar{a}_{i,k}^{(i)} \cdot \bar{\xi}_i$ + erreur

(7) μ_{ik} = erreur relative dans le calcul de ces $S_{i,k}$

γ_i = erreur relative dans le calcul : $\bar{\xi}_i = \bar{S}_{i,i+1} / \bar{a}_{ii}^{(i)}$ + erreur

$\bar{\xi}_i$ = $i^{ème}$ résultat réellement obtenu.

Si la matrice du premier membre est de la forme



et si l'on en tient compte selon la particularisation de la méthode exposée plus haut (*), à chaque pas en k, on ne transforme que la ligne (k+1), et, dans cette ligne, on ne transforme par calcul que les éléments (k+1,k) et (k+1,n+1) (2^{nd} membre) ; les éléments (k+1,k) et (k+1,k+2) sont pris respectivement égaux à 0 et 1 sans calcul; de plus :

$$\bar{\rho}_{k+1}^{(k)} = 1 / \bar{a}_{kk}^{(k)} + \text{erreur}$$

et, en fait, on ne calcule pas successivement :

$$\bar{\rho}_{k+1}^{(k)} = 1 / \bar{a}_{kk}^{(k)} + \text{erreur}$$

puis

$$\bar{a}_{k+1,j}^{(k+1)} = \bar{a}_{k+1,j}^{(k)} - \bar{\rho}_{k+1}^{(k)} \cdot \bar{a}_{kj}^{(k)} + \text{erreur}$$

(*) page 43

mais directement :

$$\bar{a}^{(k+1)}_{k+1,j} = \bar{a}^{(k)}_{k+1,j} - \bar{a}^{(k)}_{kj} / \bar{a}^{(k)}_{kk} + \text{erreur}$$

Les seules erreurs non nulles sont donc :

$$p^{(k)}_{k+1} , \eta^{(k)}_{k+1,k+1} , \gamma_i$$

On pose :

(8) $$\bar{d}_k = \bar{a}^{(k)}_{kk}$$

On peut alors définir plus précisément les erreurs non nulles dans notre cas :

(9) $p_{k+1}=p^{(k)}_{k+1}$= erreur relative sur le calcul: $\bar{y}^{(k+1)}_{k+1}=\bar{y}^{(k)}_{k+1}-\bar{y}^{(k)}_{k}/d_k$+erreur.

$\eta_{k+1}=\eta^{(k)}_{k+1,k+1}$=erreur relative sur le calcul: $\bar{d}_{k+1}=c-1/\bar{d}_k$+erreur

γ_i= erreur relative sur le calcul : $\xi_i = (y_i - \xi_{i+1})/d_i$

soit

(10) E_{ij} la matrice carrée d'ordre n qui a un 1 en position (i,j) et des 0 partout ailleurs.

En reportant dans (4) et (5), on obtient :

(11) $$c^{(k)} = 0 \; ; \quad \delta_1 A^{(k)} = \eta_{k+1} \cdot \bar{d}_{k+1} \cdot E_{k+1,k+1}$$

(12) $$\delta_1 Y^{(k)} = \begin{bmatrix} 0 \\ \vdots \\ p_{k+1} \cdot \bar{y}^{(k+1)}_{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} ; \quad \delta_2 Y = \begin{bmatrix} \gamma_1 \cdot d_1 \cdot \xi_1 \\ \gamma_2 \cdot d_2 \cdot \xi_2 \\ \vdots \\ \gamma_n \cdot d_n \cdot \xi_n \end{bmatrix} ; \quad \delta_3 Y = 0$$

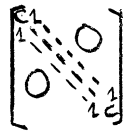
En portant (12) et (11) dans (3) et (2), on obtient :

(13) $$\delta A = \begin{bmatrix} 0 & & & 0 \\ & \eta_2 \cdot \bar{d}_2 & & \\ & & \eta_3 \cdot \bar{d}_3 & \\ & & & \ddots \\ 0 & & & \eta_n \cdot \bar{d}_n \end{bmatrix} ; \quad \delta Y = \begin{bmatrix} 0 \\ p_2 \cdot \bar{y}^{(2)}_2 \\ p_3 \cdot \bar{y}^{(3)}_3 \\ \vdots \\ p_n \cdot \bar{y}^{(n)}_n \end{bmatrix} + \begin{bmatrix} \gamma_1 \cdot d_1 \cdot \xi_1 \\ \gamma_2 \cdot d_2 \cdot \xi_2 \\ \vdots \\ \gamma_n \cdot d_n \cdot \xi_n \end{bmatrix}$$

d_i étant défini en (8) et η_i, p_i et γ_i en (9)

Donc :

Théorème 1

Dans la résolution du système linéaire AX=Y, où A =  si l'on calcule en point décimal flottant, si l'on utilise l'élimination avec divisions particularisée, la solution réellement obtenue peut être considérée comme la solution exacte de :

$$(A+\delta A)X = Y + \delta Y$$

A et δY étant définies en (13)

Si maintenant on suppose de plus que l'erreur relative sur chaque calcul du type (a-b)/c ou a-b/c est de l'ordre de ε. (13) devient :

(14) $$\delta A = \varepsilon . \begin{bmatrix} 0 & d_2 & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} ; \quad \delta Y = \varepsilon . \begin{bmatrix} 0 \\ y_2^{(2)} \\ \vdots \\ y_n^{(n)} \end{bmatrix} + \varepsilon . \begin{bmatrix} d_1 \xi_2 \\ d_2 \xi_2 \\ \vdots \\ d_n \xi_n \end{bmatrix}$$

En raison des approximations que l'on a faites, on a le droit de modifier un peu la 1ère ligne de la matrice δA et celle de la colonne δY : il vient alors :

(15) $$\delta A = \varepsilon . \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} ; \quad \delta Y = \varepsilon . \begin{bmatrix} y_1^{(1)} \\ y_2^{(2)} \\ \vdots \\ y_n^{(n)} \end{bmatrix} + \varepsilon . \begin{bmatrix} d_1 \xi_2 \\ d_2 \xi_2 \\ \vdots \\ d_n \xi_n \end{bmatrix}$$

on voit alors que :

(16) $$\delta A = \varepsilon . D \; ; \quad \delta Y = \varepsilon . Y' + \varepsilon D . X$$

(17) $$D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$; (d_i étant défini en (8)

Y' est le vecteur issu de Y après qu'on lui ait appliqué les transformations qui triangulent la matrice A.
On peut exprimer ces transformations de façon matricielle : à chaque pas en k, on soustrait à la $k^{ème}$ ligne la $(k-1)^{ème}$ ligne multipliée par $1/d_{k-1}$: ce qui revient à pré-multiplier par $(I - \frac{1}{d_{k-1}} \cdot E_{k,k-1})$.

E_{ij} étant la matrice définie en (10). On fait cela pour k=2,3,.
Donc, l'ensemble des transformations aboutissant à la triangularisation de la matrice A peut s'exprimer par la pré-multiplication par la matrice

$$T = (I - \frac{1}{d_k} \cdot E_{2,1})(I - \frac{1}{d_k} \cdot E_{32}) \cdots (I - \frac{1}{d_{n-1}} E_{n,n-1})$$

$$= \prod_{i=1}^{n-1} (I - \frac{1}{d_i} \cdot E_{i+1,i})$$

Mais l'on a

(18) $$E_{ij} \, E_{kl} = 0 \text{ si } j \neq k$$

Donc

$$T = I - \frac{1}{d_1} \cdot E_{21} - \frac{1}{d_2} \cdot E_{32} \cdots - \frac{1}{d_{n-1}} E_{n,n-1} =$$

$$I - \sum_{i=1}^{n-1} \left(\frac{1}{d_i} \cdot E_{i+1,i} \right)$$

(19) $$T = \begin{bmatrix} 1 & & 0 \\ \frac{1}{d_1} & \ddots & \\ 0 & & 1 \end{bmatrix}$$ (d_i étant défini en (8))

T étant ainsi défini, on a :

$$Y' = T.Y$$

(16) devient :

$$\delta A = \varepsilon.D \; ; \quad \delta Y = \varepsilon.T.Y + \varepsilon.D.X$$

On reporte dans (1) : il vient :

$$(A + \varepsilon.D).X = Y + \varepsilon.T.Y + \varepsilon.D.X$$

On met en facteur en tenant compte du fait que le signe de ε est incertain : il vient :

$$(A + 2.\varepsilon.D)\, X = (I + \varepsilon.T)\, Y$$

Donc

Théorème 2

Dans la résolution de l'équation linéaire $AX = Y$

où $A = \begin{pmatrix} \varepsilon & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & \varepsilon \end{pmatrix}$

si l'on calcule en point décimal flottant,
si l'on utilise l'élimination avec divisions particularisée
si l'on considère que sur chaque opération du type a-b/c ou (a-b)/c on fait une erreur relative de l'ordre de ε,
la solution réellement obtenue peut être considérée comme la solution exacte de

$$(A + 2.\varepsilon.D).X = (I + \varepsilon.T).\, Y$$

D et T étant définis en (17) et (19)

II - Erreurs de calcul en "flottant" dans la méthode de Crank et Nicolson quand on utilise l'élimination avec divisions

On a vu (*) que la méthode de Crank et Nicolson exige à chaque pas en t la résolution du système linéaire :

$$A.U_{p+1} = B.U_p + V_p$$

Le second membre exige les calculs :

$K''(K'x_1+x_2+ug_p)-ug_{p+1}$ et $K''(x_{n-1}+K'x_n+ud_p)-ud_{p+1}$ pour la 1ère et la dernière composante, $K''(x_{i-1}+K'x_i+x_{i+1})$ pour la $i^{ème}$ composante.

Si l'on considère que l'erreur relative sur chaque calcul du type a+b.c ou (a+b)c est de l'ordre de ε, les erreurs relatives sur les expressions ci-dessus sont à peu près de l'ordre de 2ε et le second membre est en fait remplacé par

$$(1+2\varepsilon).(B.U_p + V_p)$$

Donc si l'on applique le théorème 2 du paragraphe précédent on a :

Théorème :

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,
si l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires,
si l'on calcule en point décimal flottant,
si l'on fait sur chaque calcul du type a-b/c,(a-b)/c,a+b.c, (a+b).c une erreur relative de l'ordre de ε,
si l'on appelle U_p le vecteur des résultats théoriques de l'équa

(*) page 35

-tion aux différences aux noeuds du $p^{ème}$ niveau en t,
si l'on appelle $\bar{U}_p$ le vecteur analogue à U_p, mais obtenu réellement,
si U_0 est le vecteur des conditions initiales, et V_p le vecteur des conditions de bord au $p^{ème}$ niveau, on a

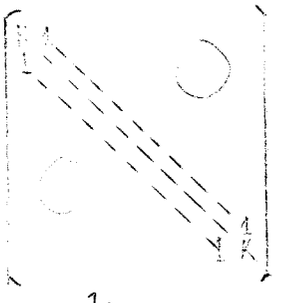
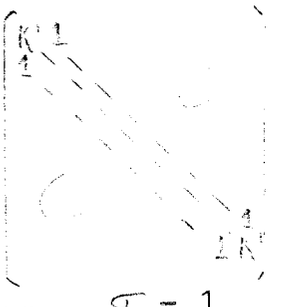
$$A.U_{p+1} = B.U_p + V_p$$

$$(A + 2.\varepsilon.D).\bar{U}_{p+1} = (I+\varepsilon.T).(1+2\varepsilon).(B.\bar{U}_p + V_p)$$

$$U_p = (A^{-1}.B)^p.U_0 + \sum_{j=1}^{p} (A^{-1}.B)^{j-1}.A^{-1}.V_{p-j}$$

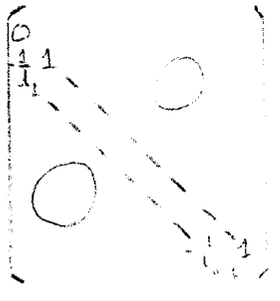
$$\bar{U}_p = (1+2\varepsilon).(A+2\varepsilon.D)^{-1}.(I+\varepsilon.T).B^p.U_0 + \sum_{j=1}^{p} (1+2\varepsilon)(A+2\varepsilon.D)^{-1}(I+\varepsilon T)B^{j-1} \cdot$$

$$\cdot (1+2\varepsilon).(A+2\varepsilon D)^{-1}.(I+\quad T).V_{p-j}$$

$A =$  ; $B = K'$ 

$$K = -(2+\frac{1}{\sigma k}) \; ; \; K' = -(2-\frac{1}{(1-\sigma)k}) \; ; \; K'' = \frac{\sigma - 1}{\sigma}$$

$k = \dfrac{\text{pas en t}}{(\text{pas en x})^2}$; $0 \leq \sigma \leq 1$; caractéristique de la méthode

$D =$  ; $T =$ 

d_i étant le $i^{ème}$ élément diagonal de A après triangularisation.

Pour tirer des conclusions pratiques de ce théorème on peut remarquer que :

- dans le coefficient matriciel de U_0, le terme $(1+2\varepsilon)^p$ introduit une erreur relative proportionnelle à ε et à p car $(1+2\varepsilon)^p \simeq 1+2.\varepsilon.p$.
- l'erreur introduite par le facteur $(I+\varepsilon T)$ peut aussi être considérée comme une erreur relative proportionnelle à ε et à p si l'on s'intéresse à l'erreur moyenne sur les résultats d'un niveau en p et à condition que l'ordre des matrices (c'est-à-dire le nombre de points selon l'axe des x) soit grand, et à condition également que les composantes de U_p aient des ordres de grandeur pas trop différents les uns des autres.
- il en est de même de l'erreur introduite par le remplacement de A^{-1} par $(A+2\varepsilon D)^{-1} \simeq A^{-1} - 2\varepsilon.A^{-1}.D.A^{-1}$.
- le changement du coefficient matriciel des V_{p-j} n'introduit aucune erreur si les conditions de bord sont nulles car alors les vecteurs V_{p-j} sont tous nuls ;
- si les conditions de bord ne sont pas nulles, les V_{p-j} ne sont pas nuls mais ont seulement leur première et leur dernière composante non nulles; donc, si le nombre de points selon l'axe des x est grand, c'est-à-dire si le nombre total de composantes des vecteurs V_{p-j} et U_0 est grand, les erreurs introduites par la modification des coefficients matriciels des V_{p-j} sont négligeables devant les erreurs dues aux modifications du coefficient matriciel de U_0 ;
- les termes d'erreurs ne dépendent pas des valeurs de la fonction solution.

Ces remarques donnent une idée de la forme des erreurs. Pour étudier leur ordre de grandeur, il faut évaluer les éléments des matrices d'erreurs, c'est-à-dire ici des d_i.

Ceux-ci sont définis par la récurrence :

$$d_{i+1} = K - \frac{1}{d_i} \; ; \; d_1 = K$$

or

$$K = - (2 + \frac{1}{\sigma\lambda})$$

et

$$\sigma > 0 \quad \text{et} \quad \lambda > 0$$

donc :

$$K < - 2$$

donc

$$d_1 < - 1$$

et

$$d_2 = K - \frac{1}{d_1} < - 1$$

Si l'on suppose que pour un certain i l'on ait :

$$d_i < - 1$$

alors on aura aussi :

$$d_{i+1} = K - \frac{1}{d_i} < - 1$$

donc :

$$d_i < - 1 \quad \forall \; i$$

donc :

$$|d_{i+1}| = |K| - \frac{1}{|d_i|}$$

On a :

$$\frac{K}{2} < d_1 = K < K$$

et donc aussi

$$\frac{K}{2} < d_2 = K - \frac{1}{d_1} < K$$

Si l'on suppose que l'on ait pour un certain i :

$$\frac{K}{2} < d_i < K$$

alors on aura aussi :

$$\frac{K}{2} < d_{i+1} = K - \frac{1}{d_i} < K$$

donc finalement

$$1 < \frac{K}{2} < d_i < K \qquad \forall\, i$$

Les d_i restent donc bornés et non nuls quand i augmente indéfiniment.

On en conclut que les éléments des matrices d'erreurs D et T restent bornés quel que soit le nombre de points selon l'axe de x.

D'où finalement :

Corollaire pratique

Dans la résolution de l'équation de la chaleur par la méthode d Crank et Nicolson,
si l'on calcule en point décimal flottant,
si l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires,
l'erreur moyenne de calcul sur les résultats d'un niveau en t est une erreur relative indépendante des valeurs de la fonction solution et sensiblement proportionnelle à l'erreur relative élémentaire commise sur chaque opération arithmétique.
Si, de plus, les conditions de bord sont nulles,
ou si le nombre de points selon l'axe des x est grand,
cette erreur relative sur les résultats est aussi sensiblement proportionnelle au nombre de pas en t calculés depuis le début.
Les facteurs de proportionnalité restent bornés quand on augmente le nombre de points selon l'axe des x.

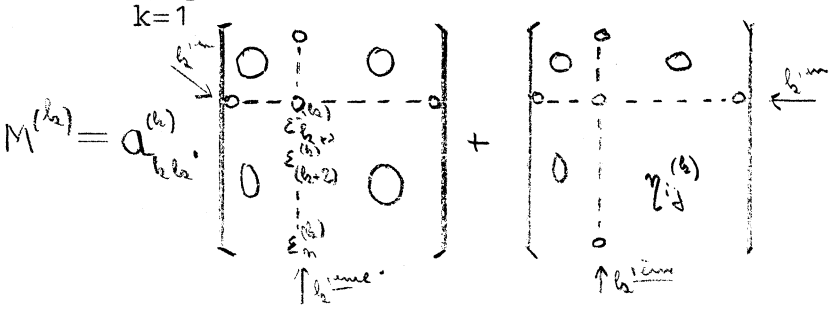
III - Erreurs de calcul en fixe dans la résolution de systèmes linéaires par l'élimination avec divisions

Si l'on résoud le système linéaire AX=Y en calculant en point décimal fixe et en utilisant la méthode générale de simple élimination de Gauss, dans sa version avec divisions, la solution effectivement obtenue peut être considérée comme la solution exacte du système (*)

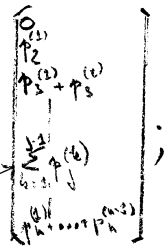
(1) $$(A + \delta A)\, X = Y + \delta Y$$

(2) $$\delta A = \sum_{k=1}^{n-1} M^{(k)} \quad ; \qquad \delta Y = \delta_1 Y + \delta_2 Y + \delta_3 Y$$

(3) $M^{(k)}$ =

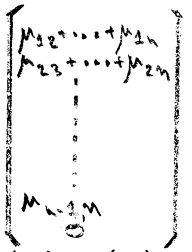


(4) $\delta_1 Y$ =



; $\delta_2 Y$ =

; $\delta_3 Y$ =



(5)

$\varepsilon_{k+i}^{(k)}$ = erreur absolue dans la division de $\bar{\rho}_{k+i}^{(k)} = \bar{a}_{k+i,k}^{(k)} / \bar{a}_{kk}^{(k)}$ + erreur

$\eta_{i,j}^{(k)}$ = erreur absolue dans la multplication de $a_{ij}^{(k+1)} = a_{ij}^{(k)}$ $- \rho_i^{(k)} \cdot \bar{a}_{k,j}^{(k)}$ + erreur

$p_i^{(k)}$ = erreur absolue dans la multiplication de $y_i^{(k+1)} = y_i^{(k)}$ $- \rho_i^{(k)} \cdot y_k^{(k)}$ + erreur

---

(*) Mr Gastinel l'a montré : (6) théorème III page 65 et explicitations pages 62, 64 et 65

ε_{ik} = erreur absolue dans la multiplication de la récurrence des sommes partielles

ε_i = erreur absolue dans la division de $\bar{y}_i=(\bar{y}^{(i)}-\bar{s}_{i,i+1})/\bar{a}^{(i)}_{ii}$

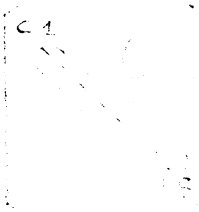
Rappelons que, dans le calcul en "fixe", on considère que les additions et les soustractions sont effectuées sans erreur.

Posons, comme plus haut :

(6) $d_i = a^{(i)}_{ii}$

Si la matrice du premier membre est de la forme :

(7)



et si l'on en tient compte selon la particularisation de la méthode exposée plus haut (*), on peut raisonner pour les erreurs de calcul fixe d'une façon analogue à ce que l'on a fait pour les erreurs de calcul flottant et on voit que les seules erreurs non nulles sont dans ce cas :

$-P_{k+1}=p^{(k)}_{k+1}$ = erreur absolue sur la division de $\bar{y}^{(k+1)}_{k+1}$

(8) $= \bar{y}^{(k)}_{k+1} - \bar{y}^{(k)}_{k}/d_k$ + erreur

$\varepsilon_{k+1}=\varepsilon^{(k)}_{k+1,k+1}$ = erreur absolue sur la division de $d_{k+1}=c-1/d_k$ + erreur

ε_i = erreur absolue sur la division de $\bar{y}_i = (y_i - \bar{s}_{i+1})/d_i$

Posons comme plus haut :

(9) E_{ij} = matrice carrée d'ordre n ayant un 1 en position (i,j) et des 0 partant ailleurs.

(*) page 43

En reportant (8) dans (3) et (4), on obtient :

(10) $$M^{(k)} = \eta_{k+1} \cdot E_{k+1,k+1} \; ; \; \delta_1 Y = \begin{bmatrix} 0 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{bmatrix} \; ; \; \delta_2 Y = \begin{bmatrix} \gamma_1 . d_1 \\ \gamma_2 . d_2 \\ \vdots \\ \gamma_n . d_n \end{bmatrix} \; ; \; \delta_3 Y = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

En reportant (10) dans (2), il vient :

(11) $$\delta A = \begin{bmatrix} 0 & & \\ \eta_2 . d_2 & \ddots & 0 \\ & \ddots & \eta_n . d_n \end{bmatrix} \; ; \quad \delta Y = \begin{bmatrix} 0 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} + \begin{bmatrix} \gamma_1 . d_1 \\ \gamma_2 . d_2 \\ \vdots \\ \gamma_n . d_n \end{bmatrix}$$

d_i défini en (6) ; η_i, p_i et γ_i définis en (8)

Donc :

Théorème 1

Dans la résolution du système linéaire A.X = Y où $A = \begin{bmatrix} c & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & c \end{bmatrix}$
si l'on calcule en point décimal fixe
si l'on utilise l'élimination avec divisions particularisée,
la solution réellement obtenue peut être considérée comme
la solution exacte de

$$(A + \delta A) . X = Y + \delta Y$$

δA et δY étant définis en (11)

Si maintenant on suppose de plus que l'erreur absolue commise sur chaque division est toujours de l'ordre de ε, (11) devient :

$$\delta A = \varepsilon . \begin{bmatrix} 0 & & \\ d_2 & \ddots & 0 \\ & \ddots & d_n \end{bmatrix} \; ; \quad \delta Y = \varepsilon . \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \varepsilon . \begin{bmatrix} d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}$$

En raison des approximations que l'on a faites, on ne change à peu près pas δA et δY en modifiant un peu leur première ligne il vient alors :

$$\delta A = \varepsilon . \begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix} ; \quad \delta Y = \varepsilon . \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \varepsilon . \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

c'est-à-dire :

$$\delta A = \varepsilon . D \quad ; \quad \delta Y = \varepsilon . (I+D) . \underline{z}$$

$$D = \begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix} ; \quad \underline{z} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Théorème 2

Dans la résolution du système linéaire $A . X = Y$ où $A = \begin{bmatrix} \varepsilon & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & \varepsilon \end{bmatrix}$
si l'on calcule en point décimal fixe;
si l'on utilise l'élimination avec divisions particularisée
si l'on considère que l'on fait sur chaque division une erreur absolue ε,
la solution réellement obtenue peut être considérée comme la solution exacte de

$$(A + \varepsilon . D)\ X = Y + \varepsilon\ (I+D) . \underline{z}$$

avec

$$D = \begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{bmatrix} ; \quad \underline{z} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

d_i étant le $i^{ème}$ élément diagonal de A après triangularisation.

On remarque que les erreurs sur les multiplications n'interviennent pas.

IV - Erreurs de calcul en "fixe" dans la méthode de Crank et Nicolson quand on utilise l'élimination avec divisions

On a vu (*) que la méthode de Crank et Nicolson exige à chaque pas en t la résolution du système linéaire :

(1) $$A.U_{p+1} = B.U_p + V_p$$

Le second membre exige les calculs

$[K''.(K'x_1+x_2+ug_p)-ug_{p+1}]$ et $[K''.(K'x_n+x_{n+1}+ud_p)-ud_{p+1}]$ pour la 1e composante et la dernière
$[K''.(x_{i-1}+K'.x_i+x_{i+1})]$ pour la $i^{ème}$ composante.

Si l'on admet que l'erreur absolue sur chaque produit élémentaire est de l'ordre de η, les erreurs absolues sur les calculs ci-dessus sont $\eta.(1+K'')$ et le second membre de (1) est remplacé en fait par :

(2) $$B.U_p + V_p + \eta.(1+K'').Z \quad \text{avec} \quad Z = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Donc, si l'on applique le théorème 2 du paragraphe précédent, on a :

(*) page 40

Théorème

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,
si l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires,
si l'on calcule en point décimal fixe,
si l'on fait sur chaque multiplication une erreur absolue de l'ordre de η
et sur chaque division une erreur absolue de l'ordre de ε,
si l'on estime négligeables les erreurs sur les additions et les soustractions,
si l'on appelle U_p le vecteur des résultats théoriques de l'équation aux différences aux noeuds du $p^{ième}$ niveau en t
si l'on appelle $\overline{U}_p$ le vecteur analogue à U_p mais réellement obtenu,
si U_0 est le vecteur des conditions initiales et V_p celui des $p^{ièmes}$ conditions de bord,

On a :

$$AU_{p+1} = BU_p + V_p$$

$$(A+\varepsilon D)\,\overline{U}_{p+1} = B\overline{U}_p + V_p + [\eta(1+K'')I + \varepsilon(I+D)]\,\xi$$

$$U_p = (A^{-1}B)^p U_0 + \sum_{j=1}^{p} (A^{-1}B)^{j-1} A^{-1} V_{p-j}$$

$$\overline{U}_p = [(A+\varepsilon D)^{-1}B]^p U_0 + \sum_{j=1}^{p} [(A+\varepsilon D)^{-1}B]^{j-1} (A+\varepsilon D)^{-1} V_{p-j}$$
$$+ \left[\sum_{j=1}^{p} [(A+\varepsilon D)^{-1}B]^{j-1}\right] (A+\varepsilon D)^{-1} [\eta(1+K'')I + \varepsilon(I+D)]\,\xi$$

avec

$$A = \begin{bmatrix} K & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K \end{bmatrix} ; \qquad B = K'' \cdot \begin{bmatrix} K' & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K' \end{bmatrix}$$

$K = -\left(2 + \frac{1}{\sigma\lambda}\right)$; $K' = -\left(2 - \frac{1}{(1-\sigma)\lambda}\right)$; $K'' = \frac{\sigma - 1}{\sigma}$

$= \frac{\text{pas en t}}{(\text{pas en x})^2}$; $0 \leq \sigma \leq 1$: caractéristique de la méthode

$$D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{bmatrix} ; \quad Z = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

d_i étant le $i^{ème}$ élément diagonal de A après triangularisation

Pour tirer des conclusions pratiques de ce théorème, on peut comparer U_p et $\overline{U}_p$. et faire les remarques suivantes :

- le premier terme de la formule donnant $\overline{U}_p$ introduit une erreur relative proportionnelle à ε et à p mais moins importante que l'erreur relative trouvée dans le cas du calcul en "flottant"

- pour les mêmes raisons que dans le cas du calcul en "flottant" (*) l'erreur introduite par le 2e terme de la formule donnant $\overline{U}_p$ peut être considérée comme négligeable si les conditions de bord sont nulles ou si le nombre de points selon l'axe des x est grand

- il apparaît dans le calcul en "fixe" de U_p un 3e terme qui n'a pas d'analogue dans le cas du calcul en "flottant" ; ce terme est entièrement un terme d'erreur et introduit une erreur absolue supplémentaire à chaque pas en t ; cette erreur absolue ajoutée à chaque pas en t peut être considérée comme à peu près constante quand p varie si l'on admet que le coefficient matriciel $\left[(A + \varepsilon D)^{-1} . B\right]^{j-1}$ ne dépend pas trop de j : cette hypothèse est vérifiée si les conditions de stabilité de la méthode de Crank et Nicolson le sont, car alors les valeurs

(*) page 63

propres de la matrice $(A+\varepsilon D)^{-1}.B$ ont un module ≤ 1. (*) ; cette erreur absolue ajoutée à chaque pas est proportionnelle aux erreurs absolues élémentaires ε et η

On voit que l'erreur totale de calcul est somme d'une erreur relative et d'une erreur absolue, toutes deux proportionnelles à ε et η ; l'erreur relative peut évidemment être considérée comme une erreur absolue proportionnelle aux valeurs de la fonction-solution ; l'erreur absolue due au 3e terme est par contre indépendante des valeurs de la fonction ; l'erreur totale de calcul est donc une fonction croissante mais non linéaire des valeurs de la fonction solution.

Au point de vue ordre de grandeur de l'erreur, on remarque que les matrices d'erreurs ont pour éléments les d_i qui sont bornés comme on l'a vu plus haut (**)

Corollaire pratique :

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson
si l'on calcule en point décimal fixe
si l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires,
si l'on est dans un cas où la méthode est stable, (***)
si les conditions de bord sont nulles,
ou si le nombre de points selon l'axe des x est grand,
l'erreur moyenne de calcul sur les résultats d'un niveau en t est une erreur absolue proportionnelle aux erreurs absolues élémentaires commises sur les multiplications et les divisions ; cette erreur absolue sur les résultats est aussi proportionnelle au nombre de pas en t calculés depuis le début ;
enfin cette erreur absolue est une fonction croissante (mais non linéaire) des valeurs de la fonction solution
Les facteurs de proportionnalité restent bornés quand on augmente le nombre de points selon l'axe des x.

* la question de l'influence des erreurs de calcul sur la stabilité de la méthode de Crank et Nicolson est étudiée plus loin § IX

(**) page 65

(***) conditions de stabilité page 53

V - Erreurs de calcul en flottant dans la résolution de systèmes linéaires par l'élimination sans division

On a vu (*) que, dans le cas général de la résolution d'un système linéaire AX = Y, l'algorithme de cette version de la méthode de simple élimination de Gauss peut s'exprimer ainsi :

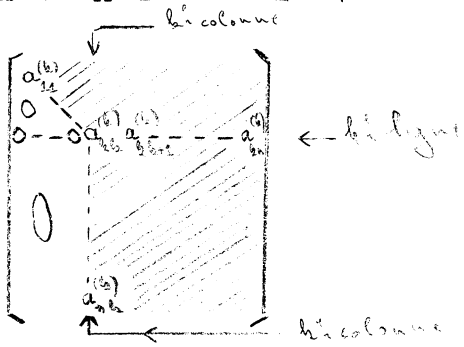
(1) $$a_{ij}^{(k+1)} = a_{ij}^{(k)} \cdot a_{kk}^{(k)} - a_{kj}^{(k)} \cdot a_{ik}^{(k)} \; ; \; k=1,\ldots,n-1 \; ; \; i=k+1,\ldots,n \; ; \; j=k+1,\ldots,n+1 \; ;$$

La triangularisation de la matrice A peut être considérée dans cette version comme une récurrence de terme initial A_1 (matrice donnée) et de terme final A_n (matrice triangularisée) ; cette récurrence peut s'exprimer matricellement ainsi :

(2) $$A_{k+1} = J_k \cdot A_k$$

(3) $$A_n = J_{n-1} \cdot J_{n-2} \cdots J_2 \cdot J_1 \cdot A$$

(4) $A_k =$



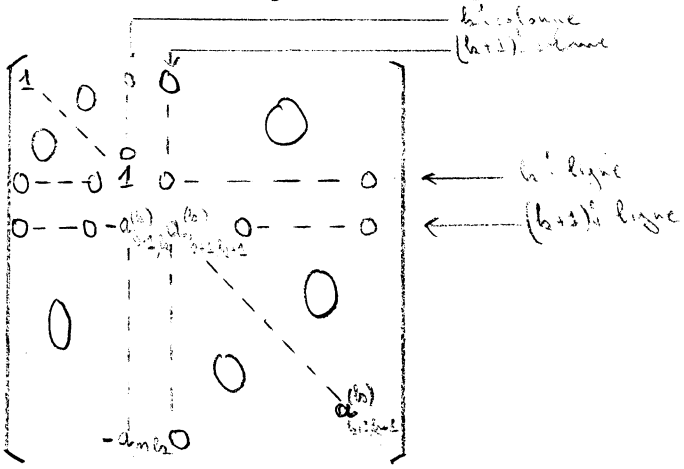
(la partie hachurée représente les zones non nulles de la matrice A_k)

(5) $$J_k = I + \sum_{i=k+1}^{n} \left[(a_{kk}^{(k)} - 1) \cdot E_{ii} - a_{ik}^{(k)} E_{ik} \right]$$

(*) page 44

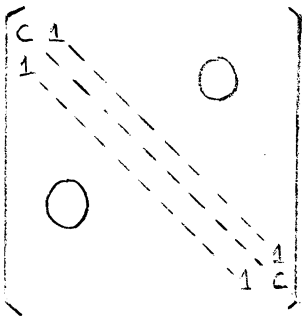
(6) E_{ij} = matrice carrée d'ordre n ayant 1 en position (i,j), des 0 partout ailleurs

(7) $A_k =$



On se limite au type de matrice premier membre tri-diagonale qui intervient dans la méthode de Crank et Nicolson :

(8) $A =$



On peut alors particulariser l'algorithme de l'élimination sans division comme on l'a vu plus haut (*)
Dans ce cas, la triangularisation de la matrice A peut s'exprimer par la récurrence :

(9) $$A_{k+1} = J_k \cdot A_k \quad ; \quad A_n = J_{n-1} \cdot J_{n-2} \cdots J_2 \cdot J_1 \cdot A$$

A_n = matrice triangularisée

A = matrice donnée

(10) $$J_k = I + (a_{kk}^{(k)} - 1) \cdot E_{k+1,k+1} - E_{k+1,k} \text{ avec } E_{ij} \text{ définie en (6)}$$

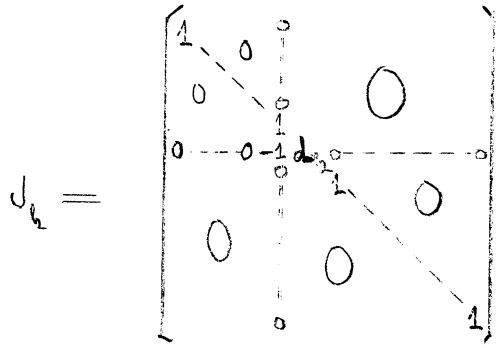
Posons :

(11) $$d_k = a_{kk}^{(k)}$$

(*) page 45

on a

(12)



Au point de vue pratique numérique, la triangularisation de la matrice premier membre A se fait en exécutant la récurrence :

(13) $$d_{i+1} = c.d_i - d_{i-1}$$

cette récurrence fournit les termes diagonaux de la matrice triangularisée mais aussi les termes de position (i,i+1) de cette matrice triangularisée car le calcul montre que, dans cet algorithme et avec ce type de matrice, l'on a

(14) $$a^{(k+1)}_{k+1,k+2} = a^{(k)}_{kk}$$

Pour étudier l'influence des erreurs de calcul, on suit les hypothèses faites plus haut (*) sur les calculs enpoint décimal flottant et l'on pose :

(15) ε_{k+1} = erreur relative sur le calcul $d_{k+1}=cd_k-d_{k-1}$

+ erreur

On peut donc considérer que la matrice effectivement obtenue par l'exécution en machine de l'algorithme de

---

(*) page 4

triangularisation satisfait exactement la récurrence :

(16) $$A_{k+1} = J_k\, A_k + M'_k$$

avec

(17) $$M'_k = \varepsilon_{k+1} \cdot d_{k+1} \cdot E_{k+1,k+1} + \varepsilon_k \cdot d_k \cdot E_{k+1,k+2}$$

(avec E_{ij} défini en (6))

Si l'on pose

(18) $$M_k = \frac{1}{d_k} M'_k$$

on constate que

(19) $$J_k \cdot M_k = M'_k \text{ et } J_k \cdot M_{k+p} = M_{k+p} \quad \forall\, p \text{ entier} > 0$$

En tenant compte de (18) dans (16), on obtient :

$$\overline{A_2} = \overline{J_1}\, A_1 + M'_1 = \overline{J_1}(A_1 + M_1) \quad \text{car } \overline{A_1} = A_1 = A$$

$$\overline{A_3} = \overline{J_2}\,\overline{A_2} + M'_2 = J_2 \cdot (\overline{A_2} + M_2) = \overline{J_2} \cdot (\overline{J_1} \cdot (A_1 + M_1) + M_2) = \overline{J_2} \cdot \overline{J_1}(A + M_1 + M_2$$

Raisonnons par récurrence et supposons que l'on ait :

$$\overline{A_k} = \overline{J_{k-1}}\,\overline{J_{k-2}} \cdots \overline{J_1} \cdot (A + M_1 + M_2 + \ldots + M_{k-1})$$

alors l'on a aussi

$$\overline{A_{k+1}} = \overline{J_k}\,\overline{A_k} + M'_k = \overline{J_k}(\overline{A_k} + M_k) = \overline{J_k}(\overline{J_{k-1}} \cdots \overline{J_1}(A + M_1 + \ldots + M_{k-1}) + M_k)$$

donc finalement :

(20) $$A_n = \overline{J_{n-1}} \cdot \overline{J_{n-2}} \cdot \ldots \cdot \overline{J_2} \cdot \overline{J_1} \cdot (A + M_1 + \ldots + M_{n-1})$$

On remarque que les $\overline{J_k}$ ne diffèrent des J_k qu'en ce qu'ils comportent l'élément $\overline{d_k}$ à la place de l'élément d_k : la multiplication par $\overline{J_k}$ a donc sur $\overline{A_k}$ le même effet que la multiplication par J_k sur A_k.

(20) exprime donc que la matrice $\overline{A_n}$ correspond à la triangularisation exacte de la matrice

$$A + \sum_{i=1}^{n-1} M_i$$

Raisonnons de même pour le second membre. On a :

$$Y_{k+1} = J_k \cdot Y_k$$

Dans notre cas où A est du type indiqué en (8), J_k a la valeur donnée en (10) et (12)
Soit

ε'_{k+1} = erreur relative dans le calcul $\overline{y}^{(k+1)}_{k+1} =$

$= C.\overline{y}^{(k)}_{k+1} - \overline{y}^{(k)}_{k}$ + erreur

On a

$$\overline{Y_{k+1}} = \overline{J_k}\ \overline{Y_k} + \overline{Z'_k}$$

(21)
$$\overline{Z'_k} = \begin{pmatrix} 0 \\ \vdots \\ \overline{a}^{(k)}_{k+1} \cdot \varepsilon_{k+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

On pose :

(22)
$$\overline{Z_k} = \frac{1}{d_k} \cdot \overline{Z'_k}$$

On constate que :

$$J_k \cdot \overline{Z_k} = \overline{Z'_k} \text{ et } J_k \cdot \overline{Z_{k+p}} = \overline{Z_{k+p}} \qquad \forall\, p \text{ entier} > 0$$

Donc, de manière analogue à (20), on a :

(23) $$\overline{Y_n} = \overline{J_{n-1}}\ \overline{J_{n-2}} \cdots \overline{J_2} \cdot \overline{J_1}(Y + \varepsilon_1 + \cdots + \varepsilon_{n-1})$$

(23) exprime que Y_n est le vecteur issu exactement du vecteur $(Y + \sum_{i=1}^{n-1} \varepsilon_i)$ après qu'on lui ait appliqué les transformations qui triangularisent exactement la matrice $(A + \sum_{i=1}^{n-1} M_i)$

Les résultats (20) et (23) donnent :

Théorème 1

Dans la triangularisation de la matrice A d'un système linéaire où

$$A = \begin{pmatrix} c & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & c \end{pmatrix}$$

si l'on utilise l'élimination sans division,
si l'on calcule en point décimal flottant,
si l'on commet les erreurs relatives élémentaires ε_i sur $d_i = c.d_{i-1} - d_{i-2}$ et ε'_i sur $y_i^{(i)} = c.y_i^{(i-1)} - y_{i-1}^{(i-1)}$
le système triangulaire réellement obtenu peut être considéré comme résultant de la triangularisation exacte du système :

$$(A + \sum_{k=1}^{n-1} M_k).X = Y + \sum_{k=1}^{n-1} \varepsilon_k$$

M_k étant définie en (18) et (17) et ε_k en (22) et (21).

Pour la résolution du système triangulaire, encore appelée "Retour Arrière", on a vu (*) que les formules exactes sont :

(24) $$x_n = y_n^{(n)}/d_n$$

(25) $$x_i = (y_i^{(i)} - d_{i-1}x_{i+1})Y\ d_i \qquad \forall\ i \text{ entier de } n-1 \text{ à } 1$$

Dans le calcul réel en point décimal flottant, on peut considérer que l'on commet les erreurs suivantes :

(26) ε''_i = erreur relative sur le calcul $y_i^{(i)} - d_{i-1} \cdot x_{i+1}$

(27) η_i = erreur relative sur la division par d_i

Les valeurs effectivement obtenues sont alors :

(28) $$\bar{x}_n = (\bar{y}_n^{(n)}/\bar{d}_n)(1+\eta_n) \simeq (\bar{y}_n^{(n)} + \eta_n \cdot \bar{x}_n)/d_n$$

$$\bar{x}_i = \left[(\bar{y}_i^{(i)} - \bar{d}_{i-1} \cdot \bar{x}_{i+1}).(1+\varepsilon''_i)/\bar{d}_i\right](1+\eta_i)$$

$$\bar{x}_i \simeq \left[(\bar{y}_i^{(i)} - \bar{d}_{i-1} \cdot \bar{x}_{i+1})/\bar{d}_i\right] \cdot (1+\varepsilon''_i + \eta_i)$$

(29) $$\bar{x}_i \simeq \left[\bar{y}_i^{(i)} + (\varepsilon''_i + \eta_i).\bar{d}_i.\bar{x}_i - \bar{d}_{i-1} \cdot \bar{x}_{i+1}\right]/\bar{d}_i$$

On voit donc que le calcul effectif avec erreurs peut s'exprimer comme étant l'exécution exacte de l'algorithme sur un vecteur second membre auquel on aurait ajouté $\delta_2 Y$:

(30) $$\delta_2 Y = \begin{bmatrix} (\varepsilon''_1 + \eta_1).\bar{d}_1.\bar{x}_1 \\ (\varepsilon''_2 + \eta_2).\bar{d}_2.\bar{x}_2 \\ \\ \\ \\ (\varepsilon''_{n-1} + \eta_{n-1}).\bar{d}_{n-1}.\bar{x}_{n-1} \\ \eta_n.\bar{d}_n.\bar{x}_n \end{bmatrix}$$

Donc :

(*) page 45

Théorème 2

Dans la résolution du système linéaire A'X=Y où A' =
si l'on calcule en point décimal flottant,
si l'on commet les erreurs relatives élémentaires définies en (26) et (27),
la solution effectivement obtenue peut être considérée comme la solution exacte du système

$$A' X = Y + \delta_2 Y$$

$\delta_2 Y$ étant le vecteur défini en (30)

En rapprochant les théorèmes 1 et 2, on obtient :

Théorème 3

Dans la résolution du système linéaire AX=Y où A=
si l'on calcule en point décimal flottant,
si l'on utilise l'élimination sans division adaptée à la matric
si l'on commet les erreurs relatives ε_i sur le calcul

$$\overline{d_i} = (c \cdot \overline{d_{i-1}} - \overline{d_{i-2}}) \cdot (1 + \varepsilon_i)$$

ε'_i

$$\overline{y}_i^{(i)} = (c \cdot \overline{y}_i^{(i-1)} - \overline{y}_{i-1}^{(i-1)}) \ (1 + \varepsilon'_i)$$

ε''_i

$$\overline{b_i} = (\overline{y_i} - \overline{d_{i-1}} \cdot \overline{x_{i+1}}) \ (1 + \varepsilon''_i)$$

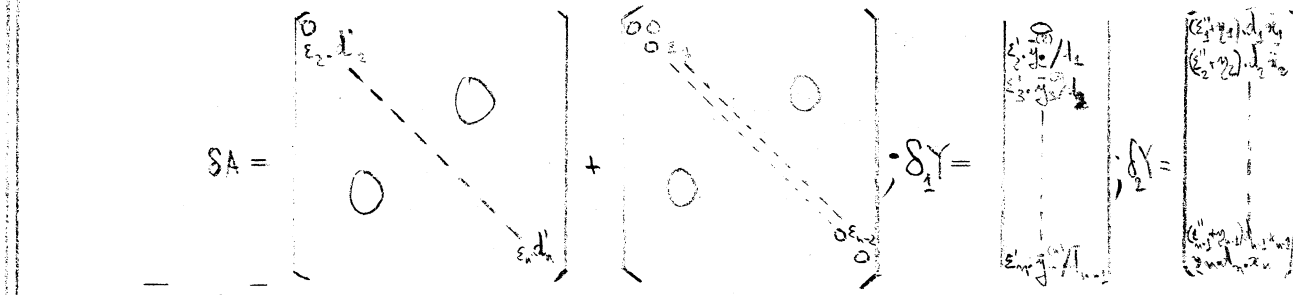
η_i

$$\overline{x_i} = (\overline{b_i} / \overline{d_i}) \ (1 + \eta_i)$$

La solution approchée effectivement obtenue peut être considéré comme la solution exacte du système :

$$(A + \delta A) X = Y + \delta_1 Y + \delta_2 Y$$

avec :



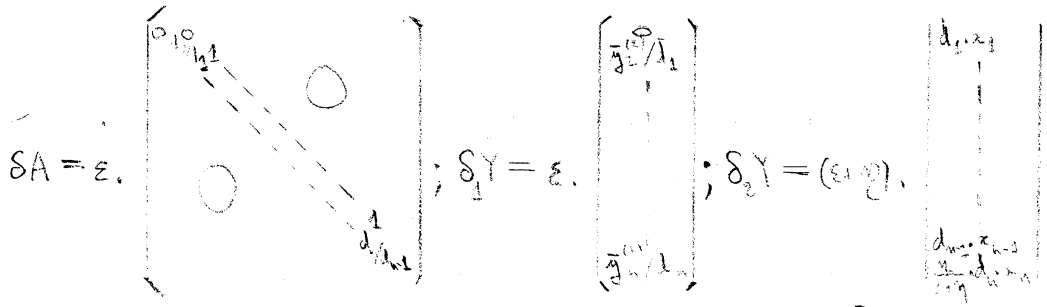
$$d'_i = \bar{d}_i \,/\, \bar{d}_{i-1}$$

Si maintenant on suppose de plus que

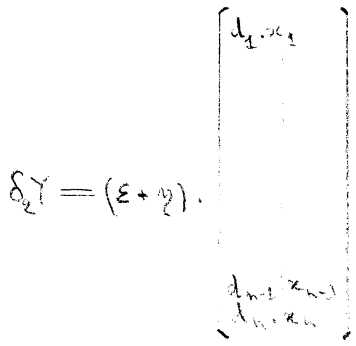
- toutes les erreurs relatives ε_i, ε'_i et ε''_i sont de l'ordre de ε,

- toutes les erreurs relatives η_i sont de l'ordre de η

on a :



En raison des approximations faites, on change peu $\delta_2 Y$ en changeant sa dernière ligne ; on peut donc écrire



Donc :

31) $$\delta A = \varepsilon . D' . A' \; ; \; \delta_1 Y = \varepsilon . D' . Y' \; ; \; \delta_2 Y = (\varepsilon + \eta) . D . X$$

avec

(32)

$$D' = \begin{bmatrix} \bar{d}_1 & & 0 \\ & \ddots & \\ 0 & & \bar{d}_{n-1} \end{bmatrix} ; A' = \begin{bmatrix} 1 & l_1 & & 0 \\ & 1 & \ddots & \\ & & \ddots & l_{n-1} \\ 0 & & & 1 \end{bmatrix} ; Y' = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} ; D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$

On remarque que A' et Y' sont les transformés respectivement de la matrice A et du vecteur Y par les transformations qui triangularisent la matrice A. Donc

$$A' = T.A \text{ et } Y' = T.Y \tag{33}$$

avec

$$T = \prod_{k=n-1}^{1} J_k \text{ et } J_k \text{ défini en (12) et (10)} \tag{34}$$

On remarque que T est unematrice triangulaire inférieure "plein et qu'elle comporte les $\bar{d}_i$ sur sa diagonale.

En tenant compte de (31) et (33), la formule du théorème 3 devient :

$$(A + \varepsilon.D'.T.A).X = Y + \varepsilon.D'.T.Y + (\varepsilon + \eta).D.X$$

On met en facteur en tenant compte de ce que le signe de ε et η n'est pas défini

$$\left[(I + \varepsilon.D'.T)A + (\varepsilon + \eta).D\right] . X = (I + \varepsilon.D'.T).Y$$

Donc :

Théorème 4

Dans la résolution du système linéaire AX = Y où A = 
si l'on utilise l'élimination sans division avec adaptation à la structure de la matrice A,
si l'on calcule en point décimal flottant,
si l'on fait des erreurs relatives de l'ordre de ε sur les calculs du type $a \pm b \cdot c$,
et des erreurs relatives de l'ordre de η sur les divisions,
la solution réellement obtenue peut être considérée comme la solution exacte du système :

$$\left[(I+\varepsilon.D'.T).A+(\varepsilon+\eta).D\right].X = (I+\varepsilon D'.T).Y$$

avec

$$D' = \begin{bmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_n \end{bmatrix}; \qquad D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix}$$

d_i = $i^{ème}$ élément diagonal de A après sa triangularisation
T = matrice exprimant les transformations qui triangularisent A
T est une matrice triangulaire inférieure "pleine" avec les d_i sur sa diagonale.

VI - Erreurs de calcul en "flottant" dans la méthode de Crank et Nicolson quand on utilise l'élimination sans division

On a vu (*) que la méthode de Crank et Nicolson exige à chaque pas en t la résolution du système linéaire

(1) $$A.U_{p+1} = B.U_p + V_p$$

Pour les mêmes raisons que plus haut (**), le second membre est remplacé en fait par

$$(1 + 2\varepsilon).(B.U_p + V_p)$$

Donc, si l'on applique le théorème 4 du paragraphe précédent, on a :

Théorème

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,
si l'on utilise l'élimination sans division pour la résolution des systèmes linéaires,
si l'on calcule en point décimal flottant,
si l'on commet sur chaque calcul du type a+bc une erreur relative de l'ordre de ε,
et sur chaque division une erreur relative de l'ordre de η,
si l'on appelle U_p le vecteur des résultats théoriques de l'équation aux différences aux noeuds du $p^{ième}$ niveau en t,
si l'on appelle $\overline{U}_p$ le vecteur analogue à U_p mais effectivement obtenu, si U_0 est le vecteur des conditions initiales,
et V_p le vecteur des conditions de bord au $p^{ième}$ niveau en t,
on a :

(*) page 40
(**) page 61

$$A.U_{p+1} = B.U_p + V_p$$

$$\left[(I+\varepsilon D'.T)A+(\varepsilon+\eta).D\right].\bar{U}_{p+1} = (1+2\varepsilon).(I+\varepsilon D'.T).(B\bar{U}_p+V_p)$$

$$U_p = (A^{-1}B)^p.U_0 + \sum_{j=1}^{p}(A^{-1}B)^{j-1}.A^{-1}.V_{p-j}$$

$$\bar{U}_p = \left\{\left[(I+\varepsilon.D'.T)A+(\varepsilon+\eta).D\right]^{-1}.(1+2\varepsilon).(I+\varepsilon D'.T).B\right\}^p.U_0$$

$$+\sum_{j=1}^{p}\left\{\left[(I+\varepsilon D'.T)A+(\varepsilon+\eta)D\right]^{-1}(1+2\varepsilon)(I+\varepsilon D'T)B\right\}^{j-1}.\left[(I+\varepsilon.D'.T)A+(\varepsilon+\eta).D\right]^{-1}$$

$$.V_{p-j}$$

$$A = \begin{bmatrix} K & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K \end{bmatrix}; \qquad B = K''.\begin{bmatrix} K' & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K' \end{bmatrix}$$

$$K = -\left(2+\frac{1}{\nabla.k}\right) \quad ; \quad K' = -\left(2-\frac{1}{(1-\nabla).k}\right) \quad ; \quad K'' = \frac{\nabla-1}{\nabla}$$

$k = \dfrac{\text{pas en } t}{(\text{pas en } x)^2}$; $0 \leq \nabla \leq 1$: caractéristique de la méthode

$$D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}; \qquad D' = \begin{bmatrix} 0 & & \\ & \ddots & \\ & & 1/d_{n-1} \end{bmatrix}$$

d_i = $i^{ème}$ élément diagonal de A après sa triangularisation

T = matrice exprimant les transformations qui triangularisent A

T est une matrice triangulaire inférieure "pleine" avec les d_i sur sa diagonale.

Pour tirer les conclusions pratiques de ce théorème, on peut comparer les pertubations apportées aux formules théoriques par les erreurs dans les 2 cas du calcul en point décimal flottant soit par l'élimination sans division (cas étudié ici), soit par l'élimination avec divisions (cas étudié plus haut) (*

On voit que la forme des perturbations apportées par les erreurs est assez analogue dans les 2 cas.

Pour comparer l'ordre de grandeur de ces perturbations, il faut comparer les éléments des matrices d'erreur.

Pour cela, il faut ici évaluer les d_i de l'élimination sans division. Ceux-ci sont définis par la récurrence :

$$d_{i+1} = K.\ d_i - d_{i-1} \ ;\ d_0 = 1 \ ;\ d_1 = K$$

on voit que :

$$\frac{d_{i+1}}{d_i} = K - \frac{1}{d_i/d_{i-1}} \quad \text{avec} \quad \frac{d_1}{d_0} = K$$

Donc la suite :

$$\frac{d_i}{d_{i-1}}$$

est exactement la même que la suite des éléments diagonaux de la méthode avec divisions (**): donc d'après ce que l'on a vu plus haut, on a ici :

$$1 < \left|\frac{K}{2}\right| < \left|\frac{d_i}{d_{i-1}}\right| \qquad \forall\, i$$

Donc ici :

$$|d_n| > |K| \cdot \left|\frac{K}{2}\right|^n$$

(*) page 61 à 65

(**) page 64

comme

$$|K| > 2$$

on voit que d_n n'est pas borné quand n augmente.

On en conclut que la matrice D introduit un terme d'erreur très grand et non borné quand on augmente le nombre de pas selon l'axe des x.

Dans le cas de l'élimination avec divisions, les éléments des matrices d'erreur restaient au contraire bornés.

On résume ces remarques ainsi :

Corollaire pratique :

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,
si l'on calcule en point décimal flottant,
si l'on utilise l'élimination sans division pour la résolution des systèmes linéaires,
l'erreur de calcul a la même forme que dans le cas où l'on calcule en point décimal flottant par l'élimination avec division mais les facteurs de proportionnalité sont beaucoup plus grands et ne sont pas bornés quand on augmente le nombre de points selon l'axe des x.

VII - Erreurs de calcul en "fixe" dans la résolution de systèmes linéaires par l'élimination sans division

On peut raisonner comme plus haut dans le cas de la même méthode en calcul en point décimal flottant (*). On reprend les mêmes notations sans les définir à nouveau.

L'algorithme théorique de triangularisation s'exprime par la récurrence matricielle

(1) $$A_n = J_{n-1} \cdot J_{n-2} \cdots J_2 \cdot J_1 \cdot A \; ; \; A_{k+1} = J_k \cdot A_k \; ; \; A_1 = A$$

Au point de vue numérique, il suffit d'exécuter la récurrence scalaire :

(2) $$d_{i+1} = c\, d_i - d_{i-1}$$

En suivant les hypothèses que nous avons faites sur les erreurs de calcul en point décimal fixe,(**), on peut dire que l'exécution effective de cette récurrence donne :

(3) $$\bar{d}_{i+1} = c \cdot \bar{d}_i - d_{i-1} + \varepsilon_{i+1}$$

ce qui s'exprime matriciellement ainsi :

(4) $$\bar{A}_{k+1} = \bar{J}_k \cdot \bar{A}_k + \bar{M'}_k$$

M'_k ayant une valeur différente de sa valeur dans le cas "flottant" :

(5) $$M'_k = \varepsilon_{k+1} \cdot E_{k+1,k+1} + \varepsilon_k \cdot E_{k+1,k+2}$$

avec

(6) E_{ij} = matrice carrée d'ordre n ayant un 1 en position

(*) pages 74 et 75

(**) page 4

(i,j), des 0 ailleurs, M'_k a la même structure matricielle que la matrice correspondante du cas "flottant" on a donc aussi ici :

$$\overline{A}_n = \overline{J}_{n-1} \cdot \overline{J}_{n-2} \cdots \overline{J}_1 \cdot (A + \sum_{k=1}^{n-1} M_k) \quad (7)$$

avec

$$M_k = \frac{1}{d_k} M'_k \quad (8)$$

De même, pour l'effet sur le second membre des transformations qui triangularisent la matrice du premier membre, l'algorithme théorique s'exprime par la récurrence matricielle :

$$Y_n = J_{n-1} J_{n-2} \cdots J_1 \cdot Y \; ; \; Y_{k+1} = Y \cdot J_k \cdot Y_k \; ; \; Y_1 = Y \quad (9)$$

Au point de vue numérique, on exécute la récurrence scalaire :

$$y_{i+1}^{(i+1)} = y_{i+1}^{(i)} \cdot d_i - y_i^{(i)} \quad (10)$$

L'exécution effective de cette récurrence donne :

$$\overline{y}_{i+1}^{(i+1)} = \overline{d}_i \; \overline{y}_{i+1}^{(i)} - \overline{y}_i^{(i)} + \mathcal{E}'_{i+1} \quad (11)$$

ce qui s'exprime matriciellement ainsi :

$$\overline{Y}_{k+1} = \overline{J}_k \cdot \overline{Y}_k + \overline{\mathcal{Z}}'_k \quad (12)$$

avec

$$\overline{\mathcal{Z}}'_k = \mathcal{E}'_{k+1} \cdot e_{k+1} \quad (13)$$

où

$$e_\lambda = \begin{vmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \\ 0 \end{vmatrix} \leftarrow \lambda^{\text{ième}} \text{ composante} \quad (14)$$

Ce $\mathcal{Z}'_k$ a une valeur différente du vecteur correspondant du cas du calcul en flottant, mais sa forme est la même; donc on peut écrire dans le cas "fixe" comme dans le cas "flottant" :

$$\overline{Y}_n = \overline{J}_{n-1} \cdot \overline{J}_{n-2} \cdots \overline{J}_2 \cdot \overline{J}_1 \cdot \left(Y + \sum_{k=1}^{n-1} \mathcal{Z}_k\right) \tag{15}$$

avec

$$\mathcal{Z}_k = \frac{1}{d_k} \cdot \mathcal{Z}'_k \tag{16}$$

D'où finalement :

Théorème 1

Dans la triangularisation de la matrice A d'un système linéaire AX = Y où A = $\begin{bmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_n \end{bmatrix}$
si l'on utilise l'élimination sans division adaptée à la forme de A,
si l'on calcule en point décimal fixe,
si l'on commet les erreurs absolues élémentaires ε_i sur $\overline{d}_i = c \cdot \overline{d}_{i-1} - \overline{d}_{i-2}$ + erreur et ε'_i sur $\overline{y}_i^{(i)} = c \cdot \overline{y}_i^{(i-1)} - \overline{y}_{i-1}^{(i-1)}$ + erreur
le système triangulaire réellement obtenu peut être considéré comme résultant de la triangularisation exacte du système

$$\left(A + \sum_{k=1}^{n-1} M_k\right) \cdot X = Y + \sum_{k=1}^{n-1} \mathcal{Z}_k$$

la matrice M_k étant définie en (8) et (5) et le vecteur $\mathcal{Z}_k$ en (16) et (13)

Pour la résolution du système triangulaire ou "retour arrière", les calculs théoriques s'expriment par la récurrence scalaire :

$$x_n = y_n / d_n \tag{17}$$

$$x_i = (y_i - d_{i-1} \cdot x_{i+1}) / d_i \qquad \forall\, i \text{ entier de } n-1 \text{ à } 1. \tag{18}$$

Dans le calcul effectif en point décimal fixe, on peut considérer que l'on commet les erreurs absolues élémentaires :

(19) ε''_i = erreur absolue sur le produit de $(y_i - d_{i-1}x_{i+1})$

(20) η_i = erreur absolue sur la division $(y_i - d_{i-1}x_{i+1})/d_i$

Les additions et les soustractions étant considérées comme effectuées sans erreur, on peut dire que l'exécution effective de la récurrence donne :

(21) $$\overline{x_n} = \overline{y_n}/\overline{d_n} + \eta_n = (\overline{y_n} + \eta_n \overline{d_n})/\overline{d_n}$$

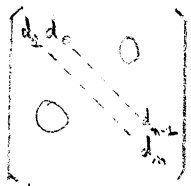
(22) $$\overline{x_i} = (\overline{y_i} - \overline{d}_{i-1}\overline{x_{i+1}} + \varepsilon''_i)/\overline{d_i} + \eta_i = (\overline{y_i} + \varepsilon''_i + \eta_i \overline{d_i} - \overline{d}_{i-1}\overline{x_{i+1}})/\overline{d_i}$$

On voit que ces erreurs ont pour effet d'ajouter au second membre le vecteur $\delta_2 Y$:

(23) $$\delta_2 Y = \begin{vmatrix} \varepsilon''_1 \\ \varepsilon''_2 \\ \vdots \\ \varepsilon''_{n-1} \\ 0 \end{vmatrix} + \begin{vmatrix} \eta_1 d_1 \\ \eta_2 d_2 \\ \vdots \\ \eta_n d_n \end{vmatrix}$$

Donc :

Théorème 2

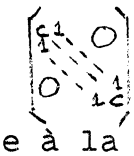
Dans la résolution du système linéaire $A'X=Y$ où $A' =$ 
si l'on calcule en point décimal fixe,
si l'on commet les erreurs absolues élémentaires ε''_i et η_i définies en (19) et (20),
la solution effectivement obtenue peut être considérée comme la solution exacte du système :

$$A' X = Y + \delta_2 Y$$

$\delta_2 Y$ étant le vecteur défini en (23).

En rapprochant les théorèmes 1 et 2, on obtient :

Théorème 3

Dans la résolution du système linéaire AX=Y où A= 
si l'on utilise l'élimination sans division adaptée à la matrice A, si l'on calcule en point décimal fixe,
si l'on commet les erreurs absolues ε_i sur les calculs

$\overline{d_i} = (c, \overline{d_{i-1}} - \overline{d_{i-2}}) + \varepsilon_i$

ε'_i sur $\overline{y_i^{(i)}} = (\overline{y_i^{(i-1)}}, \overline{d_{i-1}} - \overline{y_{i-1}^{(i-1)}}) + \varepsilon'_i$

ε''_i sur $\overline{b_i} = (\overline{y_i} - \overline{d_{i-1}} \cdot \overline{x_{i+1}}) + \varepsilon''_i$

η_i sur $\overline{x_i} = (\overline{b_i} / \overline{d_i}) + \eta_i$

la solution approchée effectivement obtenue peut être considérée comme la solution exacte du système :

$$(A + DE).X = Y + D'.H_1 + H_2 + D.H_3$$

avec :

$$D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_m \end{bmatrix} ; \quad D' = \begin{bmatrix} 0 & & \\ & \ddots & \\ & & d_{m-1} \end{bmatrix} ; \quad E = \begin{bmatrix} 0 & & & \\ \varepsilon_2 & \ddots & & \\ & \ddots & \ddots & \\ & & \varepsilon_m & 0 \end{bmatrix}$$

d_i = $i^{ème}$ élément diagonal de A après triangularisation

$$H_1 = \begin{bmatrix} 0 \\ \varepsilon'_2 \\ \vdots \\ \varepsilon'_m \end{bmatrix} ; \quad H_2 = \begin{bmatrix} \varepsilon''_1 \\ \varepsilon''_2 \\ \vdots \\ 0 \end{bmatrix} ; \quad H_3 = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix}$$

Si maintenant on suppose de plus que

- toutes les erreurs absolues $\varepsilon_i, \varepsilon'_i$ et ε''_i, qui sont des erreurs des produits, sont de l'ordre de ε

- toutes les erreurs absolues η_i, qui sont des erreurs sur des divisions, sont de l'ordre de η, on a :

$$E = \varepsilon . \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 1 \\ & & & 1 \end{bmatrix} ; \quad H_1 = \varepsilon . \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} ; \quad H_2 = \varepsilon . \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} ; \quad H_3 = \eta . \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

En raison des approximations que l'on a faites, on ne change pas H_1 et H_2 de façon appréciable en modifiant respectivement leur première et leur dernière ligne ; on peut donc écrire :

$$H_1 = \varepsilon . \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} ; \quad H_2 = \varepsilon . \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

On peut donc écrire :

$$E = \varepsilon L \; ; \quad H_1 = \varepsilon . Z \; ; \quad H_2 = \varepsilon . Z \; ; \quad H_3 = \eta . Z$$

avec

$$L = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 1 \\ & & & 1 \end{bmatrix} ; \qquad Z = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

D'où finalement :

Théorème 4

Dans la résolution du système linéaire AX=Y où A= 
si l'on utilise l'élimination sans division avec adaption à la matrice A,
si l'on calcule en point décimal fixe,
si l'on commet des erreurs absolues élémentaires de l'ordre de ε sur les multiplications, et de l'ordre de η sur les divisions,
la solution effectivement obtenue peut être considérée comme la solution exacte de

$$(A+\varepsilon.DL).X = Y + \left[\varepsilon.(I+D') + \eta.D\right].Z$$

avec

$$D = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix};\ D' = \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ 0 & & 1/d_{n-1} \end{bmatrix};\ L = \begin{bmatrix} 0 & & 0 \\ 1 & \ddots & \\ 0 & 1 & 1 \end{bmatrix};\ Z = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

d_i = $i^{ème}$ élément diagonal de A après sa triangularisation

VIII-Erreurs de calcul en "fixe" dans la méthode de Crank et Nicolson quand on utilise l'élimination sans division :

On a vu (*) que la méthode de Crank et Nicolson exige à chaque pas en t la résolution du système linéaire :

$$A \cdot U_{p+1} = B \cdot U_p + V_p$$

Lorsque l'on calcule en point décimal fixe, pour les mêmes raisons que plus haut (**), le 2nd membre est en fait remplacé par

$$B \cdot U_p + V_p + \eta(1+K'') \cdot \mathcal{Z}$$

avec

η = erreur absolue sur chaque multiplication élémentaire

$$\mathcal{Z} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Donc, si l'on applique le théorème 4 du paragraphe précédent, on obtient :

Théorème

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,
si l'on utilise l'élimination sans division pour la résolution des systèmes linéaires,
si l'on calcule en point décimal fixe,
si l'on commet des erreurs absolues élémentaires de l'ordre de ε sur les multiplications et de l'ordre de η sur les divisions,
si l'on appelle U_p le vecteur des résultats théoriques de l'équation aux différences aux noeuds du $p^{ième}$ niveau en t,
si l'on appelle $\overline{U}_p$ le vecteur analogue à U_p mais effectivement obtenu, si U_o est le vecteur des conditions initiales
et V_p le vecteur des conditions de bord au $p^{ième}$ niveau en t.

(*) page 40
(**) page 70

on a

$$A.U_{p+1} = B.U_p + V_p$$

$$(A+\varepsilon DL)\overline{U}_{p+1} = B\overline{U}_p + V_p + \left\{\varepsilon.(I+D') + \eta\left[(1+K'')I+D\right]\right\}.Z$$

$$U_p = (A^{-1}.B)^p.U_0 + \sum_{j=1}^{p}(A^{-1}B)^{j-1}.A^{-1}.V_{p-j}$$

$$\overline{U}_p = \left[(A+\varepsilon DL)^{-1}B\right]^p.U_0 + \sum_{j=1}^{p}\left[(A+\varepsilon DL)^{-1}B\right]^{j-1}.(A+\varepsilon DL)^{-1}V_{p-j}$$

$$+\left\{\sum_{j=1}^{p}\left[(A+\varepsilon DL)^{-1}B\right]^{j-1}\right\}.(A+\varepsilon DL)^{-1}.\left\{\varepsilon.(I+D')+\eta.\left[(1+K'')I+D\right]\right\}Z$$

avec

$$A = \begin{bmatrix} K & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K \end{bmatrix}; \qquad B = K''.\begin{bmatrix} K' & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & K' \end{bmatrix}$$

$$K = -\left(2 + \frac{1}{\nabla.\lambda}\right) \quad ; \quad K' = -\left(2 - \frac{1}{(1-\sigma)\lambda}\right) \quad ; \quad K'' = \frac{\nabla - 1}{\nabla}$$

$\lambda = \dfrac{\text{pas en t}}{(\text{pas en x})^2}$; $0 \leq \nabla \leq 1$: caractéristique de la méthode

$$D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}; \quad D' = \begin{bmatrix} 0 & & \\ & \ddots & \\ & & 1/d_{n-1} \end{bmatrix}; \quad L = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}; \quad Z = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

d_i = $i^{\text{ème}}$ élément diagonal de A après sa triangularisation.

Pour tirer les conclusions pratiques de ce théorème, on peut comparer les perturbations apportées aux formules théoriques par les erreurs dans les 2 cas du calcul en point décimal fixe soit par l'élimination sans division (cas étudié ici), soit par l'élimination avec divisions (cas étudié plus haut (*)).

On voit que la forme des perturbations causées par les erreurs est assez analogue dans les 2 cas.

(*) pages 70 à 73

Pour comparer l'ordre de grandeur de ces perturbations il faut comparer les éléments des matrices d'erreurs.

Dans le cas de l'utilisation de l'élimination avec divisions, on a vu (*) que tous les éléments des matrices d'erreur sont bornés.

Ici, dans le cas du calcul en point décimal fixe par l'élimination sans division, la matrice d'erreur D a sur sa diagonale principale les éléments d_i qui sont les mêmes que les d_i du calcul en point décimal flottant par la même méthode, et l'on a vu (**) que ces d_i n'étaient pas bornés.

On en conclut que la matrice d'erreur D introduit un terme d'erreur très grand et non borné quand on augmente le nombre de points selon l'axe des x.

Finalement :

Corollaire pratique :

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson ,
si l'on calcule en point décimal fixe,
si l'on utilise l'élimination sans division pour la résolution des systèmes linéaires, l'erreur de calcul a la même forme que dans le cas où l'on calcule en point décimal fixe par l'élimination avec divisions,
mais les facteurs de proportionnalité sont beaucoup plus grands, et ne sont pas bornés quand on augmente le nombre de points selon l'axe des x.

(*) page 65

(**) page 88

IX - Influence des erreurs de calcul sur la stabilité de la méthode de Crank et Nicolson

Les résultats théoriques de l'équation aux différences de la méthode satisfont à

$$U_p = (A^{-1}.B)^p . U_o + \sum_{j=1}^{p} (A^{-1}. B)^{j-1} . A^{-1} . V_{p-j}$$

On a vu vu (*) qu'une condition nécessaire et suffisan de stabilité théorique est que toutes les valeurs propres de la matrice $(A^{-1}\ B)$ aient un module inférieur ou égal à 1.

On a vu qu'en raison des erreurs de calcul, les résultats effectivement obtenus satisfont, à la place de la relatior (1), à des relations légèrement différentes qui dépendent des modes de calcul utilisés.

Un raisonnement analogue à celui fait pour l'étude de la stabilité théorique montre que, si l'on tient compte des erreurs de calcul, la stabilité réelle est liée aux valeurs propres des matrices qui "remplacent" la matrice $(A^{-1}\ B)$ dans les relations "légèrement différentes" dont on vient de parler.

Comme on a établi plus haut les "différences matriciel les" entre la matrice $(A^{-1}B)$ et les matrices qui la "remplacen et comme on s'intéresse maintenant aux valeurs propres et à leurs variations, on va utiliser le théorème suivant qui relie les variations de valeurs propres aux variations de matrice.

(*) page 48

Théorème (*)

Soit M une matrice carrée; soient U_p et V_p ses vecteurs propres nommés à droite et à gauche ; soient μ_p les valeurs propres correspondantes ; si l'on fait subir à la matrice M une petite variation exprimée par la matrice δM, il en résulte pour chaque valeur propre μ_p une variation $\delta\mu_p$; si l'on néglige les termes du 2e ordre , $\delta\mu_p$ et δM sont liés par

$$\delta\mu_p = V_p^T . \delta M . U_p$$

si de plus, M est symétrique, $U_p = V_p$ et l'on a

$$\delta\mu_p = U_p^T . \delta M . U_p$$

Dans le cas où l'on calcule en point décimal flottant et où l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires, on a vu (**) que la matrice $(A^{-1}B)$ est remplacée par la matrice

$$M' = \left[(1+2\varepsilon).(A+2\varepsilon D)^{-1}.(I+\varepsilon T).B\right]$$

On a vu (***) que A, A^{-1}, B et $(A^{-1}B)$ ont mêmes vecteurs propres et que, si μ_{A_p} , μ_{B_p} et μ_p désignent les valeurs propres respectivement de A, B et $(A^{-1}B)$ correspondant au même vecteur propre, on a :

$$\mu_p = \frac{\mu_{B_p}}{\mu_{A_p}}$$

(*) Mr Gastinel a énoncé et démontré ce théorème dans [6] pages 126 et 127

(**) pages 61 et 62

(***) pages 50 et 51

Dans notre cas, si l'on appelle μ_p les valeurs propres de M', $\delta\mu_{A_p}$ la variation de A_p causée par la variation matricielle $2\varepsilon D$ et $\delta\mu_I$ la variation de valeur propre de la matrice unité I produite par la variation matricielle εT, on a :

$$\mu'_p = \frac{(1+2\varepsilon)\ (1+\delta\mu_I)\,\mu_{B_p}}{\mu_{A_p} + \delta\mu_{A_p}}$$

La matrice A est symétrique et on a vu (*) qu'elle admet pour vecteurs propres :

$$U_p = \left(\sin\frac{p\pi}{NX-1}, \ldots, \sin\frac{(NX-2)p\pi}{NX-1}\right)^t$$

où NX est le nombre total de points selon l'axe des x dans la méthode de Crank et Nicolson : les matrices sont de dimensions (NX-2,NX-2) et les vecteurs de dimensions (NX-2)
Donc, en appliquant le théorème ci-dessus, et en normant le vecteur U_p, on obtient en écriture matricielle

$$\delta\mu_{A_p} = \frac{1}{\sqrt{\sum_{j=1}^{NX-2}\sin^2\left(\frac{jp\pi}{NX-1}\right)}}\left(\sin\frac{p\pi}{NX-1}, \ldots, \sin\frac{(NX-2)p\pi}{NX-1}\right)\cdot 2\varepsilon \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_{NX-2} \end{pmatrix} \begin{pmatrix} \sin\frac{p\pi}{NX-1} \\ \vdots \\ \sin\frac{(NX-2)p\pi}{NX-1} \end{pmatrix} \frac{1}{\sqrt{\sum_{j=1}^{NX-2}\sin^2\left(\frac{jp\pi}{NX-1}\right)}}$$

ce qui donne, en écriture scalaire :

$$\delta\mu_{A_p} = 2\varepsilon \cdot \frac{\sum_{j=1}^{NX-2} d_j \sin^2\left(\frac{jp\pi}{NX-1}\right)}{\sum_{j=1}^{NX-2} \sin^2\left(\frac{jp\pi}{NX-1}\right)}$$

(*) page 50

On voit que nous avons besoin de connaître

$$\sum_{j=1}^{n} \sin^2 \left(\frac{jp\pi}{n+1}\right) \quad \text{avec } p = 1, 2, .., n$$

$$\sin^2 \left(\frac{jp\pi}{n+1}\right) = \frac{1}{2} \cdot \left[1 - \cos \left(\frac{2\, jp\pi}{n+1}\right)\right]$$

$$\sum_{j=1}^{n} \sin^2 \left(\frac{jp\pi}{n+1}\right) = \frac{1}{2}\left[\sum_{j=1}^{n} \left(1-\cos \frac{2\, jp\pi}{n+1}\right)\right]$$

$$\sum_{j=1}^{n} \sin^2 \left(\frac{jp\pi}{n+1}\right) = \frac{n}{2} - \frac{S'_n}{2}$$

avec

$$s'_n = \sum_{k=1}^{n} \cos \left(\frac{2\, k\, p\pi}{n+1}\right)$$

$$s'_n = \sum_{k=1}^{n} \cos \gamma k$$

avec

$$\gamma = \frac{2p\pi}{n+1}$$

$$\cos \gamma k = R\,(e^{i\gamma k}) \quad \text{avec } i^2 = -1$$

$$s'_n = R \left(\sum_{k=1}^{n} e^{i\gamma k}\right) = R\,(S_n)$$

avec

$$S_n = \sum_{k=1}^{n} e^{i\gamma k}$$

S_n est donc la somme des n premiers termes d'une progression géométrique de premier terme $e^{i\gamma}$ et de raison $e^{i\gamma}$

$$S_n = e^{i\gamma} \cdot \frac{1 - e^{i.\gamma.n}}{1 - e^{i\gamma}}$$

$$\text{Conjugué } (1-e^{i\gamma}) = (1 - e^{-i\gamma})$$

$$S_n = \frac{e^{i\gamma}.(1-e^{i\gamma n})(1-e^{-i\gamma})}{(1-e^{+i\gamma})\ (1-e^{-i\gamma})}$$

$$(1-e^{i\gamma}).(1-e^{-i\gamma}) = 2.(1-\cos\gamma)$$

$$e^{i\gamma}.(1-e^{i\gamma n})(1-e^{-i\gamma}) = e^{i\gamma} - 1 - e^{i\gamma.(n+1)} + e^{i\gamma n}$$

$$S_n = \frac{e^{i\gamma} - 1 - e^{i\gamma.(n+1)} + e^{i\gamma n}}{2.(1-\cos\gamma)}$$

$$S'_n = R(S) = \frac{\cos\gamma - 1 - \cos(\gamma.(n+1) + \cos\gamma n}{2\ (1 - \cos\gamma)}$$

$$S'_n = -\frac{1}{2} + Q$$

avec

$$Q = \frac{\cos\gamma n - \cos(\gamma\ (n+1))}{2\ (1 - \cos\gamma)}$$

comme

$$\gamma = \frac{2p\pi}{n+1}$$

on a

$$Q = \frac{\cos\frac{2np\pi}{n+1} - \cos\frac{2(n+1)p\pi}{n+1}}{2\ (1-\cos\frac{2p\pi}{n+1})}$$

$$\cos\frac{2(n+1)p\pi}{n+1} = \cos 2p\pi = 1 \text{ puisque p est entier}$$

$$\cos\frac{2np\pi}{n+1} = \cos\left(\frac{2(n+1)p\pi}{n+1} - \frac{2p\pi}{n+1}\right) = \cos\left(2p\pi - \frac{2p\pi}{n+1}\right) = \cos\left(\frac{2p\pi}{n+1}\right)$$

$$Q = \frac{-1 + \cos\left(\frac{2p\pi}{n+1}\right)}{2\left(1-\cos\left(\frac{2p}{n+1}\right)\right)} = -\frac{1}{2}$$

$$S'_n = -\frac{1}{2} + Q = -1$$

$$\sum_{k=1}^{n} \sin^2\left(\frac{kp\pi}{n+1}\right) = \frac{n}{2} - \frac{1}{2} s'_n = \frac{n+1}{2}$$

Donc :

$$\sum_{j=1}^{n} \sin^2\left(\frac{jp\pi}{n+1}\right) = \frac{n+1}{2} \quad \text{pour } p = 1,2,\ldots,n \quad (*)$$

On remarque que, contrairement à ce à quoi on pouvait s'attendre, le résultat de la sommation est indépendant de p.

D'autre part, on a vu (**) que, dans le cas de l'élimination avec divisions, l'on a :

$$-1 > \frac{K}{2} \geqslant d_j \geqslant K \quad \text{pour } j = 1,2,\ldots, NX-2$$

donc

$$\frac{K}{2} - \frac{NX-1}{2} \geqslant \sum_{j=1}^{NX-2} d_j \cdot \sin^2\left(\frac{jp\pi}{NX-1}\right) \geqslant K \cdot \frac{NX-1}{2}$$

donc, en reportant dans (5), on obtient

$$|\varepsilon K| \leq |\delta_M A_p| \leq |2\,\varepsilon K|$$

D'autre part on remarque que la matrice unité I admet en particulier comme vecteur propre normé :

(*) On aurait pu obtenir le même résultat en faisant $x = \frac{p\pi}{n+1}$ dans la formule

$$\sum_{k=1}^{n} \sin^2(kx) = \frac{n}{2} - \frac{\cos((n+1)x).\sin(nx)}{x.\sin x}$$

formule 1.351 de la page 31.

(**) p. 65

$$\frac{1}{\sqrt{n}} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

On voit donc, d'après le théorème ci-dessus, que l'on a :

$$\delta\mu_2 = \frac{1}{\sqrt{n}} \cdot (1,1,\ldots,1) \cdot \varepsilon \cdot \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \cdot \frac{1}{\sqrt{n}} \quad \text{avec } n = NX-2$$

$$\delta\mu_I = \frac{\varepsilon}{n} \left(n-1 - \sum_{j=1}^{n-1} \frac{1}{d_j} \right)$$

En raisons des approximations faites, on peut changer cela en

$$\delta\mu_I = \frac{\varepsilon}{n} \left(n - \sum_{j=1}^{n} \frac{1}{d_j} \right) \tag{8}$$

Comme on a vu (*) que

$$-1 > \frac{K}{2} \geqslant d_j \geqslant K$$

on a :

$$\frac{n}{K} \geqslant \sum_{j=1}^{n} \frac{1}{d_j} \geqslant \frac{2n}{K}$$

en reportant dans (8), on obtient :

$$\left|\varepsilon \cdot \left(1 - \frac{1}{K}\right)\right| \leqslant \left|\delta\mu_I\right| \leqslant \left|\varepsilon\left(1 - \frac{2}{K}\right)\right|$$

mais

$$K = -\left(2 + \frac{1}{n}\right) < -2$$

donc

$$\left|\delta\mu_I\right| \leqslant 2 \cdot |\varepsilon| \tag{9}$$

(*) pages 64 et 65

D'autre part, on a vu (*) que

$$\mu_{A_p} = \frac{1}{\sigma.\lambda}.\left[-1 - 2\sigma.\lambda.(1 - \cos\frac{p.\pi}{n+1})\right]$$

$$\mu_{B_p} = \frac{1}{\sigma.\lambda}.\left[-1 + 2.\lambda.(1-\sigma).(1 - \cos\frac{p.\pi}{n+1})\right]$$

Donc, d'après (4) on a :

$$\mu'_p = \frac{\frac{1}{\sigma.\lambda}\left[1 - 2.\lambda.(1-\sigma)\ (1-\cos\frac{p.\pi}{NX-1})\right].(1+2\varepsilon)(1+\delta\mu_I)}{\frac{1}{\sigma.\lambda}.\left[1+2.\sigma.\lambda(1-\cos\frac{p.\pi}{NX-1})\right] + \delta\mu_{A_p}}$$

$$\mu'_p = \frac{\left[1-2.\lambda.(1-\sigma).(1-\cos\frac{p.\pi}{NX-1})\right]\ (1+2\varepsilon)\ (1+\delta\mu_I)}{\left[1+2.\sigma.\lambda.(1-\cos\frac{p\pi}{NX-1})\right] + \sigma.\lambda.\delta\mu_{A_p}}$$

alors que dans le cas théorique les valeurs propres sont . (*)

$$\mu_p = \frac{1-2.\lambda.(1-\sigma).(1-\cos\frac{p\pi}{NX-1})}{1+2\sigma.\lambda.(1-\cos\frac{p.\pi}{NX-1})}$$

D'après (9) on a :

$$|(1 + 2\varepsilon)\ (1 + \delta\mu_I)| \leq 4.|\varepsilon|$$

D'après (7) on a

$$|\sigma.\lambda.\delta\mu_{A_p}| \leq 2.\sigma.\lambda.K\ |\varepsilon|$$

Pour établir les conditions théoriques de stabilité on est parti de la condition $|\mu_p| \leq 1$ et on a abouti aux conditions exposés plus haut (**), mais pour passer de l'un

---

(*) page ~~53~~ 51

(**) page 53

à l'autre, on a majoré $\left|(1-\cos\frac{p\pi}{NX-1}) \text{ pour } p = 1,2,\ldots NX-2\right|$ par 2 ; devant le peu de "finesse" de cette majoration, on peut estimer que les "différences" entre μ_p et μ'_p sont négligeables en raison de (12) et (13).

Dans le cas où l'on calcule en point décimal fixe en utilisant l'élimination avec division pour la résolution des systèmes linéaires, on a vu (*) que la matrice $(A^{-1}B)$ est remplacée par la matrice :

$$M'' = \left[(A + \varepsilon D)^{-1} B\right]$$

si, on appelle $\delta\mu'_{Ap}$ la variation de μ_{A_p} causée par la variation matricielle εD, on a :

$$\mu''_p = \frac{\mu_{B_p}}{\mu_{A_p} + \delta\mu'_{A_p}}$$

$$\delta\mu'_{A_p} = \frac{1}{\sqrt{\sum_{j=1}^{m} \sin\left(\frac{jp\pi}{m+1}\right)}} \left(\sin\frac{p\pi}{n+1}, \ldots, \sin\frac{mp\pi}{m+1}\right) \cdot \varepsilon \cdot \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{pmatrix} \cdot \begin{pmatrix} \sin\frac{p\pi}{n+1} \\ \vdots \\ \sin\frac{mp\pi}{m+1} \end{pmatrix} \cdot \frac{1}{\sqrt{\sum_{j=1}^{n} \sin^2\left(\frac{jp\pi}{n+1}\right)}}$$

$$\delta\mu'_{A_p} = \varepsilon \cdot \frac{\sum_{j=1}^{n} d_j \sin^2\left(\frac{jp\pi}{n+1}\right)}{\sum_{j=1}^{n} \sin^2\left(\frac{jp\pi}{n+1}\right)} \quad \text{avec} \quad n = NX-2$$

or

$$-1 > \frac{K}{2} \geqslant d_j \geqslant K$$

et

$$\sum_{j=1}^{n} \sin^2\left(\frac{jp\pi}{n+1}\right) = \frac{n+1}{2}$$

donc

$$\left|\frac{\varepsilon K}{2}\right| \leqslant \left|\delta\mu'_{A_p}\right| \leqslant \left|\varepsilon K\right|$$

(*) page 71

D'autre part :

$$\mu'_p = \frac{1-2.\lambda.(1-\nu).(1-\cos\frac{p\pi}{NX-1})}{\left[1+2.\lambda.\nu.(1-\cos\frac{p}{NX-1})\right]+\nu.\lambda.\delta\mu_{A_p}}$$

On peut faire les mêmes remarques sur les ordres de grandeur que dans le cas du calcul en point décimal flottant par l'élimination avec divisions.

On voit donc que les perturbations apportées aux conditions de stabilité par les erreurs de calcul sont négligeables quand on utilise l'élimination avec divisions, que ce soit en point décimal flottant ou en point décimal fixe.

Pour les cas où l'on utilise l'élimination sans division pour la résolution des systèmes linéaires dans la méthode de Crank et Nicolson, on a vu que les matrices qui remplacent la matrice $(A^{-1}B)$ différent de celle-ci par des matrices d'erreurs comportant des éléments non bornées. On peut donc penser que dans ces deux cas la stabilité est perturbée par les erreurs de calcul, mais pratiquement, il est difficile de distinguer ce qui, dans les erreurs globales, provient directement des erreurs de calcul ou provient d'une éventuelle instabilité.

Conclusion

Dans la résolution de l'équation de la chaleur par la méthode de Crank et Nicolson,

1) si l'on utilise l'élimination avec divisions pour la résolution des systèmes linéaires, que l'on calcule en point décimal flottant ou en point décimal fixe, les erreurs de calcul ne modifient pas de façon appréciable les conditions théoriques de stabilité, c'est-à-dire :

ou bien $\sigma \geq \frac{1}{2}$ et λ quelconque

ou bien $\sigma < \frac{1}{2}$ et $\lambda \leq \frac{1}{2(1-\sigma)}$

2) si l'on utilise l'élimination sans division pour la résolution des systèmes linéaires, que l'on calcule en point décimal flottant ou en point décimal fixe, la stabilité est perturbée, sans qu'il soit facile de distinguer dans l'erreur globale ce qui provient de l'instabilité et ce qui provient directement des erreurs de calcul.

X - Etude expérimentale des erreurs de calcul dans la méthode de Crank et Nicolson :

On doit vérifier expérimentalement les conclusions auxquelles on a abouti dans les paragraphes précédents.

Pour cela on a écrit des programmes qui exécutent la méthode de Crank et Nicolson simultanément avec la précision maximum de la calculatrice et avec des erreurs d'arrondi grossies fixes ou flottantes et qui comparent les résultats : ces programmes et leur utilisation sont analogues à ceux écrits pour la méthode explicite dont on a parlé plus haut (*).

Pour le calcul en point décimal flottant par la méthode de Crank et Nicolson avec utilisation de l'élimination avec divisions pour la résolution des systèmes linéaires, on doit vérifier les 3 conclusions suivantes (**)

A - L'erreur de calcul moyenne sur les résultats d'un certain niveau en t est une erreur relative proportionnelle au nombre de pas en t calculés depuis le début et indépendante des valeurs de la fonction solution.

B - Cette erreur relative sur les résultats est proportionnelle à l'erreur relative élémentaire.

C - Les conditions de stabilité ne sont pas modifiées de façon appréciable par les erreurs de calcul, et, en particulier, quand $\sigma \geq \frac{1}{2}$ il y a stabilité quel que soit λ.

(*) page 20

(**) corollaire pratique page 65 et conclusion page 109

La conclusion A semble vérifiée par l'examen des courbes tracées au bas des graphiques 41 à 61 : (*) ces courbes donnent l'erreur relative moyenne de calcul d'un certain niveau en t en fonction du nombre de pas en t calculés depuis le début. Les points portés dans le haut de ces graphiques donnent, toujours en fonction du nombre de pas en t calculés depuis le début, la pente locale de la courbe dont on vient de parler : leur plus ou moins grande dispersion autour de la droite horizontale qui indique la valeur moyenne de cette pente permet d'apprécier le caractère plus ou moins approximatif de la courbe du dessous.

On trouvera page 112 un tableau des paramètres qui résument les expériences numériques qui sont à l'origine de ces courbes. On voit qu'on a fait varier 3 paramètres

- d'une part, pour une erreur relative élémentaire en 10^{-6} et pour la fonction solution $[e^{-t} \sin x]$, on a pris les valeurs suivantes de $k = \Delta t/(\Delta x)^2$: 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 1 2 3 4 5 10 et 20 ;

- d'autre part, pour $k = 0,5$ et toujours pour la fonction solution $[e^{-t} . \sin x]$, on a pris des erreurs relatives élémentaires en 10^{-7}, 10^{-5}, 10^{-4} et 10^{-3} ;

- enfin, pour $k = 0,5$ et pour une erreur relative élémentaire en 10^{-6}, on a pris la fonction solution $[10^3 . e^{-t} . \sin x]$, afin d'avoir des valeurs de la fonction solution 10^3 fois plus grandes que précédemment.

(*) On a tracé les graphiques 41 à 61 des expériences numériques décrites dans le tableau de la page mais pour des raisons matérielles, on n'a reproduit ici que les graphiques ~~45~~ : 59 et 61.

TABLEAU DES PARAMETRES DES COURBES 41 A 61 -

Numéro	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
Date	26/2/64	26/2/64	27/2/64	27/2/64	27/2/64	27/2/64	28/2/64	28/2/64	3/3/64	28/2/64	2/3/64	2/3/64	3/3/64	3/3/64	12/3/64	27/5/64	13/3/64	14/3/64	20/3/64	27/5/64	26/5/64
Fonction solution	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$10^{3}.e^{-t}.\sin x$
Val. inf. de x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Val. inf. de t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
Pas en t	10^{-5}	$0,2.10^{-4}$	$0,3.10^{-4}$	$0,4.10^{-4}$	$0,5.10^{-4}$	$0,6.10^{-4}$	$0,7.10^{-4}$	$0,8.10^{-4}$	$0,9.10^{-4}$	10^{-4}	$0,2.10^{-3}$	$0,3.10^{-3}$	$0,4.10^{-3}$	$0,5.10^{-3}$	10^{-3}	$0,2.10^{-2}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$
N^{bre} points // Ox	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101
N^{bre} points // Ot	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201	201
N^{bre} bits bin. tronqués	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	3	10	13	17	7
N^{bre} chiffres décimaux significatifs exacts à chaque opérat.	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	4	3	2	5
Ordre de grandeur erreur relative élémentaire	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-5}	10^{-4}	10^{-3}	10^{-6}
σ	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
$\lambda = \frac{\Delta t}{(\Delta x)^2}$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	2	3	4	5	10	20	0,5	0,5	0,5	0,5	0,5
Pente Moyenne	$0,30.10^{-5}$	$0,29.10^{-5}$	$0,28.10^{-5}$	$0,20.10^{-5}$	$0,19.10^{-5}$	$0,32.10^{-5}$	$0,18.10^{-5}$	$0,27.10^{-5}$	$0,16.10^{-5}$	$0,14.10^{-5}$	$0,26.10^{-5}$	$0,10.10^{-5}$	$0,43.10^{-7}$	$0,60.10^{-6}$	$0,23.10^{-6}$	$0,32.10^{-5}$	$0,11.10^{-6}$	$0,98.10^{-5}$	$0,13.10^{-3}$	$0,40.10^{-2}$	$0,19.10^{-5}$

On constate que ces courbes donnant l'erreur relative en fonction du nombre de pas en t calculés sont bien approximativement des droites : on a donc une vérification expérimentale de la conclusion A.

Il faut toutefois mentionner comme exceptions les courbes 53, 55 et 56 dont la configuration est assez peu linéaire : elles correspondent respectivement à $\lambda = 4$ $\lambda = 10$ et $\lambda = 20$: on ne peut pas expliquer facilement ces anomalies à partir des théorèmes établis plus haut : mais, même dans ces cas-là si l'erreur est peu conforme à ce qui a été prévu au point de vue évolution, elle reste du moins dans les limites de ce qui a été prévu au point de vue ordre de grandeur de l'erreur relative.

On constate que les courbes 61 et 45 sont à peu près identiques : elles correspondent à des paramètres identiques sauf en ce qui concerne les valeurs de la fonction solution qui sont 10^3 plus grandes dans le cas 61 : cela semble une vérification expérimentale de ce que nous avons affirmé : l'erreur sur les résultats est une véritable erreur relative, indépendante des valeurs de la fonction solution.

La conclusion B semble vérifiée par l'examen des pentes moyennes des courbes qu'on vient de décrire (on trouve les valeurs de ces pentes moyennes à la dernière ligne du tableau de la page 112). On voit que l'ordre de grandeur de cette pente moyenne ne dépend que de l'ordre de grandeur de l'erreur relative élémentaire.

On note comme exceptions les courbes 53, 54 et 55 qui correspondent respectivement à $\lambda = 4$, $\lambda = 5$ et $\lambda = 10$: mais là aussi les valeurs trouvées expérimentalement , si elles ne sont pas conformes à ce que laissaient prévoir les

théorèmes établis plus haut, sont du moins inférieures à ces prévisions théoriques.

Si l'on compare avec la méthode explicite, également en point décimal flottant, on remarque qu'avec la méthode de Crank et Nicolson les pentes moyennes des courbes sont approximativement de 5 à 10 fois plus grandes que celles trouvées dans les mêmes conditions avec la méthode explicite : comme la valeur de ces pentes exprime pour chaque cas l'augmentation moyenne d'erreur relative pour chaque pas supplémentaire en t, il semble que l'on puisse en conclure que, lorsque l'on calcule en point décimal flottant, la méthode de Crank et Nicolson ne surpasse en précision la méthode explicite que si l'on prend des pas en t 5 à 10 fois plus grands ; cela correspond à des valeurs de λ comprises entre 2 et 5 ; il semblerait d'après des expériences ci-dessus que les choix $\lambda = 2$ et $\lambda = 3$ soient assez intéressants car on a vu que, pour des valeurs supérieures de λ, l'erreur relative sur les résultats reste dans des limites acceptables mais évolue de façon assez irrégulière en fonction du nombre de pas en t calculés.

Pour vérifier la conclusion C, il aurait fallu en toute rigueur tracer des graphiques donnant l'erreur globale : mais on s'est aperçu qu'en pratique, dans tous les cas, l'évolution de l'erreur globale suit de très près l'évolution de l'erreur de calcul : aussi peut-on se contenter d'examiner les courbes citées plus haut et de constater qu'en aucun cas on n'a un "départ" de l'erreur comme on l'avait constaté pour la méthode explicite dans les cas où la condition de stabilité n'était pas satisfaite.

Pour le calcul en point décimal fixe par la méthode de Crank et Nicolson, avec l'utilisation de l'élimination avec divisions pour la résolution des systèmes linéaires, on doit vérifier les conclusions suivantes :

D - L'erreur de calcul moyenne sur les résultats d'un certain niveau en t est une erreur absolue proportionnelle au nombre de pas en t calculés depuis le début ;

E - Cette erreur absolue sur les résultats est aussi proportionnelle à l'erreur absolue élémentaire ;

F - Cette erreur absolue sur les résultats est une fonction croissante mais non linéaire des valeurs de la fonction solution ;

G - Les conditions de stabilité ne sont pas modifiées par les erreurs de calcul, et, en particulier, quand $\sigma \geq \frac{1}{2}$, il y a stabilité quelque soit λ.

Les expériences numériques faites pour vérifier ces conclusions sont résumées par le tableau de la page 117 et par les courbes 71 à 91 (*). Elles sont tout à fait analogues aux expériences faites pour la même méthode dans le cas du calcul en point décimal flottant : simplement on a remplacé les erreurs relatives, tant élémentaires que sur les résultats, par des erreurs absolues.

La conclusion D semble vérifiée par l'examen de l'aspect général des courbes 71 à 91 ; ce ne sont pas exactement des droites et leur pente tend à décroître, et ce d'autant plus que le pas en t est grand. Cela n'est pas contraire à la conclusion D mais exprime l'influence de la conclusion F : l'erreur absolue sur les résultats est proportionnelle au nombre de pas en t calculés depuis le début

(*) Pour des raisons matérielles, on n'a reproduit ici que les graphiques 75, 89 et 91.

(conclusion D) mais est aussi une fonction croissante non linéaire des valeurs de la fonction solution (conclusion F) or, quand t augmente, ces valeurs de la fonction solution diminuent, et, si l'on prend comme échelle non pas t mais le nombre de pas en t, cette croissance est évidemment d'autant plus rapide que le pas en t est plus grand : c'est cela qui explique que les pentes des courbes 71 à 91 tendent à diminuer quand augmente l'abscisse.

La conclusion E semble vérifiée par l'examen des pentes moyennes des courbes dont on vient de parler.

TABLEAU DES PARAMETRES DES COURBES 71 à 91 -

Numéro	71	72	73	74	75	76	77	78	79	80	81
Date	27/5/64	27/5/64	27/5/64	27/5/64	27/5/64	28/5/64	28/5/64	28/5/64	29/5/64	29/5/64	29/5/64
Fonction Solution	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$
Val. inf. de x	0	0	0	0	0	0	0	0	0	0	0
Val. inf. de t	0	0	0	0	0	0	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,1.10^{-4}$	$0,2.10^{-4}$	$0,3.10^{-4}$	$0,4.10^{-4}$	$0,5.10^{-4}$	$0,6.10^{-4}$	$0,7.10^{-4}$	$0,8.10^{-4}$	$0,9.10^{-4}$	10^{-4}	$0,2.10^{-3}$
Nbre points // Ox	101	101	101	101	101	101	101	101	101	101	101
Nbre points // Ot	201	201	201	201	201	201	201	201	201	201	201
Nbre bits bin. après virgule	20	20	20	20	20	20	20	20	20	20	20
Nbre chiffres décim. exacts après virgule pour chaque opération	5	5	5	5	5	5	5	5	5	5	5
Ordre de grandeur erreur absolue [illegible]	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
σ	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
$\lambda = \frac{\Delta t}{(\Delta x)^2}$	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	2
Pente diagramme	$0,48.10^{-6}$	$0,52.10^{-6}$	$0,61.10^{-6}$	$0,56.10^{-6}$	$0,56.10^{-6}$	$0,77.10^{-6}$	$0,77.10^{-6}$	$0,80.10^{-6}$	$0,77.10^{-6}$	$0,72.10^{-6}$	$0,96.10^{-6}$

Numéro	82	83	84	85	86	87	88	89	90	91
Date	29/5/64	29/5/64	29/5/64	29/5/64	29/5/64	27/3/64	27/3/64	25/3/64	27/3/64	28/4/64
Fonction Solution	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$e^{-t}.\sin x$	$10.e^{-3t}.\sin x$
Val. inf. de x	0	0	0	0	0	0	0	0	0	0
Val. inf. de t	0	0	0	0	0	0	0	0	0	0
Pas en x	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
Pas en t	$0,3.10^{-3}$	$0,4.10^{-3}$	$0,5.10^{-3}$	10^{-3}	$0,2.10^{-2}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$	$0,5.10^{-4}$
Nbre points // Ox	101	101	101	101	101	101	101	101	101	101
Nbre points // Ot	201	201	201	201	201	201	201	201	201	201
Nbre bits bin. après virgule	20	20	20	20	20	23	17	14	10	20
Nbre chiffres décim. exacts après virgule pour chaque opération	5	5	5	5	5	6	4	3	2	5
Ordre de grandeur erreur absolue [illegible]	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-7}	10^{-5}	10^{-4}	10^{-3}	10^{-6}
σ	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
$\lambda = \frac{\Delta t}{(\Delta x)^2}$	3	4	5	10	20	0,5	0,5	0,5	0,5	0,5
Pente diagramme	$0,27.10^{-6}$	$0,14.10^{-5}$	$0,51.10^{-5}$	$0,51.10^{-6}$	$0,30.10^{-6}$	$0,93.10^{-7}$	$0,50.10^{-5}$	$0,35.10^{-4}$	$0,75.10^{-3}$	$0,23.10^{-4}$

(ces pentes moyennes se trouvent sur la dernière ligne du tableau de la page 117) ; l'ordre de grandeur de ces pentes moyennes est bien proportionnel à l'ordre de grandeur de l'erreur absolue élémentaire ; il y a comme exception la pente de la courbe 91 mais cela ne contredit pas la conclusion E et s'explique par l'influence de la conclusion F.

Si l'on compare avec le calcul en point décimal fixe par la méthode explicite, on remarque que l'on obtient avec le calcul en point décimal fixe par la méthode de Crank et Nicolson avec élimination avec divisions des pentes moyennes de ces courbes qui sont sensiblement du même ordre que celles trouvées dans les mêmes conditions avec la méthode explicite : on a vu (*) qu'il n'en est pas de même pour le calcul en point décimal flottant.

La conclusion F est vérifiée par l'examen de l'allure générale des courbes 71 à 91 comme on l'a expliqué à propos de la conclusion D. La conclusion F se vérifie aussi par la comparaison des courbes 75 et 91 pour lesquelles tous les paramètres sont identiques sauf les valeurs de la fonction solution qui sont 10^3 fois plus grandes dans le cas 91 que dans le cas 75;on constate que, toutes choses égales d'ailleurs,les erreurs absolues sur les résultats sont environ 50 fois plus grandes dans le cas 91 que dans le cas 75.

Pour vérifier la conclusion G, il aurait fallu ici aussi tracer des courbes d'erreurs globales mais comme dans le cas du calcul en point décimal flottant, on s'est aperçu que l'évolution de cette erreur globale suit toujours de très près l'évolution de l'erreur de calcul ; aussi s'est-on contenté des courbes donnant cette erreur de calcul; on

(*) page 114

constate qu'en aucun des cas étudiés, il n' y a "départ" de l'erreur et qu'il y a par conséquent stabilité dans tous ces cas.

En ce qui concerne l'utilisation de l'élimination sans division dans la méthode de Crank et Nicolson, la vérification expérimentale est plus difficile car on tombe très vite dans des cas de dépassement de capacité de la calculatrice : nombres au module trop grand. Aussi n'a-t-on pu se placer dans les mêmes conditions que lors de l'utilisation de l'élimination avec divisions : toutes choses égales d'ailleurs, on n'a pas pu prendre 101 points selon l'axe des x, mais seulement 11, puis 21, puis 31, puis 41 et enfin 51.

La signification des expériences numériques n'est donc pas très claire dans ce cas ; on peut néanmoins en dégager les conclusions suivantes :

- l'erreur sur les résultats est une fonction croissante du nombre de pas en t calculées depuis le début, la croissance de cette fonction étant beaucoup moins régulière que dans le cas de l'utilisation de l'élimination avec divisions.
- la pente moyenne de cette fonction augmente avec le nombre de points selon l'axe des x.
- cette pente semble en général beaucoup plus forte que lorsque l'on utilise l'élimination avec divisions.

C H A P I T R E IV

N O T E

SUR LES PROCEDURES ET SOUS PROGRAMMES TRONC ET TROFXS

QUI AUGMENTENT LES ERREURS D'ARRONDI -

I - GENERALITES ET UTILISATION

1) Caractéristiques générales

Les procédures et sous programmes TRONC et TROFXS ont pour but de mesurer l'influence des erreurs d'arrondi élémentaires sur les résultats finaux d'un algorithme.

Elles permettent d'augmenter à volonté les erreurs d'arrondi élémentaires et ce par simple insertion dans un programme déjà écrit de 2 exécutions de procédures ou de 2 appels de sous programmes.

La procédure ou le sous programme TRONC fait que, dans une partie déterminée du programme où il est inséré, chaque opération arithmétique flottante produit un résultat élémentaire comportant un nombre de chiffres significatifs inférieur à la normale et déterminée par le paramètre effectif ou l'argument d'appel.

La procédure ou le sous programme TROFXS fait que, dans une partie déterminée du programme où elle est insérée, chaque opération arithmétique flottante est transformée en une simulation d'opération fixe dont le résultat élémentaire comporte après la virgule un nombre de chiffres significatifs déterminée à volonté par le paramètre effectif ou l'argument d'appel.

Les procédures TRONC et TROFXS peuvent s'insérer dans tout programme écrit en ALGOL : elles supposent que ce programme ALGOL est ensuite exploité par le compilateur ALGOL de Grenoble pour IBM 7044. (*) La procédure TRONC ou TROFXS doit

---

(*) [1]

être déclarée en tête du programme ; cette déclaration doit être physiquement la première déclaration de procédure du programme (ce qui interdit d'utiliser à la fois TRONC et TROFXS dans un même programme).

Les sous programmes TRONC et TROFXS peuvent s'insérer dans tout programme écrit en FORTRAN IV ou en MAP et destiné à être exécuté sur IBM 7044. Si le programme donné est en MAP, les appels de TRONC ou de TROFXS doivent être conformes aux séquences standard du compilateur FORTRAN IV. Il existe aussi une version du sous programme TRONC pouvant s'insérer dans un programme écrit en FORTRAN II ou en FAP et destiné à être exécuté sur IBM 7090 ou 7094. Si le programme donné est en FAP, les appels de TRONC doivent être conformes aux séquences standard du compilateur FORTRAN II. Il faut insérer dans le paquet du programme les cartes (éventuellement binaires) de TRONC ou TROFXS ; l'emplacement de ce groupe de cartes dans le paquet global n'a pas d'importance.

Dans tous les cas, seules sont transformées les opérations arithmétiques flottantes normalisées en simple précision.

2) Détermination de l'erreur d'arrondi élémentaire

Dans les machines IBM 7044, 7090 et 7094, le résultat élémentaire de chaque opération arithmétique flottante élémentaire comporte normalement 27 chiffres binaires significatifs; TRONC et TROFXS font qu'il en comporte moins.

Dans le cas de TRONC, chaque résultat élémentaire ne comportera plus que (27-I) chiffres binaires significatifs, I étant le paramètre effectif ou l'argument d'appel de TRONC. Les chiffres binaires significatifs supprimés seront remplacés par des 0.

Dans le cas de TROFXS, chaque résultat élémentaire comportera au plus J chiffres binaires après la virgule binaire virtuelle, J étant le paramètre formel ou l'argument d'appel de TROFXS. Les chiffres binaires éventuellement supprimés seront remplacés par des zéros. Il faut remarquer que dans ce cas il s'agit d'une simulation de point décimal fixe en faite à l'aide de point décimal flottant : les opérations arithmétiques affectées sont celles primitivement écrites en point décimal flottant et la représentation en machine des résultats élémentaires reste celle du point décimal flottant : c'est pourquoi on parle de virgule binaire "virtuelle" et on dit que chaque résultat élémentaire comporte "au plus" J chiffres binaires après cette virgule : en effet, il en comportera moins de J si, étant donné l'ordre de grandeur du nombre, J chiffres binaires après la virgule correspond à plus de 27 chiffres binaires significatifs en représentation flottante.

Le paramètre effectif ou l'argument d'appel de TRONC ou de TROFXS peut être une constante entière, une variable entière, ou n'importe quelle expression arithmétique entière, pouvant éventuellement varier au cours de l'exécution du programme. Dans le cas de TRONC, ce paramètre effectif ou cet argument d'appel doit de plus être compris entre 0 et 27, bornes comprises. Cette dernière restriction ne s'applique pas à TROFXS, l'interprétation en étant de toutes façons limitée par ce qu'on a dit sur le nombre total de chiffres binaires significatifs.

Entre le nombre de chiffres binaires significatifs et le nombre de chiffres décimaux significatifs, on a établi expérimentalement le tableau suivant qui n'a qu'une valeur

approximative.

Nombre de chiffres binaires significatifs tronqués (ou paramètre effectif de TRONC)	3 7 10 13 17 20
Nombre minimum de chiffres décimaux significatifs exacts de chaque résultat élémentaire.	6 5 4 3 2 1

3) Détermination de la séquence où l'on veut augmenter les erreurs d'arrondi

Ce n'est pas dans tout le programme où l'on a inséré TRONC ou TROFXS, mais seulement dans une ou plusieurs séquences de ce programme, que les opérations arithmétiques flottantes sont "tronquées" de la manière décrite ci-dessus.

La séquence de programme où l'on veut ainsi "grossir" les erreurs d'arrondi doit être "encadrée" par des ordres d'exécution de la procédure de troncature choisie ou par des appels du sous programme de troncature.

Il est essentiel que la séquence en question soit non seulement précédée mais aussi suivie par un ordre d'exécution de la procédure de troncature ou par un appel du sous programme de troncature : ce 2e ordre d'exécution ou ce 2e appel indique à la procédure ou au sous programme la fin de la séquence dans laquelle les opérations arithmétiques flottantes doivent être transformées. L'absence de ce 2e ordre d'exécution ou de ce 2e appel provoquerait un bouclage ou un diagnostic d'exécution.

On voit que ces ordres d'exécution ou ces appels ne jouent pas le même rôle selon leur occurence logique : ils sont pour ainsi dire groupés par "paires" et l'on appellera premier et second ordre d'une paire le premier et le second ordre d'exécution d'une procédure de troncature, ou le premier et le second appel d'un sous programme de troncature.

Il est très important de veiller à ce que dans tous les cas possibles d'exécutions du programme donné (déroulement simple, boucles, déroutements absolus ou conditionnels), les 2 ordres d'une paire ou bien ne soient exécutés ni l'un ni l'autre, ou bien soient exécutés tous les deux et toujours dans le même ordre.

On peut vouloir grossir les erreurs d'arrondi dans plusieurs séquences distinctes d'un même programme : chaque séquence doit alors être "encadrée" comme on vient de le dire et il faut veiller à ce que, dans tous les cas possibles, les "paires" soient exécutées comme telles. De plus le "second ordre" d'une certaine "paire" ne doit pas servir de "second ordre" à une autre "paire".

Cette obligation d'exécuter les "paires" correctement ne limite pas l'emploi du "grossissement" des erreurs d'arrondi et ne rend pas nécessaire la modification des programmes que l'on veut étudier : il suffit en effet pour respecter cette obligation d'insérer judicieusement les ordres d'exécution des procédures en question ou les appels des sous programmes en question.

L'erreur d'arrondi "truquée" d'une séquence est déterminé par le paramètre effectif ou l'argument d'appel du "premier ordre" de la "paire" encadrant la séquence ; le paramètre effectif ou

l'argument du "second ordre" de la "paire" n'a pas d'influence mais il doit exister sous peine de faute de grammaire et il est pratique de le prendre égal à celui du "premier ordre" de la "paire".

Les séquences où l'on veut grossir les erreurs d'arrondi peuvent comporter n'importe quel type d'instruction, sauf celles utilisant les mémoires d'adresses absolues 0 et 2. Pratiquement cette restriction sur l'utilisation des mémoires 0 et 2, ne semble pas avoir d'effet réellement limitaif sur l'étude des programmes écrit en ALGOL ou en FORTRAN : le seul cas où l'on ait rencontré des difficultés est celui des diagnostics de dépassement de capacités en point décimal flottant qui sont mal exécutés lorsqu'ils sont provoqués par une opération située à l'intérieur d'une séquence où l'on grossit les erreurs d'arrondi.

II - DETAILS DE PROGRAMMATION

1) Généralités

Les procédures TRONC et TROFXS à insérer dans des programmes ALGOL sont des procédures ALGOL dont le corps est en code c'est-à-dire en MAP.

Les sous programmes TRONC et TROFXS à insérer dans des programmes en FORTRAN IV ou MAP sont des sous programmes écrits en MAP. La version du sous programme TRONC à insérer dans des programmes en FORTRAN II ou en FAP est un sous programme écrit en FAP anglais.

C'est au moment de l'éxécution que ces procédures ou sous programmes modifient le programme dans lequel ils sont insérés.

Ces procédures ou sous programmes comportent chacune 4 parties commençant la première à l'adresse symbolique TRONC ou TROFXS ou E1, les 3 autres respectivement aux adresses symboliques E2, E3 et TRAP. Dans les procédures ALGOL, E1, E2 et E3 désignent elles-mêmes des procédures déclarées à l'intérieur de la procédure TRONC ou de la procédure TROFXS.

2) Partie commençant en TRONC, TROFXS ou E1

L'exécution de cette partie a pour effet :

a) de remplacer dans le programme donné l'ordre d'exécution de TRONC ou TROFXS ou le CALL TRONC ou CALL TROFXS par un ordre de déroutement direct en E2 ;

b) d'examiner dans le programme donné toutes les instructions qui

suivent l'ordre de déroutement qui vient d'être exécuté, et ceci jusqu'à rencontrer un nouvel ordre de déroutement semblable ; de remplacer parmi ces instructions toutes celles qui sont de l'un des 4 types d'opérations arithmétiques flottantes normalisées en simple précision par des ordres de déroutement en TRAP avec repérage du type d'opérations arithmétiques ; ces nouveaux ordres de déroutement sont des instructions MAP STR qui utiliseront les mémoires 0 et 2 ;

c) de remplacer dans le programme donné le 2e ordre d'exécution de TRONC ou TROFXS ou le 2e CALL TRONC ou CALL TROFXS par un ordre de déroutement direct en E3.

d) de se dérouter en E2.

3) Partie commençant en E2

L'exécution de cette partie a pour effet :

a) dans le cas de TRONC, de déterminer les instructions de troncatures de la partie commençant en TRAP de telle sorte que ces instructions fassent "perdre" à chaque résultat élémentaire un nombre de chiffres binaires égal au paramètre effectif du "premier ordre" d'exécution de TRONC ou à l'argument du "premier" CALL TRONC ; (dans le cas de TROFXS, cette fonction a) n'existe pas) ;

b) de mettre en réserve le contenu des mémoires 0 et 2 parce que ces mémoires seront utilisées par les déroutements en TRAP ;

c) de mettre dans la mémoire 2 une instruction servant aux déroutements en TRAP

d) de revenir au programme donné juste après le "premier" ordre d'exécution de TRONC ou TROFXS ou juste après le "premier" CALL TRONC ou CALL TROFXS ("premier" au sens de premier d'une "paire").

4) Partie commençant en E3

L'exécution de cette partie a pour effet :

a) de remettre dans les mémoires 0 et 2 ce qui y était avant l'exécution de la partie commençant en E2,

b) de revenir au programme donné juste après l'ordre de déroute ment en E3.

5) Partie commençant en TRAP

L'exécution de cette partie a pour effet :

a) d'examiner un code figurant dans l'ordre de transfert en TRAP et d'en déduire de quel type d'opération arithmétique il s'agit

b) d'exécuter normalement cette opération arithmétique flottante

c) dans le cas de TROFXS de calculer le nombre de chiffres binaires qui doivent être "perdus" par le résultat élémentaire ; ce nombre est donné par : Min (27, Max (0,27-(E+J))) ou E est l'exposant binaire du résultat élémentaire et J le paramètre effec tif du "1er" ordre d'exécution de TROFXS ; (dans le cas de TRONC cette fonction c) n'existe pas, le nombre de chiffres binaires à "perdre" ayant été calculé une fois pour toutes en a) de la partie commençant en E2) ;

d) de faire perdre au résultat élémentaire le nombre de chiffres binaires voulu

e) de revenir au programme donné juste après l'ordre de déroute ment en TRAP.

6) Exécution "pour la 1ère fois" d'une séquence "encadrée" du programme donné

Lorsqu'une séquence où l'on veut grossir les erreurs d'arrondi est exécutée pour la 1ère fois ;

a) le "premier ordre d'exécution de TRONC ou TROFXS ou le "premier" CALL TRONC ou CALL TROFXS fait exécuter successivement la partie commençant en TRONC, TROFXS ou E1 et la partie commençant en E2 ; c'est-à-dire qu'il fait remplacer les opérations arithmétiques flottantes de la séquence par des ordres de déroutement en TRAP, et qu'il fait remplacer le "premier" et le "second" ordre d'exécution de TRONC ou TROFXS ou le "premier" et le "second CALL TRONC ou CALL TROFXS par des ordres de déroutement respectivement en E2 pour le premier et en E3 pour le second; dans le cas de TRONC, il prépare les opérations de Troncature de TRAP ; enfin il met en réserve les mémoires 0 et 2 ;

b) pendant l'exécution de la séquence en question du programme donné, les instructions arithmétiques flottantes, qui ont été remplacées par des ordres de déroutement en TRAP ont pour effet d'exécuter les opérations et de faire perdre des chiffres à chaque résultat ;

c) le "second" ordre d'exécution de TRONC ou TROFXS ou le "second" CALL TRONC ou CALL TROFXS, qui a été remplacé par un ordre déroutement en E3 a pour effet de remetre dans les mémoires 0 et 2 ce qui y était avant la "première" exécution de TRONC ou TROFXS.

7) Exécution "pour la $n^{ième}$ fois" d'une séquence transformée du programme donné

Lorsque, par suite d'une boucle ou pour toute autre raison, l'exécution du programme donné passe pour la 2e fois, ou pour la $n^{ième}$ fois ($n > 1$), sur une séquence où l'on veut grossir les erreurs d'arrondi ;

a) le "premier" ordre d'exécution de TRONC ou TROFXS ou le "premier" CALL TRONC ou CALL TROFXS provoque un déroutement direct en E2 et évite l'examen de toutes les instructions de la séquence puisque celles parmi ces instructions qui devaient être remplacées par des ordres de déroutements en TRAP l'ont déjà été ; on assure uniquement la mise en réserve du contenu des mémoires 0 et 2 et, dans le cas de TRONC, la détermination du nombre de chiffres binaires à "perdre" ;

b) l'exécution de la séquence du programme donné dans laquelle on veut grossir les erreurs d'arrondi, et l'exécution du "second" appel de TRONC ou TROFXS se font exactement comme lors de l'exécution"pour la 1ère fois", le nombre de chiffres perdus pouvant seul, éventuellement, être différent.

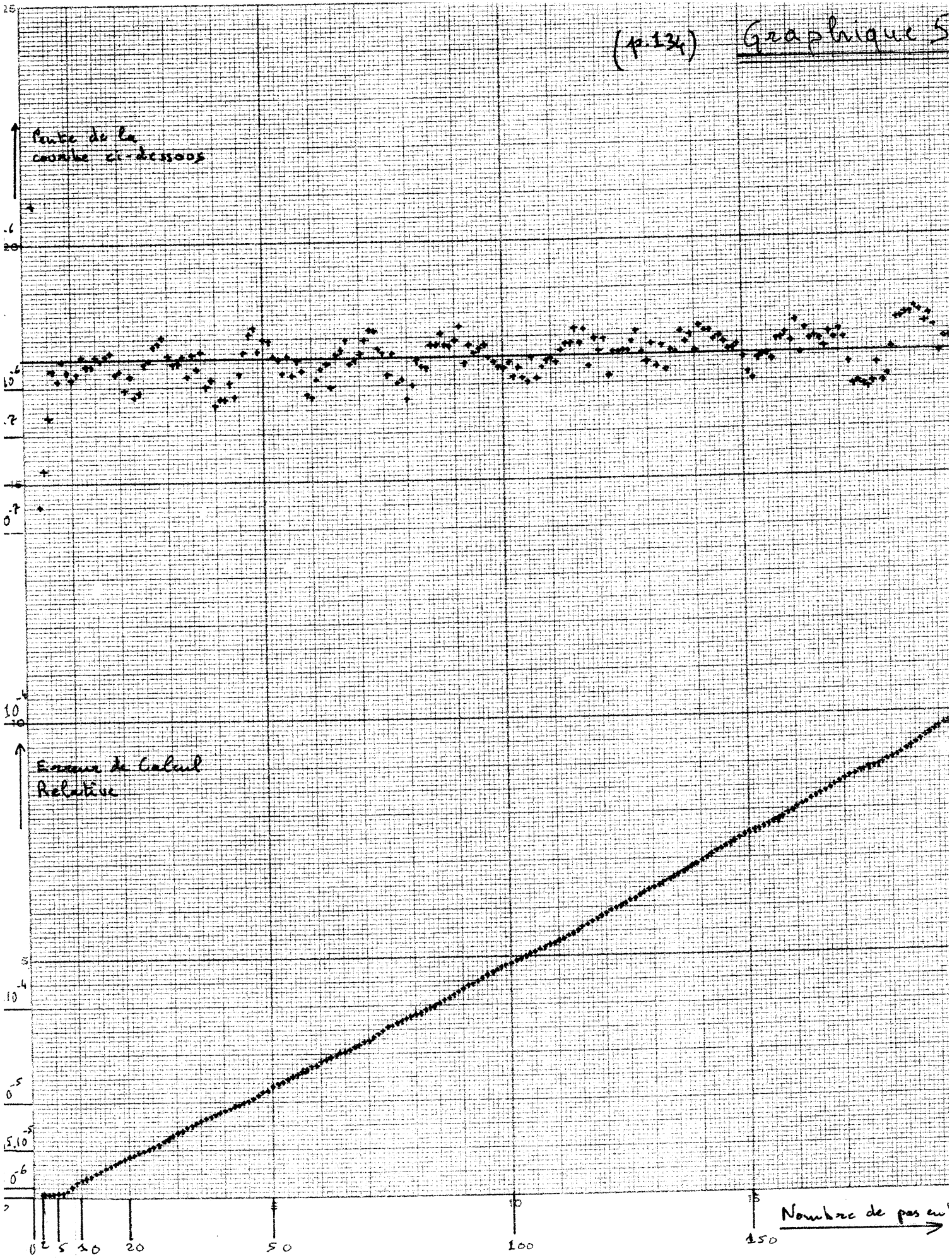
ORGANIGRAMME RESUME DE TRONC ET TROFXS -

TRONC ou TROFXS ou E1	Met en réserve le contenu de l'Index 4 et l'instruction d'appel
	Remplace l'appel de TRONC ou TROFXS qui vient d'être exécuté par un ordre de déroutement en E2
	Examine dans l'ordre de séquence et en utilisant l'Index4 toutes les instructions suivant celles correspondant à l'appel de TRONC ou de TROFXS qui vient d'être exécuté. Remplace dans cette séquence chaque instruction arithmétique flottante par un STR c'est-à-dire par un ordre de déroutement à l'instruction contenue en mémoire 2 ; met également dans ces instructions des bits repères en position 15 à 17 pour repérer de quelle opération il s'agit
	Remplace le 2e appel de TRONC ou de TROFXS par un ordre de déroutement en E3 et restaure l'Index4
E2	Dans le cas de TRONC, détermine le nombre de chiffres binaires à faire perdre aux résultats élémentaires
	Met en réserve le continu des mémoires 0 et 2 et met en mémoire 2 une instruction servant pour TRAP
	Retour au programme donné après le 1er appel de TRONC ou TROFXS
E3	Remet dans les mémoires 0 et 2 ce qui était avant l'exec. de E2
	Retour au programme donné après le 2e appel de TRONC ou TROFXS
TRAP	Examine les positions 15 à 17 de l'instruction STR pour déterminer de quelle opération arithmétique il s'agit Exécute cette opération Dans le cas de TROFXS, détermine le nombre de chiffres à "perdre" Fait perdre le nombre de chiffres voulu au résultat élémentaire Retour dans le programme donné après le STR.

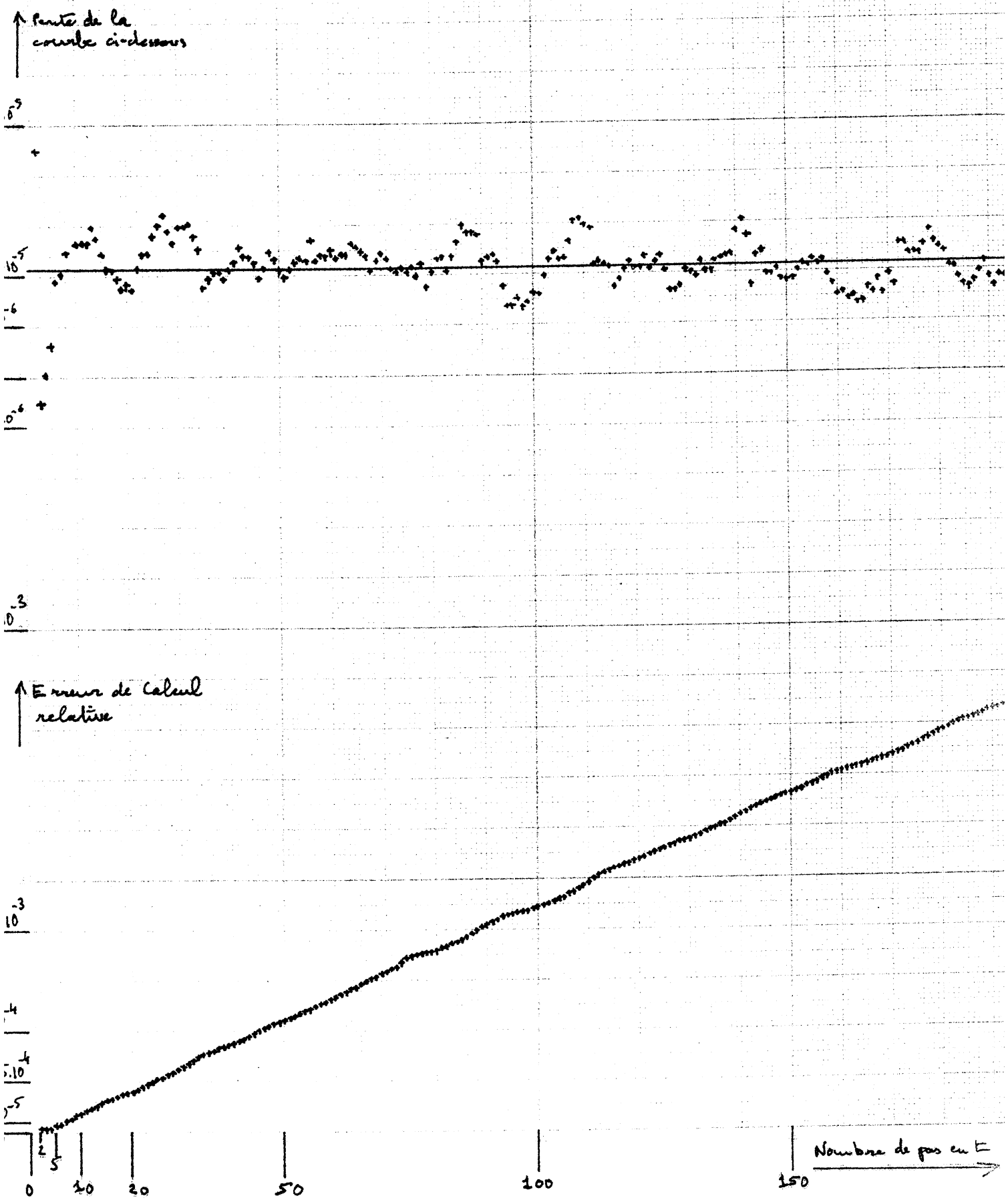
- A N N E X E S -

(p. 134)

Graphique 5

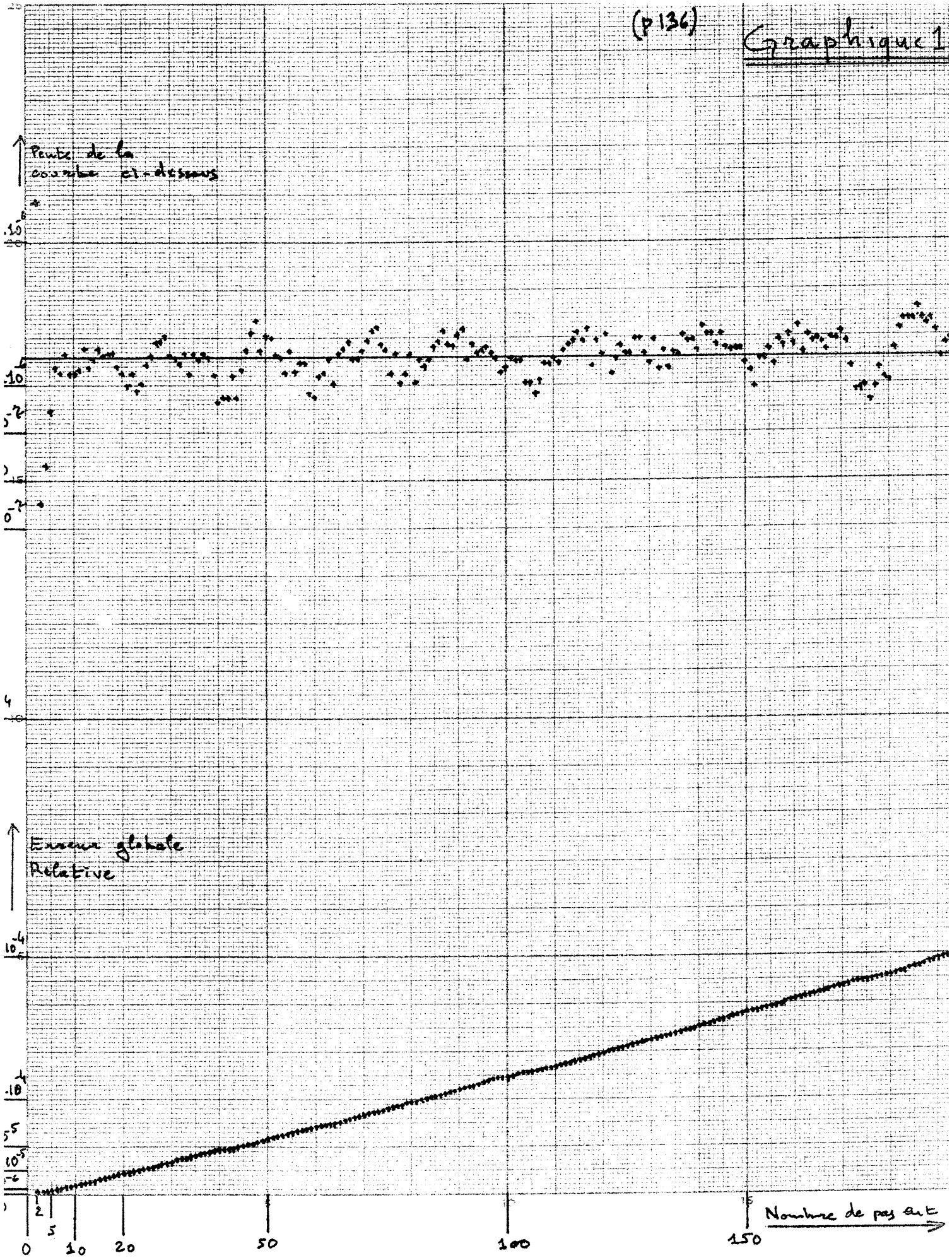


Graphique 3


Pente de la courbe ci-dessous
10^-5
10^-6
10^-3
Erreur de calcul relative
10^-3
10^-4
5.10^-4
10^-5
2
5
0
10
20
50
100
150
Nombre de pas en E

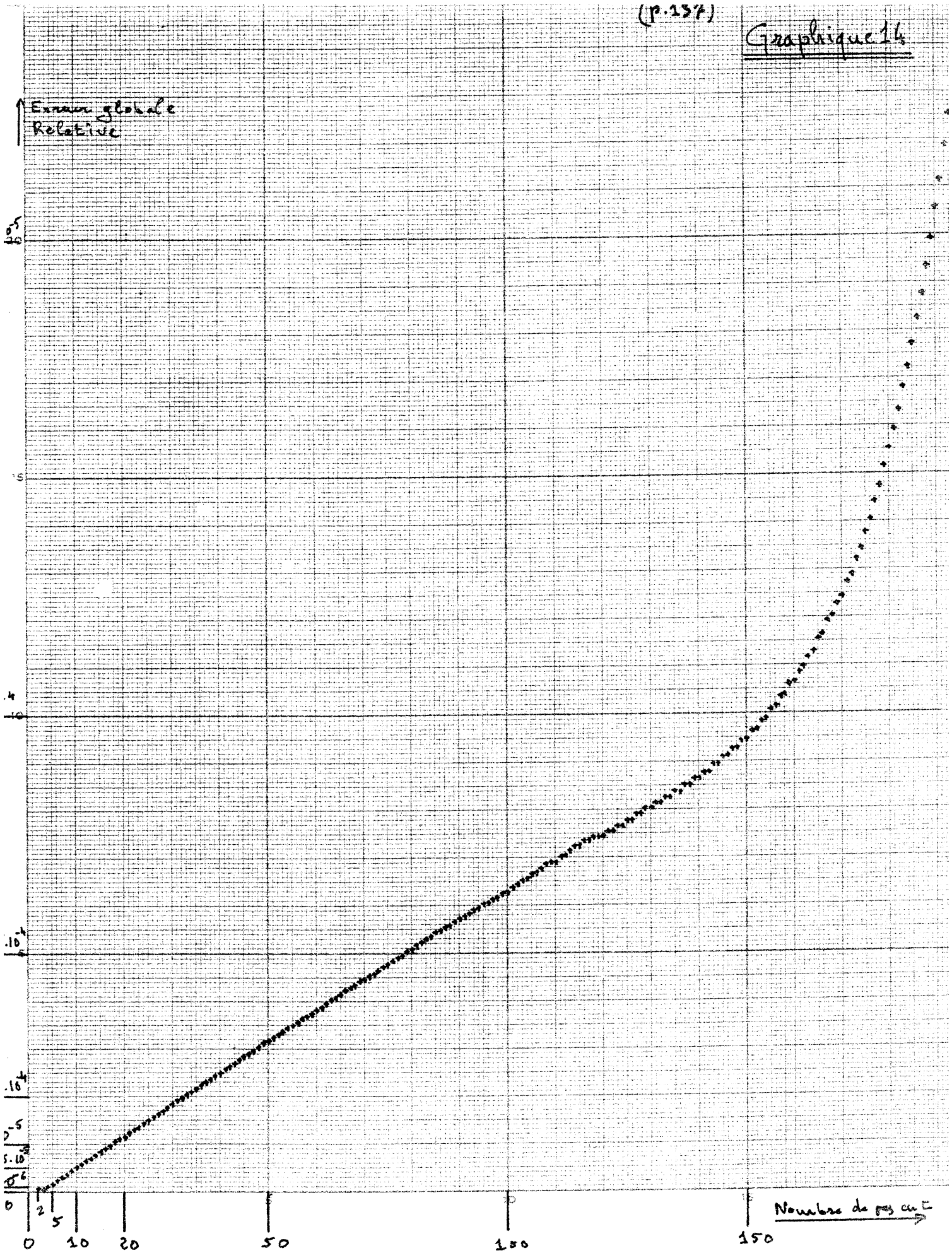
(p 136)

Graphique 1



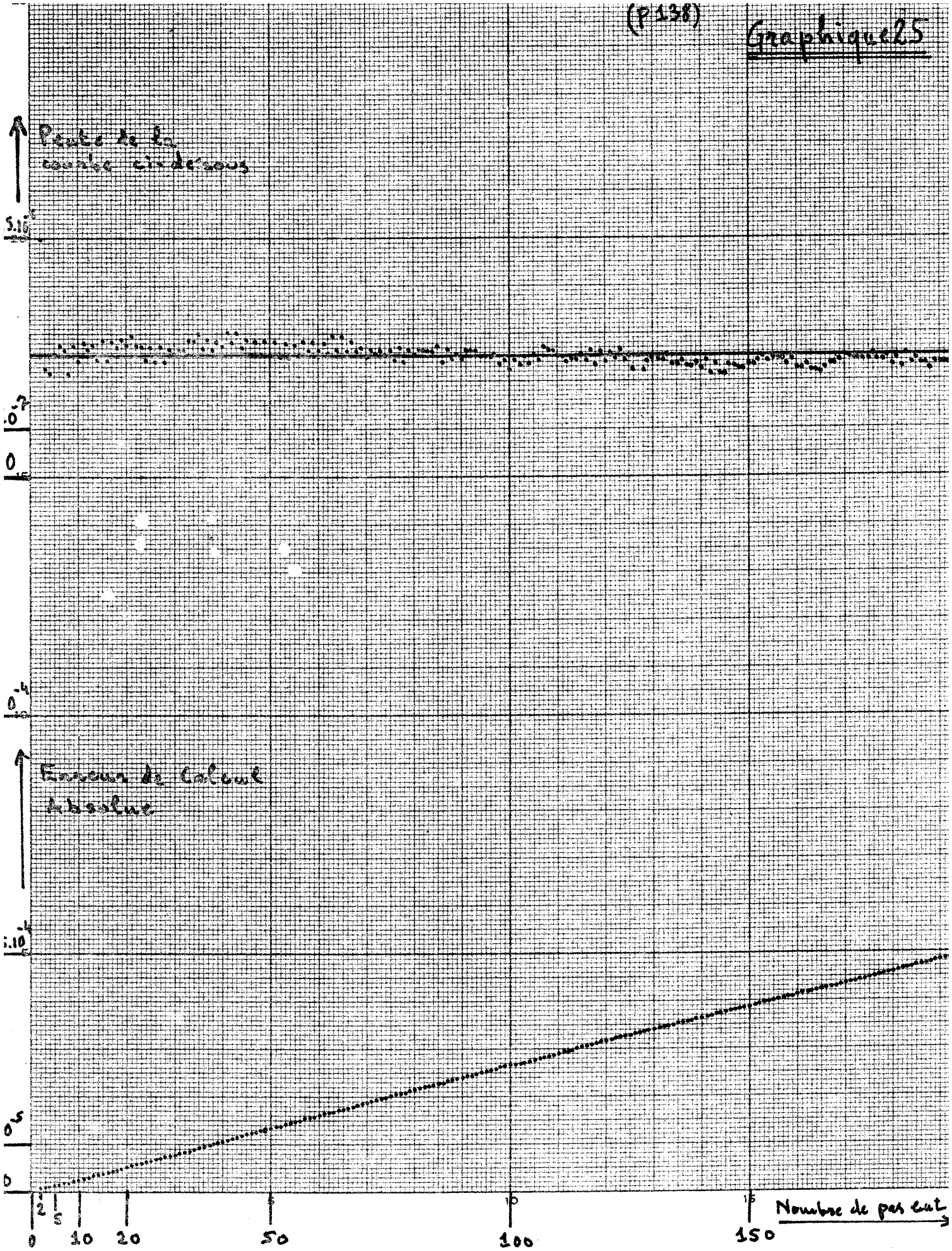
(p.137)

Graphique 14

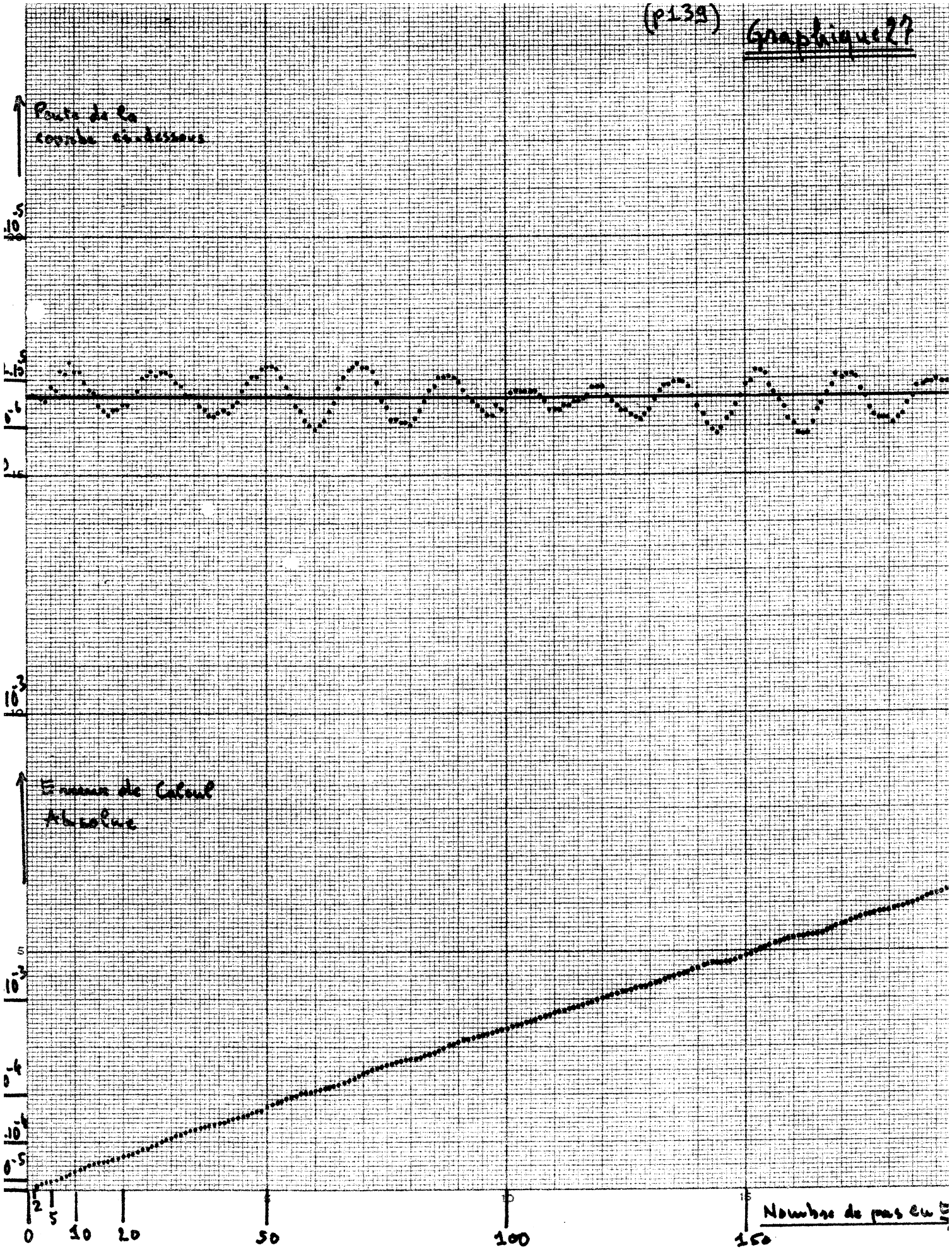


(P 138)

Graphique 25

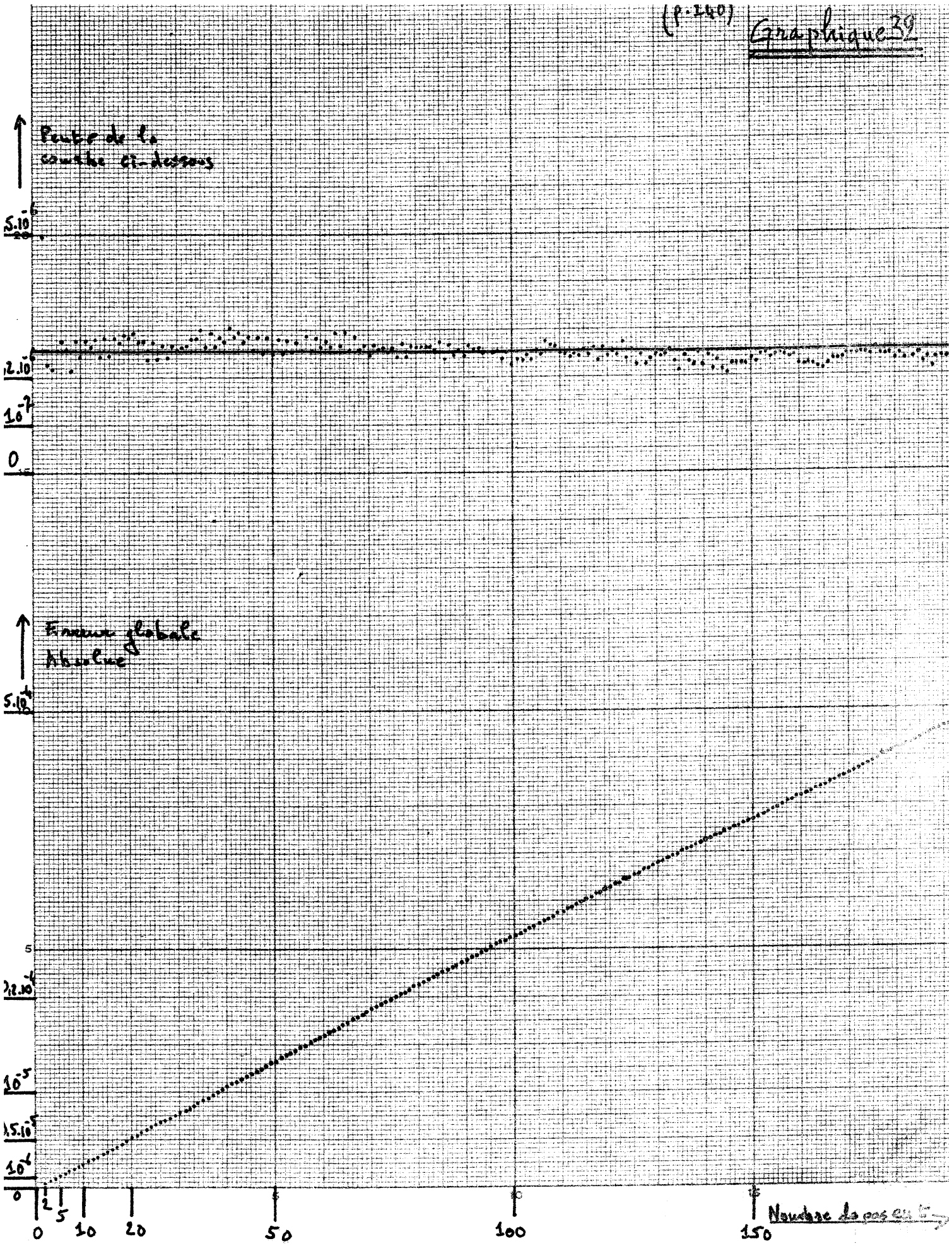


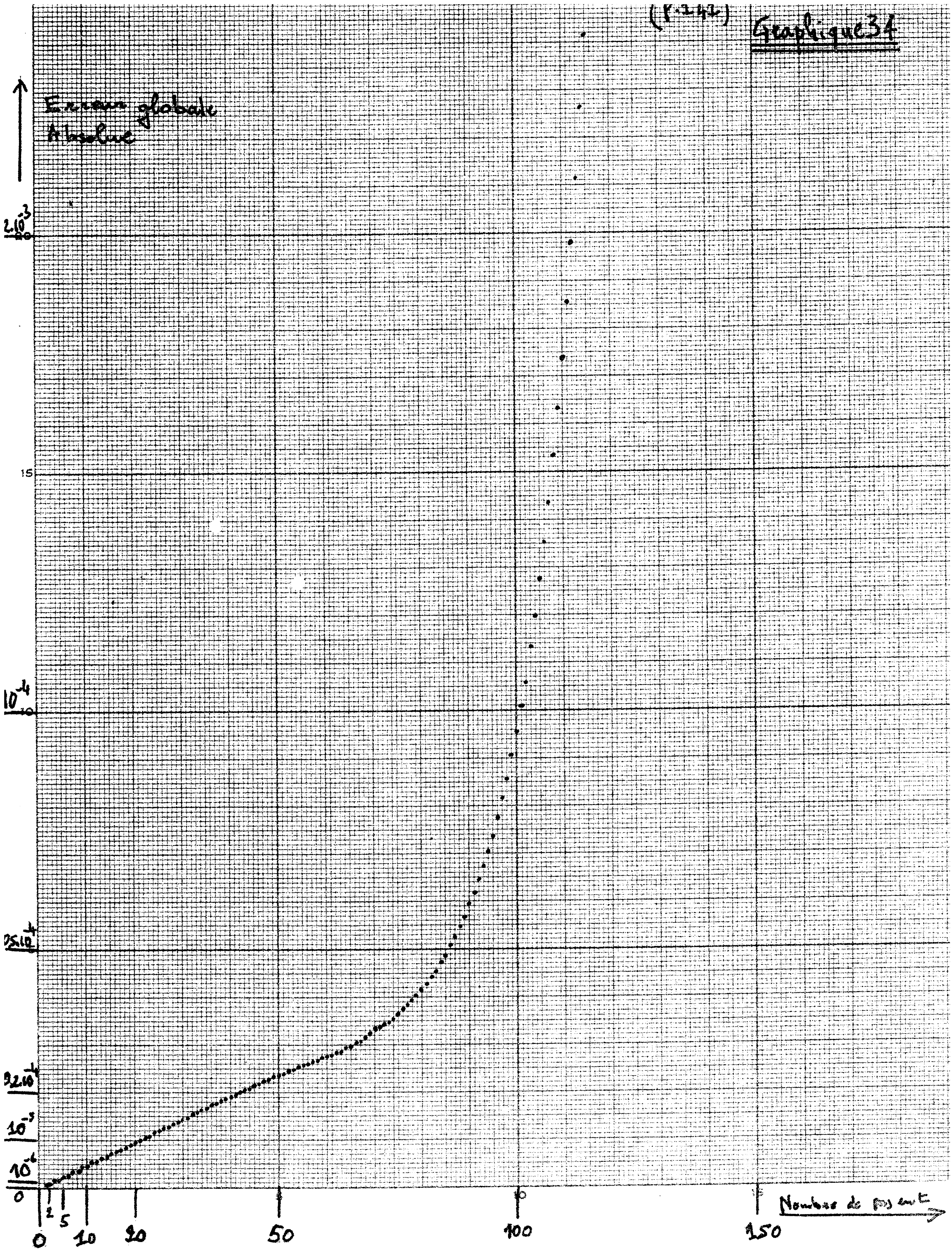
(p139) Graphique 27

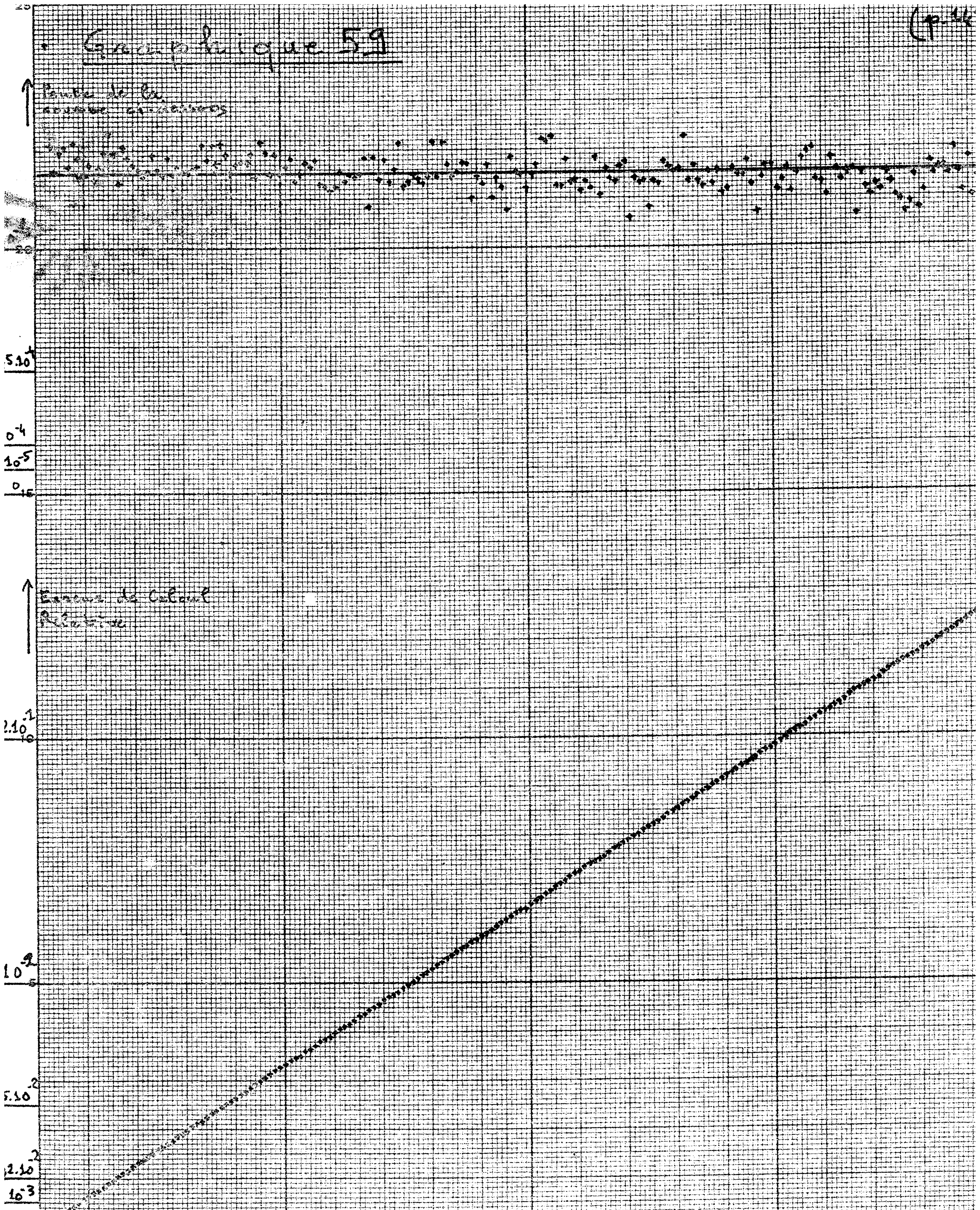


(p. 240)

Graphique 39

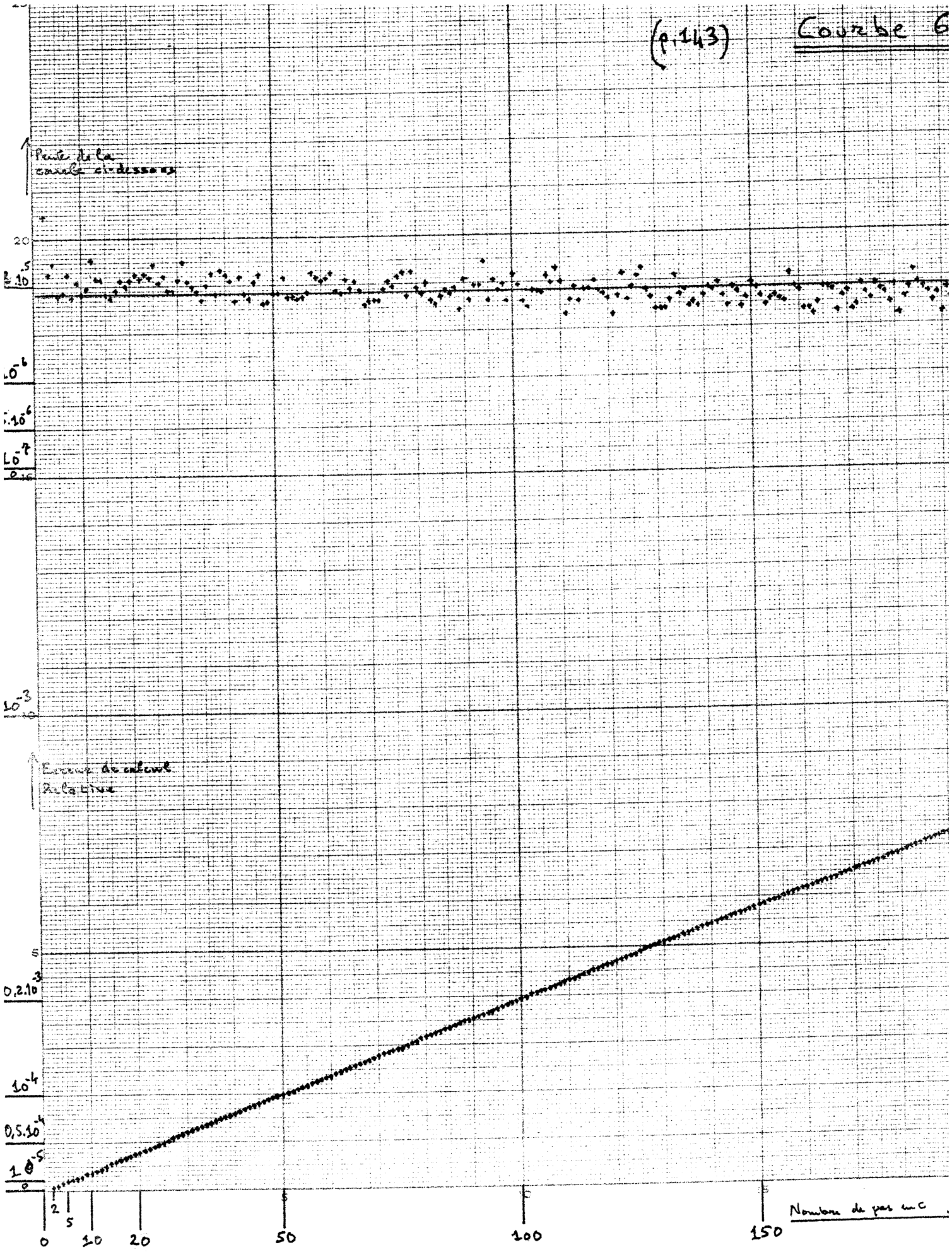



Graphique 34
Erreur globale Absolue
2.10^-3
15
10^-4
0,5.10^-4
2.10^-4
10^-5
10^-6
0
2
5
0
10
20
50
100
150



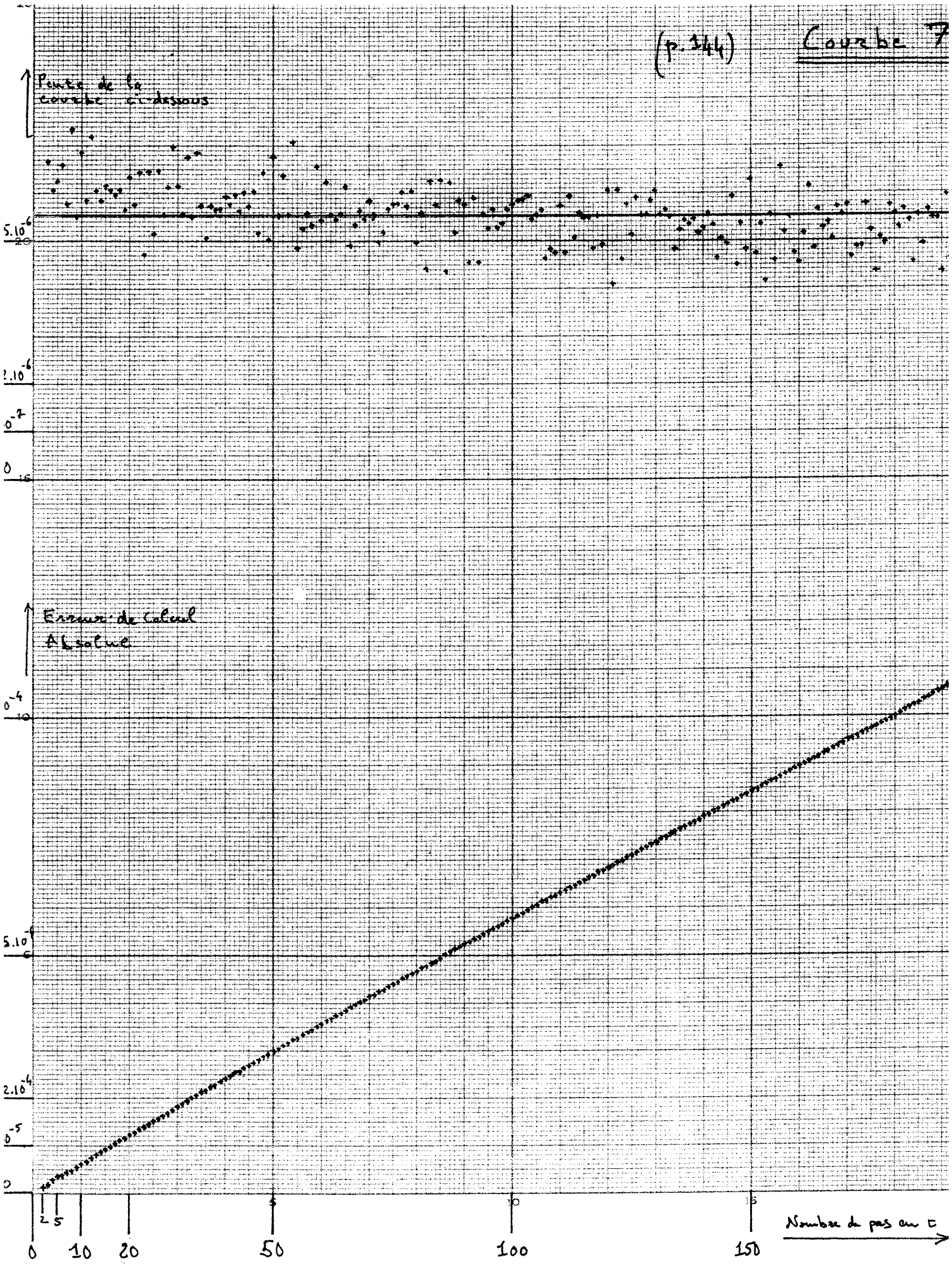
Graphique 59
Pente de la courbe ci-dessous
Erreur de Calcul Relative
5.10^-4
10^-4
10^-5
0
2.10^-1
10^-2
5.10^-2
2.10^-2
10^-3
0
0
2
5
10
20
50
100
150
Nombre de pas en t

(p.143) Courbe 6



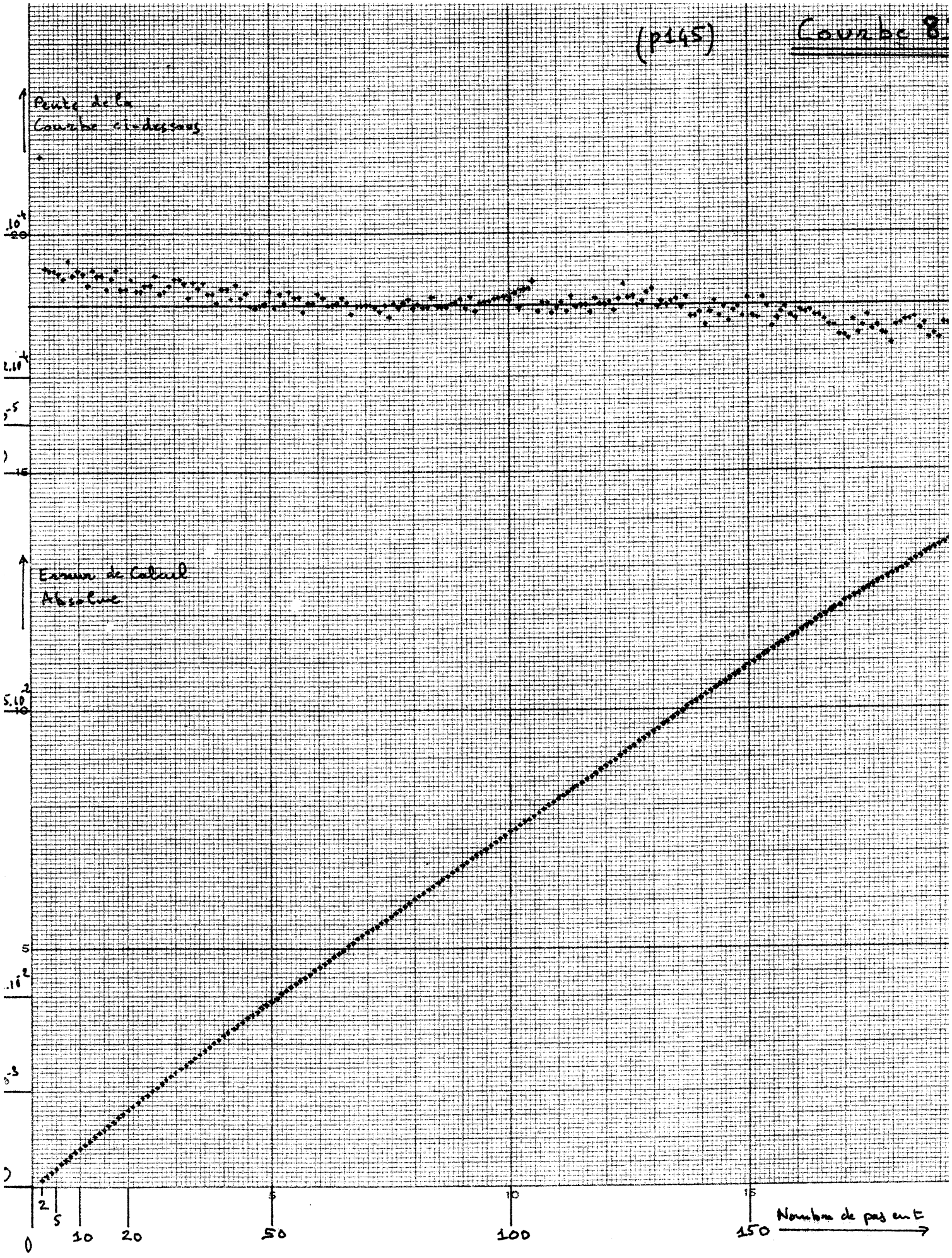
(p. 144)

Courbe 7



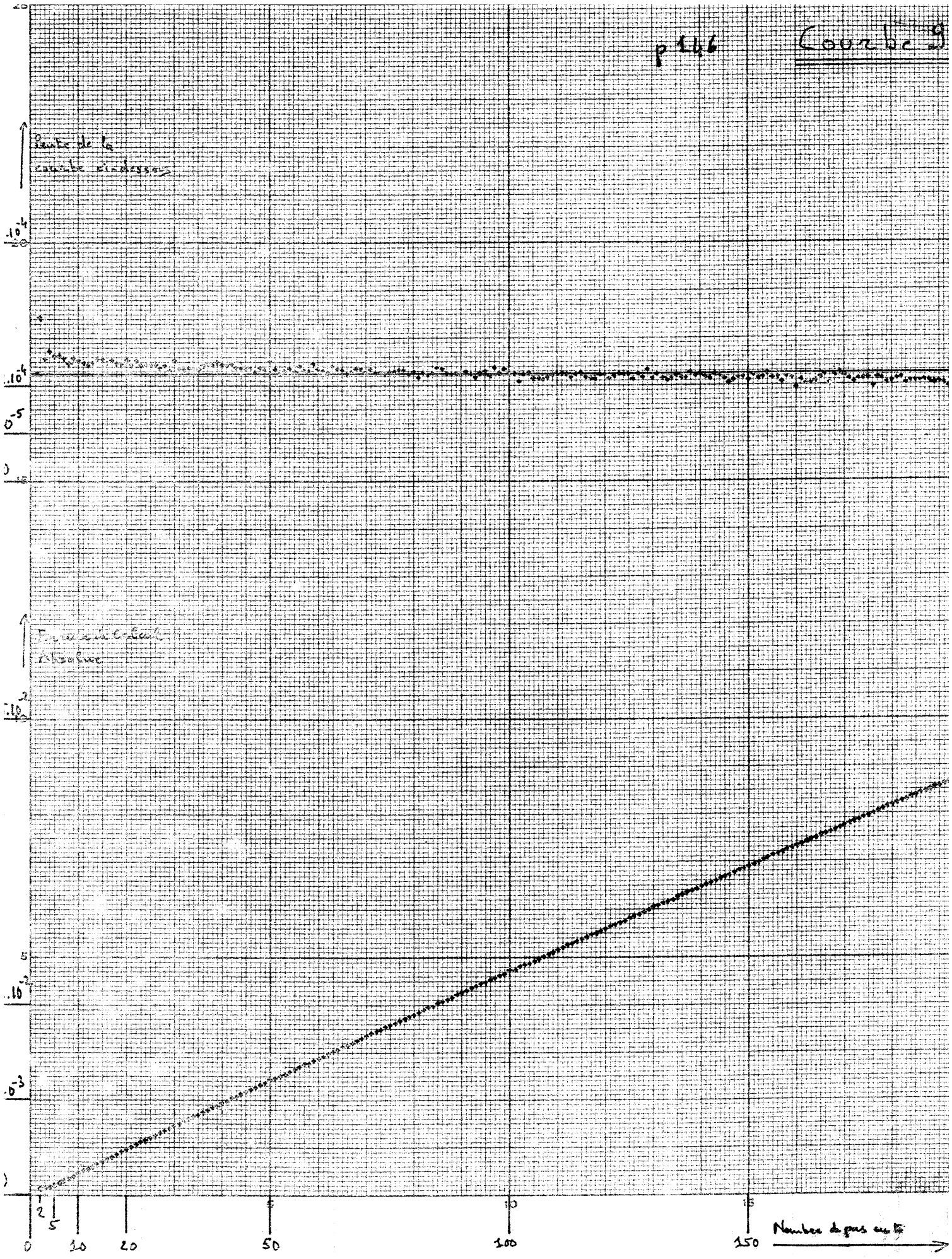
(p145)

Courbe 8



p 146

Courbe 9



```
    'PROCEDURE' TRONC(ITRON) :: 'VALEUR' ITRON :: 'ENTIER' ITRON ::
    'DEBUT'
    'PROCEDURE' E1(I) :: 'VALEUR' I :: 'ENTIER' I ::                    'CODE
        SXA     RES,4
        LAC     P1,4
        CAL     -1,4
        SLW     RES+1
        CAL     ADE2
        STA     -1,4
PROG    CAL     0,4
        LAS     RES+1
        TRA     *+2
        TRA     FIN
        LGR     24
        LAS     =0302
        TXI     PROG,4,-1
        TRA     SOU
        LAS     =0300
        TXI     PROG,4,-1
        TRA     ADD
        LAS     =0260
        TXI     PROG,4,-1
        TRA     MUL
        LAS     =0241
        TXI     PROG,4,-1
        TRA     DIV
        TXI     PROG,4,-1
SOU     CAL     =05000
        LGL     24
        TRA     REGR
ADD     CAL     =05000
        LGL     24
        ORA     =01000000
        TRA     REGR
MUL     CAL     =05000
        LGL     24
        ORA     =02000000
        TRA     REGR
DIV     CAL     =05000
        LGL     24
        ORA     =04000000
REGR    SLW     0,4
        TXI     PROG,4,-1
TRAP    STQ     RES+4
        STO     RES+5
        SXD     RES,4
        LAC     0,4
        CAL     -1,4
        LGR     18
        LBT
        TRA     *+2
        TRA     ADDI
        LGR     1
        LBT
```

```
        TRA     *+2
        TRA     MULT
        LGR     1
        LBT
        TRA     SOUS
DIVI    ANA     =O14
        ORA     =O5020
        LGL     20
        SLW     OP
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        LGR     **
        LGL     **
        TRA*    0
SOUS    ANA     =O14
        ORA     =O6040
        LGL     20
RETOUR  SLW     OP
        LDQ     RES+4
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        ARS     **
        ALS     **
        TRA*    0
MULT    ANA     =O30
        ORA     =O13000
        LGL     19
        TRA     RETOUR
ADDI    ANA     =O60
        ORA     =O30000
        LGL     18
        TRA     RETOUR
ADE2    TSL     P3
ADE3    TSL     P4
TRA     TRA     TRAP
OP      PZE
RES     BSS     6
FIN     CAL     ADE3
        STA     0,4
        LXA     RES,4
'FCODE'         ::
        'PROCEDURE' E2(I) :: 'VALEUR' I :: 'ENTIER' I ::          'CODE'
        SXA     RES,4
LDIT    CLA     F3+1
        STA     DIVI+7
        STA     DIVI+8
        STA     RETOUR+5
        STA     RETOUR+6
        LDQ     0
        STQ     RES+2
        LDQ     2
        STQ     RES+3
```

```
        LDQ     TRA
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
'FCODE'         ::
          'PROCEDURE' E3 (I) :: 'VALEUR' I :: 'ENTIER' I ::          'CODE'
        LDQ     RES+2
        SXA     RES,4
        TSX     S.SCCR,4
        STQ     0
        LDQ     RES+3
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
'FCODE'         ::
          E1(ITRON) :: E2(ITRON) ::
    'FIN' ::
```

```
    'PROCEDURE' TROFXS(ITRON) :: 'VALEUR' ITRON :: 'ENTIER' ITRON ::
    'DEBUT'
    'PROCEDURE' E1(I) :: 'VALEUR' I :: 'ENTIER' I ::                    'CODE'
        SXA     RES,4
        LAC     P1,4
        CAL     -1,4
        SLW     RES+1
        CAL     ADE2
        STA     -1,4
PROG    CAL     0,4
        LAS     RES+1
        TRA     *+2
        TRA     FIN
        LGR     24
        LAS     =O302
        TXI     PROG,4,-1
        TRA     SOU
        LAS     =O300
        TXI     PROG,4,-1
        TRA     ADD
        LAS     =O260
        TXI     PROG,4,-1
        TRA     MUL
        LAS     =O241
        TXI     PROG,4,-1
        TRA     DIV
        TXI     PROG,4,-1
SOU     CAL     =O5000
        LGL     24
        TRA     REGR
ADD     CAL     =O5000
        LGL     24
        ORA     =O1000000
        TRA     REGR
MUL     CAL     =O5000
        LGL     24
        ORA     =O2000000
        TRA     REGR
DIV     CAL     =O5000
        LGL     24
        ORA     =O4000000
REGR    SLW     0,4
        TXI     PROG,4,-1
TRAP    STQ     RES+4
        STO     RES+5
        SXD     RES,4
        LAC     0,4
        CAL     -1,4
        LGR     18
        LBT
        TRA     *+2
        TRA     ADDI
        LGR     1
        LBT
```

```
        TRA     *+2
        TRA     MULT
        LGR     1
        LBT
        TRA     SOUS
DIVI    ANA     =014
        ORA     =05020
        LGL     20
        SLW     OP
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        STQ     RES+4
        STO     RES+5
        LLS     8
        ANA     =0377
        ADD     F3+1
        SUB     =155
        TMI     DC027
        ZAC
        TRA     DTRON
DC027   CAS     =-27
        TRA     DTRON
        TRA     DZERO
DZERO   PXD     ,
        LDQ     =0
        TRA*    0
DTRON   STA     *+4
        STA     *+4
        CLA     RES+5
        LDQ     RES+4
        LGR     **
        LGL     **
        TRA*    0
SOUS    ANA     =014
        ORA     =06040
        LGL     20
RETOUR  SLW     OP
        LDQ     RES+4
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        STO     RES+5
        ARS     27
        ANA     =0377
        ADD     F3+1
        SUB     =155
        TMI     C027
        PXD     ,
        TRA     TRON
C027    CAS     =-27
        TRA     TRON
        TRA     REZERO
REZERO  PXD     ,
```

```
        TRA*    0
TRON    STA     *+3
        STA     *+3
        CLA     RES+5
        ARS     **
        ALS     **
        TRA*    0
MULT    ANA     =030
        ORA     =013000
        LGL     19
        TRA     RETOUR
ADDI    ANA     =060
        ORA     =030000
        LGL     18
        TRA     RETOUR
ADE2    TSL     P3
ADE3    TSL     P4
TRA     TRA     TRAP
OP      PZE
RES     BSS     6
FIN     CAL     ADE3
        STA     0,4
        LXA     RES,4
'FCODE'         ::
          'PROCEDURE' E2(I) :: 'VALEUR' I :: 'ENTIER' I ::          'CODE'
        SXA     RES,4
        LDQ     0
        STQ     RES+2
        LDQ     2
        STQ     RES+3
        LDQ     TRA
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
'FCODE'         ::
          'PROCEDURE' E3 (I) :: 'VALEUR' I :: 'ENTIER' I ::         'CODE'
        LDQ     RES+2
        SXA     RES,4
        TSX     S.SCCR,4
        STQ     0
        LDQ     RES+3
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
'FCODE'         ::
          E1(ITRON) :: E2(ITRON) ::
    'FIN' ::
```

```
*       SP DE TRONCATURE DES OPERATIONS FLOTTANTES
        ENTRY   TRONC
TRONC   PZE     **
        SXA     RES,4
        LAC     TRONC,4
        CAL     -1,4
        SLW     RES+1
        CAL     ADE2
        STA     -1,4
PROG    CAL     0,4
        LAS     RES+1
        TRA     *+2
        TRA     FIN
        LGR     24
        LAS     =0302
        TXI     PROG,4,-1
        TRA     SOU
        LAS     =0300
        TXI     PROG,4,-1
        TRA     ADD
        LAS     =0260
        TXI     PROG,4,-1
        TRA     MUL
        LAS     =0241
        TXI     PROG,4,-1
        TRA     DIV
        TXI     PROG,4,-1
SOU     CAL     =05000
        LGL     24
        TRA     REGR
ADD     CAL     =05000
        LGL     24
        ORA     =01000000
        TRA     REGR
MUL     CAL     =05000
        LGL     24
        ORA     =02000000
        TRA     REGR
DIV     CAL     =05000
        LGL     24
        ORA     =04000000
REGR    SLW     0,4
        TXI     PROG,4,-1
FIN     CAL     ADE3
        STA     0,4
        CAL     TRONC
        STA     E2
        TRA     E2+2
E2      PZE     **
        SXA     RES,4
        LAC     E2,4
        CAL*    2,4
        STA     DIVI+7
        STA     DIVI+8
```

```
        STA     RETOUR+5
        STA     RETOUR+6
        LDQ     0
        STQ     RES+2
        LDQ     2
        STQ     RES+3
        LDQ     TRA
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
        TRA*    E2
E3      PZE     **
        SXA     RES,4
        LDQ     RES+2
        TSX     S.SCCR,4
        STQ     0
        LDQ     RES+3
        TSX     S.SCCR,4
        STQ     2
        LXA     RES,4
        TRA*    E3
TRAP    STQ     RES+4
        STO     RES+5
        SXD     RES,4
        LAC     0,4
        CAL     -1,4
        LGR     18
        LBT
        TRA     *+2
        TRA     ADDI
        LGR     1
        LBT
        TRA     *+2
        TRA     MULT
        LGR     1
        LBT
        TRA     SOUS
 DIVI   ANA     =014
        ORA     =05020
        LGL     20
        SLW     OP
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        LGR     **
        LGL     **
        TRA*    0
 SOUS   ANA     =014
        ORA     =06040
        LGL     20
RETOUR  SLW     OP
        LDQ     RES+4
        CLA     RES+5
        LXD     RES,4
```

```
        XEC     OP
        ARS     **
        ALS     **
        TRA*    0
MULT    ANA     =030
        ORA     =013000
        LGL     19
        TRA     RETOUR
ADDI    ANA     =060
        ORA     =030000
        LGL     18
        TRA     RETOUR
ADE2    TSL     E2
ADE3    TSL     E3
TRA     TRA     TRAP
OP      PZE
RES     BSS     6
        END     TRONC
```

```
*       SP DE SIMULATION DE TRONCATURE FIXE
        ENTRY   TROFXS
TROFXS  PZE
        SXA     RES,4
        LAC     TROFXS,4
        CAL     -1,4
        SLW     RES+1
        CAL     ADE2
        STA     -1,4
 PROG   CAL     0,4
        LAS     RES+1
        TRA     *+2
        TRA     FIN
        LGR     24
        LAS     =O302
        TXI     PROG,4,-1
        TRA     SOU
        LAS     =O300
        TXI     PROG,4,-1
        TRA     ADD
        LAS     =O260
        TXI     PROG,4,-1
        TRA     MUL
        LAS     =O241
        TXI     PROG,4,-1
        TRA     DIV
        TXI     PROG,4,-1
  SOU   CAL     =O5000
        LGL     24
        TRA     REGR
  ADD   CAL     =O5000
        LGL     24
        ORA     =O1000000
        TRA     REGR
  MUL   CAL     =O5000
        LGL     24
        ORA     =O2000000
        TRA     REGR
  DIV   CAL     =O5000
        LGL     24
        ORA     =O4000000
 REGR   SLW     0,4
        TXI     PROG,4,-1
  FIN   CAL     ADE3
        STA     0,4
        CAL     TROFXS
        STA     E2
        TRA     E2+2
   E2   PZE
        SXA     RES,4
        LAC     E2,4
        CAL*    2,4
        STO     PRECIB
        LDQ     0
```

```
        STQ     RES+2
        LDQ     2
        STQ     RES+3
        LDQ     TRA
        TSX     S.SCOR,4
        STQ     2
        LXA     RES,4
        TRA*    E2
   E3   PZE
        LDQ     RES+2
        SXA     RES,4
        TSX     S.SCOR,4
        STQ     0
        LDQ     RES+3
        TSX     S.SCOR,4
        STQ     2
        LXA     RES,4
        TRA*    E3
TRAP    STQ     RES+4
        STO     RES+5
        SXD     RES,4
        LAC     0,4
        CAL     -1,4
        LGR     18
        LBT
        TRA     *+2
        TRA     ADDI
        LGR     1
        LBT
        TRA     *+2
        TRA     MULT
        LGR     1
        LBT
        TRA     SOUS
DIVI    ANA     =014
        ORA     =05020
        LGL     20
        SLW     OP
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        STQ     RES+4
        STO     RES+5
        LLS     8
        ANA     =0377
        ADD     PRECIB
        SUB     =155
        TMI     DC027
        ZAC
        TRA     DTRON
DC027   CAS     =-27
        TRA     DTRON
        TRA     DZERO
DZERO   PXD     ,
```

```
        LDQ     =0
        TRA*    0
 DTRON  STA     *+4
        STA     *+4
        CLA     RES+5
        LDQ     RES+4
        LGR     **
        LGL     **
        TRA*    0
 SOUS   ANA     =014
        ORA     =0604C
        LGL     20
RETOUR  SLW     OP
        LDQ     RES+4
        CLA     RES+5
        LXD     RES,4
        XEC     OP
        STO     RES+5
        ARS     27
        ANA     =0377
        ADD     PRECIB
        SUB     =155
        TMI     C027
        PXD     ,
        TRA     TRON
 C027   CAS     =-27
        TRA     TRON
        TRA     REZERO
REZERO  PXD     ,
        TRA*    0
 TRON   STA     *+3
        STA     *+3
        CLA     RES+5
        ARS     **
        ALS     **
        TRA*    0
 MULT   ANA     =030
        ORA     =013000
        LGL     19
        TRA     RETOUR
 ADDI   ANA     =060
        ORA     =030000
        LGL     18
        TRA     RETOUR
 ADE2   TSL     E2
 ADE3   TSL     E3
 TRA    TRA     TRAP
   OP   PZE
  RES   BSS     6
PRECIB  BSS     1
        END
```

```
C       EQUA CHALEUR METH. EXPLIC. 1 VAR D ESP ETUDE ERREUR (6 COMP.)
        DIMENSION UP(401),UG(201),UV1(401),UV2(401),UF1(401),UF2(401),
       1E(401),UD(201),UB(401),SA(401),EM(3,4),S(3,4,3)   ,PT(2)
        EXTERNAL FUTF
  240   READ(5,20)BXG,BTB,DX,DT,SIGMA,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
  248   NXZ=NX-1
        NTMOI = (NT-1)/2+1
        FNX2=FLOAT(NX-2)
        COEF=DT/DX**2
        WRITE(6,63)
        WRITE(6,15)BXG,BTB,DX,DT,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
        WRITE(6,17)COEF
  250   IF(IFUB)253,253,256
  253   READ(5,30)(UB(I),I=1,NX)
        GO TO 262
  256   DO 259 I=1,NX
  259   UB(I)=FUTF(BXG+DX*FLOAT(I-1),BTB)
  262   IF(IFUG)265,265,268
  265   READ(5,30)(UG(I),I=1,NT)
        GO TO 273
  268   DO 271 I=1,NT
  271   UG(I)=FUTF(BXG,BTB+DT*FLOAT(I-1))
  273   IF(IFUD)276,276,279
  276   READ(5,30)(UD(I),I=1,NT)
        GO TO 320
  279   DO 282 I=1,NT
  282   UD(I)=FUTF(BXG+DX*FLOAT(NX-1),BTB+DT*FLOAT(I-1))
  320   DO 325 I=2,NXZ
  321   UV1(I)=UB(I)
  325   UF1(I)=UB(I)
        UV1(1)=UG(1)
        UF1(1)=UG(1)
        UV1(NX)=UD(1)
        UF1(NX)=UD(1)
        IF(IW.EQ.0) GO TO 327
        WRITE(6,27)
        WRITE(6,75)(UV1(I),I=1,NX)
  327   DO 510 K=2,NT
        ID=1
        IDP=0
        IDL=0
        IF((K.EQ.2).OR.(K.EQ.NTMOI).OR.(K.EQ.NT)) IDL=2
        IF((IDL.EQ.2).AND.(IW.NE.0)) ID=2
        IF((IDL.EQ.2).OR.(IW.NE.0)) IDP=2
        TV=BTB+DT*FLOAT(K-1)
        DO 330 I=2,NXZ
  330   UV2(I)=UV1(I)+COEF*(UV1(I-1)-2.*UV1(I)+UV1(I+1))
        UV2(1)=UG(K)
        UV2(NX)=UD(K)
        IF(IDP.EQ.2) WRITE(6,34) K,TV
        GO TO (9338, 335),ID
  335   WRITE(6,33)K,TV
        WRITE(6,75)(UV2(I),I=1,NX)
 9338   CALL TRONC(ITRON)
```

```
      DO 340 I=2,NXZ
  340 UF2(I)=UF1(I)+COEF*(UF1(I-1)-2.*UF1(I)+UF1(I+1))
 9342 CALL TRONC(ITRON)
      UF2(1)=UG(K)
      UF2(NX)=UD(K)
      GO TO ( 344, 342),ID
  342 WRITE(6,35)ITRON,K,TV
      WRITE(6,75)(UF2(I),I=1,NX)
  344 DO 345 I=1,NX
      SA(I)=FUTF(BXG+DX*FLOAT(I-1),TV)
  345 E(I)=(UF2(I)-SA(I))/UF2(I)
      IAL=1
  348 GO TO ( 380, 370),ID
  370 GO TO (371,372,373),IAL
  371 WRITE(6,65)K,TV
      GO TO 374
  372 WRITE(6,66)K,TV
      GO TO 374
  373 WRITE(6,67)K,TV
  374 WRITE(6,75)(E(I),I=1,NX)
  380 DO 385 I=1,3
  385 EM(I,4)=0.
      DO 390 I=2,NXZ
      EM(1,4)=AMAX1(EM(1,4),ABS(E(I)))
      EM(2,4)=EM(2,4)+E(I)
  390 EM(3,4)=EM(3,4)+ABS(E(I))
      DO 395 I=2,3
  395 EM(I,4)=EM(I,4)/FNX2
  410 DO 420 J=1,3
  420 EM(J,IAL)=EM(J,4)-EM(J,IAL)
      IF(IDP.EQ.2) GO TO (430,440,450),IAL
      GO TO 460
  430 WRITE(6,95) K,TV,(EM(J,4),J=1,3),(EM(J,1),J=1,3)
      GO TO 460
  440 WRITE(6,96) K,TV,(EM(J,4),J=1,3),(EM(J,2),J=1,3)
      GO TO 460
  450 WRITE(6,97) K,TV,(EM(J,4),J=1,3),(EM(J,3),J=1,3)
  460 DO 470 J=1,3
      S(J,4,IAL)=S(J,4,IAL)+EM(J,IAL)
  470 EM(J,IAL)=EM(J,4)
      GO TO (480,490,500),IAL
  480 DO 485 I=1,NX
  485 E(I)=(UF2(I)-UV2(I))/UF2(I)
      IAL=2
      GO TO 348
  490 DO 495 I=1,NX
  495 E(I)=(UV2(I)-SA(I))/UV2(I)
      IAL=3
      GO TO 348
  500 DO 510 I=1,NX
      UV1(I)=UV2(I)
  510 UF1(I)=UF2(I)
      PT(1)=FLOAT(NT-1)
      PT(2)=DT*PT(1)
```

```
      DO 520 K=1,2
      DO 520 I=1,3
      DO 520 J=1,3
  520 S(I,K,J)=S(I,4,J)/PT(K)
      WRITE(6,98)(((S(I,K,J),I=1,3),J=1,3),K=1,2)
      STOP
   15 FORMAT(1HO//13HODONNEES LUES/10HOBXG     = ,E16.8,5X, 24H(VALEUR IN
     1FERIEURE DE X)/10H BTB     = ,E16.8,5X,24H(VALEUR INFERIEURE DE T)/
     23H DX,5X,2H= ,E16.8,5X,10H(PAS EN X)/3H DT,5X,2H= ,E16.8,5X,10H(PA
     3S EN T)/3H NX,5X,2H= ,I3,18X,36H(NOMBRE DE POINTS SELON L AXE DES
     4X)  /3H NT,5X,2H= ,I3,18X,36H(NOMBRE DE POINTS SELON L AXE DES T)/
     610H IFUB    = ,I3,18X,10H IFUG    = ,I5,16X,10H IFUD    = ,I5/10H ITE
     7ON  = ,I3,18X,61H(NOMBRE DE BITS BINAIRES PERDUS A CHAQUE OPERATIO
     8N FLOTTANTE)/3H IW,5X,1H=,I4,18X,46H(SI IW INEGAL A 0  IMPRESSIONS
     9 INTERMEDIAIRES)/1HO)
   17 FORMAT(1HO/22HOCOEFFICIENT  CALCULE /10HOCOEF    = ,E16.8)
   20 FORMAT(5E12.8,2I3,3I1,I2,I1)
   27 FORMAT(18H1VALEURS INITIALES/1HO)
   30 FORMAT(6E12.8)
   33 FORMAT(     36HORESULTATS PRECISION MAXIMUM NIVEAU ,I3,2X,4H(T =,E1
     16.8,1H))
   34 FORMAT(1HO/36HORESULTATS                          NIVEAU ,I3,2X,4H(T =,E1
     16.8,1H)/1HO)
   35 FORMAT(30HORESULTATS AVEC TRONCATION DE ,I2,22H BITS BINAIRES NIVE
     1AU ,I3,2X,4H(T =,E16.8,1H))
   63 FORMAT(6H1EX6GL,10X,82HEQUATION DE LA CHALEUR METHODE EXPLICITE ET
     1UDE ERREUR EN FLOTTANT (6 COMPARAISONS))
   65 FORMAT(42HODETAIL DIFF. REL. AVEC SOL. ANALYT. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   66 FORMAT(42HODETAIL DIFF. REL. AVEC PRECIS. MAX. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   67 FORMAT(42HODETAIL DIFF. REL. PR. MX./ SOL. AN. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   75 FORMAT(6E16.8)
   95 FORMAT(29HOD. R. S. TR. / SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   96 FORMAT(29HOD. R. S. TR. / PR. MX.  NIV.,I4,5H (T =,E16.8, 1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   97 FORMAT(29HOD. R. PR. MX./ SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   98 FORMAT(1HO////30HOPENTE MOYENNE DES DIFFERENCES///29HOEN FONCTION
     1DU NOMBRE DE PAS//24HOD. R. S. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,
     25HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. R. S. TR. / PR. MX. ,
     333X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. R. P
     4R. MX./ SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     5E12.4///42HOEN FONCTION DE L INTERVALLE PARCOURU EN T//24HOD. R. S
     6. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     7E12.4/24HOD. R. S. TR. / PR. MX. ,33X,4HMX =,E12.4,2X,5HMOY =,
     8E12.4,2X,9HMOY MOD =,E12.4/24HOD. R. PR. MX./ SOL. AN.,33X,4HMX =,
     9E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/1H1)
      END
```

```
C       EQUA CHALEUR METH. EXPLIC. 1 VAR D ESP ETUDE ERREUR (6 COMP.)
        DIMENSION UP(401),UG(201),UV1(401),UV2(401),UF1(401),UF2(401),
       1E(401),UD(201),UB(401),SA(401),EM(3,4),S(3,4,3),PT(2)
        EXTERNAL FUTF
  240   READ(5,20)BXG,BTB,DX,DT,SIGMA,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
  248   NXZ=NX-1
        NTMOI = (NT-1)/2+1
        FNX2=FLOAT(NX-2)
        COEF=DT/DX**2
        WRITE(6,63)
        WRITE(6,15)BXG,BTB,DX,DT,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
        WRITE(6,17)COEF
  250   IF(IFUB)253,253,256
  253   READ(5,30)(UB(I),I=1,NX)
        GO TO 262
  256   DO 259 I=1,NX
  259   UB(I)=FUTF(BXG+DX*FLOAT(I-1),BTB)
  262   IF(IFUG)265,265,268
  265   READ(5,30)(UG(I),I=1,NT)
        GO TO 273
  268   DO 271 I=1,NT
  271   UG(I)=FUTF(BXG,BTB+DT*FLOAT(I-1))
  273   IF(IFUD)276,276,279
  276   READ(5,30)(UD(I),I=1,NT)
        GO TO 320
  279   DO 282 I=1,NT
  282   UD(I)=FUTF(BXG+DX*FLOAT(NX-1),BTB+DT*FLOAT(I-1))
  320   DO 325 I=2,NXZ
  321   UV1(I)=UB(I)
  325   UF1(I)=UB(I)
        UV1(1)=UG(1)
        UF1(1)=UG(1)
        UV1(NX)=UD(1)
        UF1(NX)=UD(1)
        IF(IW.EQ.0) GO TO 327
        WRITE(6,27)
        WRITE(6,75)(UV1(I),I=1,NX)
  327   DO 510 K=2,NT
        ID=1
        IDP=0
        IDL=0
        IF((K.EQ.2).OR.(K.EQ.NTMOI).OR.(K.EQ.NT)) IDL=2
        IF((IDL.EQ.2).AND.(IW.NE.0)) ID=2
        IF((IDL.EQ.2).OR.(IW.NE.0)) IDP=2
        TV=BTB+DT*FLOAT(K-1)
        DO 330 I=2,NXZ
  330   UV2(I)=UV1(I)+COEF*(UV1(I-1)-2.*UV1(I)+UV1(I+1))
        UV2(1)=UG(K)
        UV2(NX)=UD(K)
        IF(IDP.EQ.2) WRITE(6,34) K,TV
        GO TO (9338, 335),ID
  335   WRITE(6,33)K,TV
        WRITE(6,75)(UV2(I),I=1,NX)
 9338   CALL TROFXS(ITRON)
```

```
      DO 340 I=2,NXZ
  340 UF2(I)=UF1(I)+COEF*(UF1(I-1)-2.*UF1(I)+UF1(I+1))
 9342 CALL TROFXS(ITRON)
      UF2(1)=UG(K)
      UF2(NX)=UD(K)
      GO TO ( 344, 342),ID
  342 WRITE(6,35)ITRON,K,TV
      WRITE(6,75)(UF2(I),I=1,NX)
  344 DO 345 I=1,NX
      SA(I)=FUTF(BXG+DX*FLOAT(I-1),TV)
  345 E(I)=(UF2(I)-SA(I))
      IAL=1
  348 GO TO ( 380, 370),ID
  370 GO TO (371,372,373),IAL
  371 WRITE(6,65)K,TV
      GO TO 374
  372 WRITE(6,66)K,TV
      GO TO 374
  373 WRITE(6,67)K,TV
  374 WRITE(6,75)(E(I),I=1,NX)
  380 DO 385 I=1,3
  385 EM(I,4)=0.
      DO 390 I=2,NXZ
      EM(1,4)=AMAX1(EM(1,4),ABS(E(I)))
      EM(2,4)=EM(2,4)+E(I)
  390 EM(3,4)=EM(3,4)+ABS(E(I))
      DO 395 I=2,3
  395 EM(I,4)=EM(I,4)/FNX2
  410 DO 420 J=1,3
  420 EM(J,IAL)=EM(J,4)-EM(J,IAL)
      IF(IDP.EQ.2) GO TO (430,440,450),IAL
      GO TO 460
  430 WRITE(6,95) K,TV,(EM(J,4),J=1,3),(EM(J,1),J=1,3)
      GO TO 460
  440 WRITE(6,96) K,TV,(EM(J,4),J=1,3),(EM(J,2),J=1,3)
      GO TO 460
  450 WRITE(6,97) K,TV,(EM(J,4),J=1,3),(EM(J,3),J=1,3)
  460 DO 470 J=1,3
      S(J,4,IAL)=S(J,4,IAL)+EM(J,IAL)
  470 EM(J,IAL)=EM(J,4)
      GO TO (480,490,500),IAL
  480 DO 485 I=1,NX
  485 E(I)=(UF2(I)-UV2(I))
      IAL=2
      GO TO 348
  490 DO 495 I=1,NX
  495 E(I)=(UV2(I)-SA(I))
      IAL=3
      GO TO 348
  500 DO 510 I=1,NX
      UV1(I)=UV2(I)
  510 UF1(I)=UF2(I)
      PT(1)=FLOAT(NT-1)
      PT(2)=DT*PT(1)
```

```
      DO 520 K=1,2
      DO 520 I=1,3
      DO 520 J=1,3
  520 S(I,K,J)=S(I,4,J)/PT(K)
      WRITE(6,98)(((S(I,K,J),I=1,3),J=1,3),K=1,2)
      STOP
C
C
   15 FORMAT(1HO//13HODONNEES LUES/10HOBXG     = ,E16.8,5X, 24H(VALEUR IN
     1FERIEURE DE X)/10H BTB     = ,E16.8,5X,24H(VALEUR INFERIEURE DE T)/
     23H DX,5X,2H= ,E16.8,5X,10H(PAS EN X)/3H DT,5X,2H= ,E16.8,5X,10H(PA
     3S EN T)/3H NX,5X,2H= ,I3,18X,36H(NOMBRE DE POINTS SELON L AXE DES
     4X)  /3H NT,5X,2H= ,I3,18X,36H(NOMBRE DE POINTS SELON L AXE DES T)/
     610H IFUB    = ,I3,18X,10H IFUG    = ,I5,16X,10H IFUD    = ,I5/10H ITR
     70N   = ,I3,18X,81H(NOMBRE DE BITS BINAIRES CONSERVES APRES LA VIRGU
     8LE A CHAQUE OPERATION FLOTTANTE)/3H IW,5X,1H=,I4,18X,46H(SI IW INE
     9GAL A 0  IMPRESSIONS INTERMEDIAIRES)/1HO)
   17 FORMAT(1HO/22HOCOEFFICIENT  CALCULE /10HOCOEF    = ,E16.8)
   20 FORMAT(5E12.8,2I3,3I1,I2,I1)
   27 FORMAT(18H1VALEURS INITIALES/1HO)
   30 FORMAT(6E12.8)
   33 FORMAT(     36HORESULTATS PRECISION MAXIMUM NIVEAU ,I3,2X,4H(T =,E1
     16.8,1H))
   34 FORMAT(1HO/36HORESULTATS                          NIVEAU ,I3,2X,4H(T =,E1
     16.8,1H)/1HO)
   35 FORMAT(15HORESULTATS AVEC,I3,39H BITS BINAIRES APRES LA VIRGULE  N
     1IVEAU,I4,2X,4H(T =,E16.8,1H))
   63 FORMAT(6H1EX6GX,10X,82HEQUATION DE LA CHALEUR METHODE EXPLICITE ET
     1UDE ERREUR EN FIXE      (6 COMPARAISONS))
   65 FORMAT(42HODETAIL DIFF. ABS. AVEC SOL. ANALYT. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   66 FORMAT(42HODETAIL DIFF. ABS. AVEC PRECIS. MAX. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   67 FORMAT(42HODETAIL DIFF. ABS. PR. MX./ SOL. AN. NIV. ,I3,2X,4H(T =,
     1E16.8,1H))
   75 FORMAT(6E16.8)
   95 FORMAT(29HOD. A. S. TR. / SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   96 FORMAT(29HOD. A. S. TR. / PR. MX.  NIV.,I4,5H (T =,E16.8, 1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   97 FORMAT(29HOD. A. PR. MX./ SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   98 FORMAT(1HO////30HOPENTE MOYENNE DES DIFFERENCES///29HOEN FONCTION
     1DU NOMBRE DE PAS//24HOD. A. S. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,
     25HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. S. TR. / PR. MX. ,
     333X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. P
     4R. MX./ SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     5E12.4///42HOEN FONCTION DE L INTERVALLE PARCOURU EN T//24HOD. A. S
     6. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     7E12.4/24HOD. A. S. TR. / PR. MX. ,33X,4HMX =,E12.4,2X,5HMOY =,
     8E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. PR. MX./ SOL. AN.,33X,4HMX =,
```

```
9E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/1H1)
 END
```

```
C       EQUA CHALEUR METH CRANK ET NICOLSON ETUDE ERREUR PAR 6 COMP.
        DIMENSION UP(401),UG(201),UV1(401),UV2(401),UF1(401),UF2(401),
       1E(401),UD(201),UB(401),SA(401),EM(3,4),S(3,4,3),US(401),PT(2)
        EXTERNAL FUTF
  240 READ(5,20)BXG,BTB,DX,DT,SIGMA,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
        NXZ=NX-1
        FNXZ=FLOAT(NXZ)
        NX2=NX-2
        FNX2=FLOAT(NX2)
        NTMOI=(NT-1)/2+1
        AMBDA=DT/DX**2
        CA0=-(2.+1./(AMBDA*SIGMA))
        CA1=-(2.-1./(AMBDA*(1.-SIGMA)))
        CA2=(SIGMA-1.)/SIGMA
  245 WRITE (6,63)
        WRITE (6,15)BXG,BTB,DX,DT,NX,NT,SIGMA,IFUB,IFUG,IFUD,ITRON,IW
        WRITE (6,17)AMBDA,CA0,CA1,CA2
  310 IF(IFUB)320,320,330
  320 READ(5,30)(UB(I),I=1,NX)
        GO TO 350
  330 DO 340 I=1,NX
  340 UB(I)=FUTF(BXG+DX*FLOAT(I-1),BTB)
  350 IF(IFUG)360,360,370
  360 READ(5,30)(UG(I),I=1,NT)
        GO TO 390
  370 DO 380 I=1,NT
  380 UG(I)=FUTF(BXG,BTB+DT*FLOAT(I-1))
  390 IF(IFUD)400,400,410
  400 READ(5,30)(UD(I),I=1,NT)
        GO TO 510
  410 DO 420 I=1,NT
  420 UD(I)=FUTF(BXG+DX*FNXZ,BTB+DT*FLOAT(I-1))
  510 DO 530 I=2,NXZ
        UV1(I)=UB(I)
  530 UF1(I)=UB(I)
  540 UV1(1)=UG(1)
        UV1(NX)=UD(1)
  550 UF1(1)=UV1(1)
        UF1(NX)=UV1(NX)
        IF(IW.EQ.0) GO TO 610
  560 WRITE(6,27)
  570 WRITE(6,75) (UV1(I),I=1,NX)
  610 DO 910 K=2,NT
        ID=1
        IDP=0
        IDL=0
        IF((K.EQ.2).OR.(K.EQ.NTMOI).OR.(K.EQ.NT)) IDL=2
        IF((IDL.EQ.2).AND.(IW.NE.0)) ID=2
        IF((IDL.EQ.2).OR.(IW.NE.0)) IDP=2
  615 TV=BTB+DT*FLOAT(K-1)
  620 DO 625 I=3,NX2
  625 US(I)=CA2*(UV1(I-1)+CA1*UV1(I)+UV1(I+1))
        US(2)=CA2*(CA1*UV1(2)+UV1(3))-UG(K)+CA2*UG(K-1)
        US(NX-1)=CA2*(CA1*UV1(NX-1)+UV1(NX-2))-UD(K)+CA2*UD(K-1)
```

```
 630 CALL TRISPE (CA0,NX,UV2,US,0)
 640 UV2(1)=UG(K)
     UV2 (NX)=UD(K)
8650 IF(IDP.EQ.2) WRITE(6,33) K,TV
     GO TO(9665,660),ID
 660 WRITE(6,34)K,TV
     WRITE(6,75)(UV2(I),I=1,NX)
9665 CALL TRONC (ITRON)
 670 DO 675 I=3,NX2
 675 US(I)=CA2*(UF1(I-1)+CA1*UF1(I)+UF1(I+1))
     US(2)=CA2*(CA1*UF1(2)+UF1(3))-UG(K)+CA2*UG(K-1)
     US(NX-1)=CA2*(CA1*UV1(NX-1)+UV1(NX-2))-UD(K)+CA2*UD(K-1)
9678 CALL TRONC (ITRON)
 680 CALL TRISPE (CA0,NX,UF2,US,ITRON)
 690 UF2(1)=UG(K)
     UF2(NX)=UD(K)
     GO TO (744,700),ID
 700 WRITE(6,43)ITRON,K,TV
     WRITE(6,75)(UF2(I),I=1,NX)
 744 DO 745 I=1,NX
     SA(I)=FUTF(BXG+DX*FLOAT(I-1),TV)
 745 E(I)=(UF2(I)-SA(I))/UF2(I)
     IAL=1
 748 GO TO ( 780, 770),ID
 770 GO TO (771,772,773),IAL
 771 WRITE(6,65)K,TV
     GO TO 774
 772 WRITE(6,66)K,TV
     GO TO 774
 773 WRITE(6,67)K,TV
 774 WRITE(6,75)(E(I),I=1,NX)
 780 DO 785 I=1,3
 785 EM(I,4)=0.
     DO 790 I=2,NXZ
     EM(1,4)=AMAX1(EM(1,4),ABS(E(I)))
     EM(2,4)=EM(2,4)+E(I)
 790 EM(3,4)=EM(3,4)+ABS(E(I))
     DO 795 I=2,3
 795 EM(I,4)=EM(I,4)/FNX2
 810 DO 820 J=1,3
 820 EM(J,IAL)=EM(J,4)-EM(J,IAL)
     IF(IDP.EQ.2) GO TO (830,840,850),IAL
     GO TO 860
 830 WRITE(6,95) K,TV,(EM(J,4),J=1,3),(EM(J,1),J=1,3)
     GO TO 860
 840 WRITE(6,96) K,TV,(EM(J,4),J=1,3),(EM(J,2),J=1,3)
     GO TO 860
 850 WRITE(6,97) K,TV,(EM(J,4),J=1,3),(EM(J,3),J=1,3)
 860 DO 870 J=1,3
     S(J,4,IAL)=S(J,4,IAL)+EM(J,IAL)
 870 EM(J,IAL)=EM(J,4)
     GO TO (880,890,900),IAL
 880 DO 885 I=1,NX
 885 E(I)=(UF2(I)-UV2(I))/UF2(I)
```

```
      IAL=2
      GO TO 748
  890 DO 895 I=1,NX
  895 E(I)=(UV2(I)-SA(I))/UV2(I)
      IAL=3
      GO TO 748
  900 DO 910 I=1,NX
      UV1(I)=UV2(I)
  910 UF1(I)=UF2(I)
      PT(1)=FLOAT(NT-1)
      PT(2)=DT*PT(1)
      DO 920 K=1,2
      DO 920 I=1,3
      DO 920 J=1,3
  920 S(I,K,J)=S(I,4,J)/PT(K)
      WRITE(6,98)(((S(I,K,J),I=1,3),J=1,3),K=1,2)
      STOP
C
C
   15 FORMAT(1HO//13HODONNEES LUES/10HOBXG     = ,E16.8,5X, 24H(VALEUR IN
     1FERIEURE DE X)/10H BTB     = ,E16.8,5X,24H(VALEUR INFERIEURE DE T)/
     210H DX      = ,E16.8,5X,10H(PAS EN X)/10H DT      = ,E16.8,5X,10H(PA
     3S EN T)/10H NX      = ,I3,18X,36H(NOMBRE DE POINTS SELON L'AXE DES
     4X)  /10H NT      = ,I3,18X,36H(NOMBRE DE POINTS SELON L'AXE DES T)/
     510H SIGMA   = ,E16.8,5X,31H(CARACTERISTIQUE DE LA METHODE)/
     610H IFUB    = ,I3,17X,10H IFUG    = ,I3,16X,10H IFUD    = ,I3/10H ITR
     70N   = ,I3,18X,61H(NOMBRE DE BITS BINAIRES PERDUS A CHAQUE OPERATIO
     8N FLOTTANTE)/3H IW,5X,1H=,I4,18X,46H(SI IW INEGAL A 0  IMPRESSIONS
     9 INTERMEDIAIRES)/1HO)
   17 FORMAT(1HO/22HOCOEFFICIENTS CALCULES/10HOAMBDA  = ,E16.8/10H CA0
     1  =  ,E16.8/10H CA1     = ,E16.8/10H CA2     = ,E16.8/1HO)
   20 FORMAT(5E12.8,2I3,3I1,I2,I1)
   27 FORMAT(18H1VALEURS INITIALES/1HO)
   30 FORMAT (6E12.8)
   33 FORMAT (1HO/35HORESULTATS                              NIVEAU,I4,5X,5H( T =,
     1E16.8,2H )/1HO)
   34 FORMAT(35HORESULTATS PRECISION MAXIMUM NIVEAU,I4,5X,4H(T =,E16.8,
     11H))
   43 FORMAT(     30HORESULTATS AVEC TRONCATION DE ,I2,23H BITS BINAIRES,
     1 NIVEAU ,I3,5X,5H( T =,E16.8,2H ))
   63 FORMAT(6H1CN6GL,10X,93HEQUATION DE LA CHALEUR METHODE DE CRANK ET
     1NICOLSON ETUDE ERREUR EN FLOTTANT (6 COMPARAISONS))
   65 FORMAT(42HODETAIL DIFF. REL. AVEC SOL. ANALYT. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   66 FORMAT(41HODETAIL DIFF.REL. AVEC PRECIS. MAX. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   67 FORMAT(42HODETAIL DIFF. REL. PR. MX./ SOL. AN. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   75 FORMAT(6E16.8)
   95 FORMAT(29HOD. R. S. TR. / SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   96 FORMAT(29HOD. R. S. TR. / PR. MX.  NIV.,I4,5H (T =,E16.8, 1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
```

```
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
  97 FORMAT(29HOD. R. PR. MX./ SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
  98 FORMAT(1HO////30HOPENTE MOYENNE DES DIFFERENCES///29HOEN FONCTION
     1DU NOMBRE DE PAS//24HOD. R. S. TR. / SOL. AN.,33X,4HMX =,E12.4,2X
     25HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. R. S. TR. / PR. MX. ,
     333X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. R.
     4R. MX./ SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =
     5E12.4///42HOEN FONCTION DE L INTERVALLE PARCOURU EN T//24HOD. R.
     6. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =
     7E12.4/24HOD. R. S. TR. / PR. MX. ,33X,4HMX =,E12.4,2X,5HMOY =,
     8E12.4,2X,9HMOY MOD =,E12.4/24HOD. R. PR. MX./ SOL. AN.,33X,4HMX =,
     9E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/1H1)
      END
```

```
C       EQUA CHALEUR METH CRANK ET NICOLSON ETUDE ERREUR PAR 6 COMP. (FIXE)
        DIMENSION US(401),UG(201),UV1(401),UV2(401),UF1(401),UF2(401),
       1E(401),UD(201),UB(401),SA(401),EM(3,4),S(3,4,3),PT(2)
        EXTERNAL FUTF
  240 READ(5,20)BXG,BTB,DX,DT,SIGMA,NX,NT,IFUB,IFUG,IFUD,ITRON,IW
        NXZ=NX-1
        FNXZ=FLOAT(NXZ)
        NX2=NX-2
        FNX2=FLOAT(NX2)
        NTMOI=(NT-1)/2+1
        AMBDA=DT/DX**2
        CA0=-(2.+1./(AMBDA*SIGMA))
        CA1=-(2.-1./(AMBDA*(1.-SIGMA)))
        CA2=(SIGMA-1.)/SIGMA
  245 WRITE (6,63)
        WRITE (6,15)BXG,BTB,DX,DT,NX,NT,SIGMA,IFUB,IFUG,IFUD,ITRON,IW
        WRITE (6,17)AMBDA,CA0,CA1,CA2
  310 IF(IFUB)320,320,330
  320 READ(5,30)(UB(I),I=1,NX)
        GO TO 350
  330 DO 340 I=1,NX
  340 UB(I)=FUTF(BXG+DX*FLOAT(I-1),BTB)
  350 IF(IFUG)360,360,370
  360 READ(5,30)(UG(I),I=1,NT)
        GO TO 390
  370 DO 380 I=1,NT
  380 UG(I)=FUTF(BXG,BTB+DT*FLOAT(I-1))
  390 IF(IFUD)400,400,410
  400 READ(5,30)(UD(I),I=1,NT)
        GO TO 510
  410 DO 420 I=1,NT
  420 UD(I)=FUTF(BXG+DX*FNXZ       ,BTB+DT*FLOAT(I-1))
  510 DO 530 I=2,NXZ
        UV1(I)=UB(I)
  530 UF1(I)=UB(I)
  540 UV1(1)=UG(1)
        UV1(NX)=UD(1)
  550 UF1(1)=UV1(1)
        UF1(NX)=UV1(NX)
        IF(IW.EQ.0) GO TO 610
  560 WRITE(6,27)
  570 WRITE(6,75) (UV1(I),I=1,NX)
  610 DO 910 K=2,NT
        IDP=0
        IDL=0
        ID=1
        IF((K.EQ.2).OR.(K.EQ.NTMOI).OR.(K.EQ.NT)) IDL=2
        IF((IDL.EQ.2).AND.(IW.NE.0)) ID=2
        IF((IDL.EQ.2).OR.(IW.NE.0)) IDP=2
  615 TV=BTB+DT*FLOAT(K-1)
  620 DO 625 I=3,NX2
  625 US(I)=CA2*(UV1(I-1)+CA1*UV1(I)+UV1(I+1))
        US(2)=CA2*(CA1*UV1(2)+UV1(3))-UG(K)+CA2*UG(K-1)
        US(NX-1)=CA2*(CA1*UV1(NX-1)+UV1(NX-2))-UD(K)+CA2*UD(K-1)
```

```
 630 CALL TRISPX (CA0,NX,UV2,US,155)
 640 UV2(1)=UG(K)
     UV2 (NX)=UD(K)
8650 IF(IDP.EQ.2) WRITE(6,33) K,TV
     GO TO(9665,660),ID
 660 WRITE(6,34)K,TV
     WRITE(6,75)(UV2(I),I=1,NX)
9665 CALL TROFXS(ITRON)
 670 DO 675 I=3,NX2
 675 US(I)=CA2*(UF1(I-1)+CA1*UF1(I)+UF1(I+1))
     US(2)=CA2*(CA1*UF1(2)+UF1(3))-UG(K)+CA2*UG(K-1)
     US(NX-1)=CA2*(CA1*UV1(NX-1)+UV1(NX-2))-UD(K)+CA2*UD(K-1)
9678 CALL TROFXS(ITRON)
 680 CALL TRISPX (CA0,NX,UF2,US,ITRON)
 690 UF2(1)=UG(K)
     UF2(NX)=UD(K)
     GO TO (744,700),ID
 700 WRITE(6,43)ITRON,K,TV
     WRITE(6,75)(UF2(I),I=1,NX)
 744 DO 745 I=1,NX
     SA(I)=FUTF(BXG+DX*FLOAT(I-1),TV)
 745 E(I)=(UF2(I)-SA(I))
     IAL=1
 748 GO TO ( 780, 770),ID
 770 GO TO (771,772,773),IAL
 771 WRITE(6,65)K,TV
     GO TO 774
 772 WRITE(6,66)K,TV
     GO TO 774
 773 WRITE(6,67)K,TV
 774 WRITE(6,75)(E(I),I=1,NX)
 780 DO 785 I=1,3
 785 EM(I,4)=0.
     DO 790 I=2,NXZ
     EM(1,4)=AMAX1(EM(1,4),ABS(E(I)))
     EM(2,4)=EM(2,4)+E(I)
 790 EM(3,4)=EM(3,4)+ABS(E(I))
     DO 795 I=2,3
 795 EM(I,4)=EM(I,4)/FNX2
 810 DO 820 J=1,3
 820 EM(J,IAL)=EM(J,4)-EM(J,IAL)
     IF(IDP.EQ.2) GO TO (830,840,850),IAL
     GO TO 860
 830 WRITE(6,95)K,TV,(EM(J,4),J=1,3),(EM(J,1),J=1,3)
     GO TO 860
 840 WRITE(6,96) K,TV,(EM(J,4),J=1,3),(EM(J,2),J=1,3)
     GO TO 860
 850 WRITE(6,97) K,TV,(EM(J,4),J=1,3),(EM(J,3),J=1,3)
 860 DO 870 J=1,3
     S(J,4,IAL)=S(J,4,IAL)+EM(J,IAL)
 870 EM(J,IAL)=EM(J,4)
     GO TO (880,890,900),IAL
 880 DO 885 I=1,NX
 885 E(I)=(UF2(I)-UV2(I))
```

```
      IAL=2
      GO TO 748
  890 DO 895 I=1,NX
  895 E(I)=(UV2(I)-SA(I))
      IAL=3
      GO TO 748
  900 DO 910 I=1,NX
      UV1(I)=UV2(I)
  910 UF1(I)=UF2(I)
      PT(1)=FLOAT(NT-1)
      PT(2)=DT*PT(1)
      DO 920 K=1,2
      DO 920 I=1,3
      DO 920 J=1,3
  920 S(I,K,J)=S(I,4,J)/PT(K)
      WRITE(6,98)(((S(I,K,J),I=1,3),J=1,3),K=1,2)
      STOP
C
C
   15 FORMAT(1HO//13HODONNEES LUES/10HOBXG     = ,E16.8,5X, 24H(VALEUR IN
     1FERIEURE DE X)/10H BTB     = ,E16.8,5X,24H(VALEUR INFERIEURE DE T)/
     210H DX      = ,E16.8,5X,10H(PAS EN X)/10H DT      = ,E16.8,5X,10H(PA
     3S EN T)/10H NX      = ,I3,18X,36H(NOMBRE DE POINTS SELON L'AXE DES
     4X)  /10H NT      = ,I3,18X,36H(NOMBRE DE POINTS SELON L'AXE DES T)/
     510H SIGMA   = ,E16.8,5X,31H(CARACTERISTIQUE DE LA METHODE)/
     610H IFUB    = ,I3,17X,10H IFUG    = ,I3,16X,10H IFUD    = ,I3/10H ITR
     70N   = ,I3,18X,81H(NOMBRE DE BITS BINAIRES CONSERVES APRES LA VIRGU
     8LE A CHAQUE OPERATION FLOTTANTE)/3H IW,5X,1H=,I4,18X,46H(SI IW INE
     9GAL A 0  IMPRESSIONS INTERMEDIAIRES)/1HO)
   17 FORMAT(1HO/22HOCOEFFICIENTS CALCULES/10HOAMBDA  = ,E16.8/10H CA0
     1   =  ,E16.8/10H CA1     = ,E16.8/10H CA2     = ,E16.8/1HO)
   20 FORMAT(5E12.8,2I3,3I1,I2,I1)
   27 FORMAT(18H1VALEURS INITIALES/1HO)
   30 FORMAT (6E12.8)
   33 FORMAT (1HO/35HORESULTATS                          NIVEAU,I4,5X,5H( T =,
     1E16.8,2H )/1HO)
   34 FORMAT(35HORESULTATS PRECISION MAXIMUM NIVEAU,I4,5X,4H(T =,E16.8,
     11H))
   43 FORMAT(15HORESULTATS AVEC,I3,43H CHIFFRES BINAIRES APRES LA VIRGUL
     1E, NIVEAU,I4,5X,4H(T =,E16.8,2H) )
   63 FORMAT(6H1CN6GX,10X,93HEQUATION DE LA CHALEUR METHODE DE CRANK ET
     1NICOLSON ETUDE ERREUR EN FIXE      (6 COMPARAISONS))
   65 FORMAT(42HODETAIL DIFF. ABS. AVEC SOL. ANALYT. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   66 FORMAT(41HODETAIL DIFF.ABS. AVEC PRECIS. MAX. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   67 FORMAT(42HODETAIL DIFF. ABS. PR. MX./ SOL. AN. NIV. ,I3,5X,5H( T =
     1 ,E16.8,2H ))
   75 FORMAT(6E16.8)
   95 FORMAT(29HOD. A. S. TR. / SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
   96 FORMAT(29HOD. A. S. TR. / PR. MX.  NIV.,I4,5H (T =,E16.8, 1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
```

```
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
  97 FORMAT(29HOD. A. PR. MX./ SOL. AN. NIV.,I4,5H (T =,E16.8,1H),2X,
     14HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12. 4/ 34X,21HDIFF.
     2AVEC NIV. PREC.,6X,E12.4,7X,E12.4,11X,E12.4)
  98 FORMAT(1HO////30HOPENTE MOYENNE DES DIFFERENCES///29HOEN FONCTION
     1DU NOMBRE DE PAS//24HOD. A. S. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,
     25HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. S. TR. / PR. MX. ,
     333X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. I
     4R. MX./ SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     5E12.4///42HOEN FONCTION DE L INTERVALLE PARCOURU EN T//24HOD. A. S
     6. TR. / SOL. AN.,33X,4HMX =,E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,
     7E12.4/24HOD. A. S. TR. / PR. MX. ,33X,4HMX =,E12.4,2X,5HMOY =,
     8E12.4,2X,9HMOY MOD =,E12.4/24HOD. A. PR. MX./ SOL. AN.,33X,4HMX =,
     9E12.4,2X,5HMOY =,E12.4,2X,9HMOY MOD =,E12.4/1H1)
      END
```

```
      SUB ROUTINE TRISPE(C,N,X,B,ITRON)
C     RESOUD SYST. LIN. (METH. ELIMINATION AVEC DIVISIONS)
      DIMENSION X(401),D(401),B(401)
  220 CALL TRONC(ITRON)
  260 N1=N-1
  270 D(2)=C
  280 DO 300 I=3,N1
  290 B(I)=B(I)-B(I-1)/D(I-1)
  300 D(I)=C-1./D(I-1)
  310 X(N1)=B(N1)/D(N1)
  320 DO 330 I=3,N1
      L=N-I+1
  330 X(L)=(B(L)-X(L+1))/D(L)
  340 CALL TRONC(ITRON)
  360 RETURN
      END
```

```
      SUB ROUTINE TRIVEF(C,N,X,B,ITRON)
C     RESOUD SYST. LIN. (METHODE ELIMINATION SANS DIVISION)
      DIMENSION X(101),B(101),D(101)
  220 CALL TRONC(ITRON)
  260 N1=N-1
  270 D(1)=1.
      D(2)=C
  280 DO 300 I=3,N1
  290 D(I)=C*D(I-1)-D(I-2)
  300 B(I)=B(I)*D(I-1)-B(I-1)
  310 X(N1)=B(N1)/D(N1)
  320 DO 330 I=3,N1
      L=N-I+1
  330 X(L)=(B(L)-D(L-1)*X(L+1))/D(L)
  340 CALL TRONC(ITRON)
  360 RETURN
      END
```

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

1 L. BOLLIET - N. GASTINEL - P.J. LAURENT - Algol, manuel pratique - Edition Hermann - Paris 1964 -

2 J. DOUGLAS Jr - A survey of numerical methods for parabolic differential equations - Rice University - Houston (E.U.A.) 1960 -

3 J.C. DUMAS - Erreur de discrétisation dans l'étude des systèmes différentiels linéaires - Application à l'équation de la chaleur - Thèse de 3e cycle - Université de Nancy - 1963 -

4 E. DURAND - Solutions numériques des équations algébriques Tome 2 : systèmes de plusieurs équations - Editions Masson - Paris 1961 -

5 G.E. FORSYTHE - W.R. WASOW - Finite difference methods for partial differential équations - John Wiley and sons - New York - 1960 -

6 N. GASTINEL - Matrices du second degré et normes générales en analyse numérique linéaire - Thèse de Doctorat ès Sciences Université de Grenoble - 1960 -

7 N. GASTINEL - Analyse numérique linéaire - 2e partie : méthodes directes de résolution de systèmes linéaires - Cours de l'Université de Grenoble - 1964 -

8 R.D. RICHTMYER - Difference methods for initial value problems Interscience Publischers - New York - 1957 -

9 RYSHIK, GRADSTEIN - Tables - Deutscher Verlag der Wissenschaften - Berlin - 1957 -

10 R.S. VARGA - Matrix iterative analysis - Prentice Hall - Englewood Cliffs (EUA) 1962.

- T A B L E D E S M A T I E R E S -

VU,

Grenoble, le

Le Président de la Thèse

VU,

Grenoble, le

Le Doyen de la Faculté des Sciences

VU et permis d'imprimer,

Le Recteur de l'Académie de Grenoble