50376
1965
89

50376
1965
89

U N I V E R S I T E D E L I L L E

Faculté des Sciences --- Laboratoire de Calcul

DIPLOME D'ETUDES SUPERIEURES

Traducteur A.P.B. sur 1620

JURY MM BACCHUS
POITOU
POUZET
HERZ

Présenté à LILLE en Mars 1965

par Bernard SUCHER

BIBLIOGRAPHIE

1°) - P. BACCHUS , Notice d'emploi APB

Publication du Laboratoire de Calcul de la Faculté des Sciences de l'Université de LILLE , Juin 1962.

2°) - P. BACCHUS et P. POUZET , Autoprogrammation pour calculateur BULL (Gamma ET) APB

Revue Française de Traitement de l'information , 1964 - 1.

3°) - P. BACCHUS , Notice description APB

Publications du Laboratoire de Calcul de la Faculté des Sciences de l'Université de LILLE , Novembre 62.

CHAPITRE I GENERALITES

INTRODUCTION

La publication en 1958 du rapport préliminaire sur le langage algorithmique ALGOL , a suscité beaucoup de travaux et de recherches tant sur la conception que sur la compilation des langeges de programmation. Il existe maintenant sur presque toutes les machines puissantes ou moyennes des compilateurs ALGOL dont certains sont assez performants. L'influence de ces recherches s'est fait sentir même sur le plan logique des machines les plus récentes.

L'avenir de la programmation automatique et l'extension de l'emploi des calculatrices électroniques paraît lié , au moins dans le domaine du calcul scientifique , à des langages universels dont ALGOL est un exemple.

Mais l'utilisation de ces langages de programmation universels ne présente un intérêt pratique important que pour des calculatrices déjà puissantes , dotées de mémoires à grande capacité et d'une grande vitesse de calcul. Pour des machines plus modestes et d'une conception plus ancienne , des langages moins ambitieux , tenant compte des possibilités et des caractéristiques techniques des machines , peuvent rendre des services appréciables.

Le langage APB , qui a été conçu pour le calculateur Gamma-Tambour , rentre exactement dans ce cadre.

.../...

RAPPELS SUR LE LANGAGE APB

L'APB est un langage symbolique à trois opérandes qui comprend des instructions d'entrée et sortie , des instructions d'affectation , des instructions logiques.

Suivant ce qu'ils représentent , les opérandes peuvent revêtir quatre formes :

- Opérande symbolique : c'est un identificateur de variable simple.
- Opérande indicé : c'est un identificateur de variable indicée une fois (tableau à une dimension).
- Opérande numérique : c'est une constante numérique.
- Opérande blanc : la place correspondant à l'opérande dans l'instruction est laissée vide.

Un programme APB se compose d'une suite finie d'instructions symboliques , de structure fixe , faisant chacune l'objet d'une carte perforée (carte programme). Une instruction APB comprend :

- Un numéro de niveau qui joue le rôle d'une étiquette ALGOL ;
- Un TOS (type d'opération symbolique) qui indique la nature de l'opération exécutée dans l'instruction considérée ;
- Un numéro de renvoi qui n'est significatif que pour certains TOS pouvant exécuter une rupture de séquence ;
- Trois opérandes ou , selon le TOS , seulement deux , un , zéro opérandes significatifs. Les autres sont laissés en blanc.

Le traitement des variables indicées se fait par une méthode de progression automatique des indices.

.../...

CARACTERISTIQUES TECHNIQUES DE L'ORDINATEUR I620 UTILISE

Ce texte décrit un programme de traduction APB sur l'ordinateur 1620 , qui permet d'utiliser le langage APB sur cette machine dans des limites qui seront précisées par la suite.

Le traducteur et les programmes résultants ont été étudiés sur un ordinateur 1620 équipé :

- D'une mémoire rapide à tores d'une capacité de 20 000 positions décimales ,
- D'un lecteur - perforateur de cartes ,
- Du dispositif d'adressage indirect ,
- Du dispositif de virgule flottante automatique.

Ce dernier dispositif pourrait , à la rigueur , être remplacé par une série de sous - programmes qui existent dans la bibliothèque du SPS II , au prix d'une réduction de la capacité de traitement.

La traduction des programmes APB et l'exécution des programmes résultants , ont lieu sur la même machine et nécessitent l'une et l'autre tous les dispositifs cités plus haut.

PROBLEMES POSES PAR L'UTILISATION DE L'APB SUR L'ORDINATEUR I620

L'ordinateur 1620 est un calculateur d'un plan logique très différent de celui du Gamma Tambour. Ce dernier possède une mémoire à mots fixes, un langage-machine à une adresse ; les opérations se font par l'intermédiaire de mémoires opérateur. L'ordinateur 1620 a un langage machine à deux adresses , une mémoire à mots variables adressable position par position , pas de mémoire opérateur.

Certaines particularités du langage APB sont dues aux caractéristiques techniques du Gamma Tambour et deviennent

.../...

au moins arbitraires et même génantes sur l'ordinateur 1620 ; notamment : les instructions d'entrée et sortie , les règles d'écriture des programmes et particulièrement le dessin des cartes-programme qui sont liés à la structure des organes d'entrée et sortie du Gamma Tambour , les opérandes blancs qui sont liés à la façon dont sont réalisées les opérations en virgule flottante.

Or si on peut envisager plusieurs représentations machine pour un langage très souple comme ALGOL qui a d'ailleurs prévu cette possibilité , cela est plus difficile pour le langage APB qui a une structure plus rigide et qu'il est plus délicat de modifier même au niveau de simples règles d'écriture , sans le dénaturer.

Il a donc été admis qu'un programme APB écrit pour le calculateur Gamma Tambour , devait pouvoir être traduit et exécuté sur l'ordinateur 1620 sans modification , en tenant compte toutefois de la différence des codes-cartes BULL et IBM.

Cependant , les caractéristiques de l'ordinateur 1620 sur lequel ce traducteur a été étudié ont empêché de tirer toutes les conséquences de ce principe. Il possède en effet une mémoire d'une capacité cinq fois moindre que celle de Gamma Tambour à 64 pistes ce qui a entraîné inévitablement des restrictions.

Il existe bien , sur le même ordinateur 1620 , un traducteur FORTRAN à peu près complet ; mais , outre qu'il exige de nombreuses manipulations , il ne permet de traiter que des problèmes réduits.

Finalement il a semblé plus intéressant de considérer ce traducteur comme un travail préparatoire à un traducteur APB sur un ordinateur 1620 possédant 40000 ou 60 000 positions de mémoire rapide ou même une unité de disques 1311.

.../...

Dans cette optique , les restrictions faites à l'APB sont de nature à pouvoir être supprimées facilement et à apporter en contrepartie un gain de place important en mémoire. Principalement tous les TOS correspondant à des fonctions standard ont été supprimés.

La diminution d'encombrement ainsi obtenue a permis de garder la commodité d'emploi de l'APB sur Gamma Tambour. En particulier le programme résultant n'est pas extrait sur cartes mais reste en mémoire et est exécutable immédiatement. L'opération d'intercalation a été conservée.

En outre , il est devenu possible de traiter des problèmes d'une certaine importance. Un programme APB de 316 niveaux et 199 opérandes symboliques a pu être traduit et exécuté ; mais naturellement , les dimensions maximum admises des variables indicées restemt nettement plus faibles que sur le Gamma Tambour.

RESTRICTIONS A L'APB

-Les TOS correspondant aux fonctions standard n'existent pas , c'est à dire : SICO , COSI , ATG , APB , CTTG , LOG , RAC2 , RAC3 , EXP , SH , CH , ASIN , ACOS.

-Les TOS SICI n'existent pas.

-L'écriture des nombres avec points sexagésimaux est interdite.

-Les sorties de résultats se font uniquement sur machine à écrire , en virgule flottante , dans le format existant sur l'ordinateur 1620.

Il n'est pas tenu compte des codes particularisateurs des TOS IPRG et IMPR.

-Les TOS ALL5 , ALL6 , ALL7 , RET5 , RET6 , RET7 , n'existent pas.

.../...

En outre le procédé consistant à imprimer des titres au moyen des TOS LIRE et IMPR , est strictement interdit.

Deux extensions de langage , concernant les opérandes indicés , ont été faites.

- Il est possible de faire progresser à rebours les indices.

- On peut employer plus de 16 opérandes indicés différents. La limite pratique est constituée par la dimension de la zone de stockage des variables indicées.

Toutes les autres possibilités de l'APB ont été conservées. En particulier , les règles d'emploi des opérandes blancs sont rigoureusement identiques à celles décrites dans la Notice d'Emploi APB qui a servi de référence.

.../...

CHAPITRE II ORGANISATION DU PROGRAMME RESULTANT

PROGRAMME OBJET

La structure du langage APB rend les problèmes de traduction relativement simples à résoudre et sans commune mesure avec la compilation d'un langage comme ALGOL. La difficulté essentielle a consisté à trouver un système de programme résultant de faible encombrement , quitte à sacrifier la vitesse d'exécution. Ceci a été obtenu en faisant largement appel à une technique d'interprétation.

A partir d'une instruction symbolique du programme APB ou programme source , le traducteur produit une pseudo - instruction qui est gardée en mémoire. L'ensemble des pseudo - instructions correspondant à un programme - source donné , constitue le programme objet.

Pendant la phase d'exécution , le programme - objet est pris en charge par un interpréteur.

Le langage APB est ainsi traduit dans un langage objet qui serait celui d'une calculatrice possédant :

- Une mémoire à mots fixes ,
- Un langage - machine à trois adresses ,
- Deux mémoires opérateur de 11 positions chacune appelées Z1 et Z2 ,
- Des registres d'index à progression automatique en nombre variable.

.../...

La méthode des registres d'index simulés , à progression automatique , qui assure le traitement des opérandes indicés , a été reprise de l'APB sur Gamma Tambour. Elle s'est révélée la plus commode à employer dans le cadre du système de programme résultant choisi.

Toutes les instructions de ce langage objet sont réalisées par des appels de sous - programmes implantés à des places fixes en mémoire.

MODE DE STOCKAGE DES NOMBRES

Il n'y a pas en APB d'équivalent aux déclarations ALGOL "ENTIER" et "BOOLEEN". Les opérandes numériques , les valeurs à un instant donné du calcul de toutes les variables , sont considérés comme des nombres réels et stockés sous forme flottante en mémoire. Ils interviennent sous cette forme dans les calculs , sauf les deux premiers opérandes des TOS INRG , LIRG , DMRG , IPRG qui sont transformés en entiers fixes avant leur utilisation.

La transformation en virgule flottante est faite systématiquement par le traducteur pour les opérandes numériques ; lors de l'exécution des TOS LIRE et LIRG pour les données.

Pour les programmes APB , on utilise sur l'ordinateur 1620 , un format de virgule flottante à 9 chiffres de mantisse qui est un peu différent de celui en usage sur le Gamma Tambour.

Dans ce format un nombre occupe une zone de 11 positions.

$\overline{M}$ M M M M M M M M $\overline{C}$ C

.../...

La caractéristique occupe les deux positions d'extrême droite. Le cadrage des nombres est le même que sur le Gamma Tambour mais il n'y a pas addition de 50 à la caractéristique qui peut être négative et varier de + 99 ($\overline{99}$) à - 99 ($\overline{9}\,\overline{9}$). Le signe - de la caractéristique est indiqué par une marque de mot sur sa position de droite : $\overline{C}\,\overline{C}$. De même si la mantisse est négative , son signe est indiqué par une marque de mot sur sa position de droite :

$$\overline{M}\ M\ M\ M\ M\ M\ M\ M\ \overline{M}\ \overline{C}\ C$$

C'est dans ce format que se font toutes les sorties de résultats.

ORGANISATION DE LA ZONE DE STOCKAGE

TABLE DES OPERANDES

Cette zone constitue en quelque sorte la mémoire à mots fixes de la calculatrice simulée par le programme résultant. Elle est divisée en deux parties dont l'une , formée de mots de 19 positions , est affectée aux opérandes symboliques et numériques , l'autre , formée de mots de 11 positions , aux variables indicées.

La première partie joue un double rôle de zone de stockage et de table des opérandes.

Pendant la traduction d'un programme APB , une table des opérandes est construite , qui sert de témoin de l'analyse et de l'affectation des opérandes. Elle doit pouvoir fournir le type de l'opérande et l'adresse de la zone qui lui est affectée. Cette table doit être gardée en mémoire une fois la traduction terminée , en vue d'opérations d'intercalation ultérieures.

.../...

Pour gagner de la place , la table d'opérandes utilisée par le programme de traduction est combinée avec la zone de stockage des valeurs numériques. Elle est formée de blocs de 19 positions occupant la mémoire à partir de l'adresse 19999 vers les adresses inférieures. Le nombre de ces blocs est limité à 999 ce qui permet de leur faire correspondre un numéro de trois chiffres.

Dans cette table sont gardés les opérandes symboliques et numériques , les registres d'index , les compteurs des TOS REC. Voici la structure des blocs qui leur correspondent.

- Opérande symbolique : variable simple.

$\overline{S}SSSSSSS\ \overline{X}XXXXXXXXX\overline{X}$

Le bloc est formé d'une partie identificateur de 8 positions réservées à l'opérande symbolique et une partie valeur numérique de 11 positions destinée à recevoir la valeur numérique de l'opérande au cours du calcul.

- Opérande numérique

$\overline{6}0000000\ \overline{X}XXXXXXXXX\overline{X}$

Pour tous les opérandes numériques , la partie identificateur contient la constante $\overline{6}0000000$. La partie valeur numérique reçoit , lors de la traduction , l'opérande numérique.

- Compteur de TOS REC

$\overline{4}0000000\ \overline{X}XXXXXXXXX\overline{X}$

A chaque TOS REC du programme APB , correspond un compteur qui est stocké dans la table des opérandes. La partie identificateur contient toujours $\overline{4}0000000$. La partie valeur numérique le compteur lui - même en virgule flottante.

.../...

Les constantes $\overline{4}0000000$ et $\overline{6}0000000$ ont été choisies parce qu'elles ne correspondent à aucun symbole alphanumérique possible.

- Registre d'index

| $\overline{R}RRRR\overline{m}mmmm\overline{M}MMMM0000$ | $\overline{S}SSSSSSS0000\overline{X}XXXX$ ≠ |

2ème Bloc — 1er Bloc

A chaque registre d'index, sont affectés deux blocs contigus. Le premier bloc contient dans sa partie identificateur l'opérande indicé lui-même, dans ses 6 dernières positions l'adresse de l'élément du tableau sur lequel on opère, suivie d'une marque d'enregistrement. Le deuxième bloc comporte dans ses 15 premières positions

- Le pas de progression de l'adresse de l'élément du tableau sur lequel on opère
- L'adresse du premier élément du tableau
- L'adresse du dernier élément du tableau

On voit que l'analyse d'un bloc de la table des opérandes permet de savoir à quel type d'opérande il a été affecté.

La seconde partie de la zone de stockage, réservée aux tableaux de nombres, a une structure plus simple. Elle est formée par les éléments successifs de ces tableaux, rangés dans l'ordre des adresses décroissantes, à raison de 11 positions de mémoire par élément.

.../...

PSEUDO - INSTRUCTIONS DU PROGRAMME OBJET

Chaque niveau APB du programme source donne naissance dans le programme objet , à un mot de 11 positions dont la structure est la suivante.

$\overline{X}XX$	YYY	ZZZ	$\overline{T}T$
A1	A2	A3	CO

Il est formé de trois pseudo-adresses qui sont des numéros de blocs de la table des opérandes et d'une code opération.

Le code opération représentant le TOS est obtenu de la manière suivante :

Il existe en mémoire une table des 26 TOS permis qui contient en regard de chaque TOS l'adresse du sous programme spécifique. Cette table est donc formée de 26 blocs de 13 positions : 8 pour le TOS , 5 pour l'adresse

$\overline{T}TTTTTTT$ $\overline{X}XXXX$

Chacun de ces blocs peut être repéré par un numéro de 1 à 26 , c'est ce numéro qui constitue le code opération de la pseudo - instruction.

Les trois pseudo-adresses A1 , A2 , A3 , sont respectivement les numéros des blocs de la table des opérandes affectés au premier , au deuxième , au troisième opérande des instructions symboliques APB. Ce sont des entiers de trois chiffres.

Dans le cas où un opérande est absent , soit qu'il ne soit pas significatif , soit qu'il ait été laissé en blanc, la pseudo - adresse correspondante est 00 ≠

.../...

Avec les TOS à rupture de séquence : ALL , REC , SIEG , SISU , SIIN , SIDI , ALL1 , ALL2 , ALL3 , ALL4 , RET1 , RET2 , RET3 , RET4 , le troisième opérande n'est jamais significatif. La pseudo-adresse A3 est constituée , alors , par le numéro de renvoi qui est un entier de trois chiffres.

Le second opérande n'est jamais non plus significatif avec le TOS REC. La pseudo-adresse correspondante A2 est formée par le numéro du bloc affecté au compteur dans la table des opérandes.

Les pseudo-adresses ne portent aucune indication relative au type de l'opérande auquel elles correspondent. Ces indications sont obtenues directement par analyse du bloc de la table des opérandes.

Les pseudo-instructions du programme objet sont rangées en mémoire sequentiellement vers les adresses croissantes. 11 positions de mémoire sont réservées pour les niveaux ne comportant aucune instruction , ce qui permet , d'une part d'intercaler ultérieurement un niveau , d'autre part , de trouver facilement l'adresse d'une pseudo-instruction à partir du numéro de niveau de l'instruction APB correspondante.

Aucune pseudo-instruction ne correspond à l'instruction symbolique comportant le TOS FIN mais l'adresse qui aurait dû lui correspondre est mise en réserve et utilisée dans un test d'arrêt du calcul.

ORGANISATION GENERALE DE LA MEMOIRE

CAPACITE DE TRAITEMENT

L'organisation de la mémoire est la même pendant la traduction du programme source et pendant l'exécution du programme résultant ; ce qui facilite l'intercalation.

.../...

Le programme objet est rangé en mémoire à partir de l'adresse 8500.

La table des opérandes se développe à partir de l'adresse 19999 vers les adresses inférieures , à la rencontre du programme objet.

Toute la zone comprise entre la fin de la table des opérandes et la fin du programme objet est affectée aux tableaux de nombres. Le premier élément du premier tableau est rangé immédiatement à la fin de la table des opérandes.

La table de TOS occupe 338 positions de mémoire à partir de l'adresse 626.

L'ensemble donne le schéma de la page 16

La table des opérandes , les tableaux , le programme objet , peuvent occuper au maximum 11500 positions.
soit :

N le numéro de niveau le plus grand : celui du niveau FIN

S la somme des dimensions des tableaux

R le nombre de registre d'index

C le nombre total de TOS REC du programme

P le nombre total d'opérandes numériques , et d'opérandes symboliques différents.

Le programme objet occupe 11 * (N-1) positions,
Les tableaux, 118 positions

Chaque registre d'index occupe 2 fois 19 positions , les opérandes numériques , symboliques , les compteurs de TOS REC 19 positions chacun ; soit : 19 (2R + C + P) positions pour la table des opérandes.

.../...

Pour qu'un programme APB puisse être traité , il faut que l'inégalité suivante soit vérifiée.

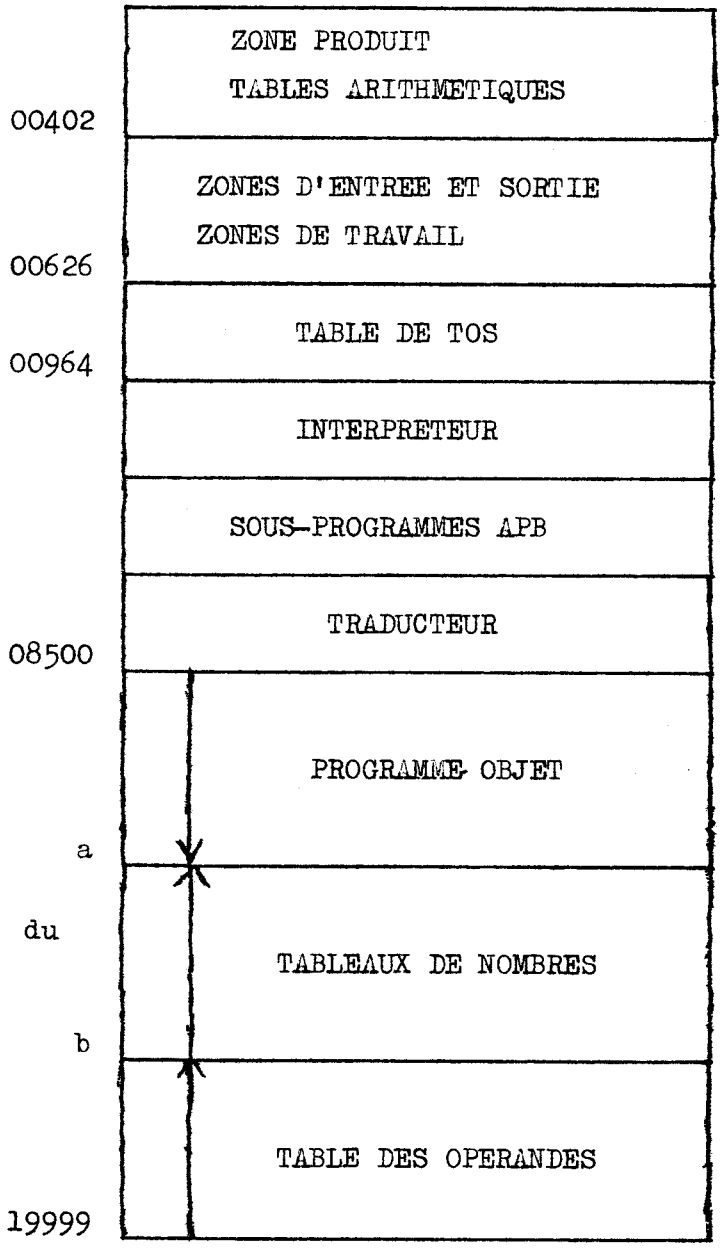
$$11\ (n - 1 + S) + 19\ (2R + P + C) < 11500$$

Une autre inégalité plutôt théorique doit être satisfaite :

$$2R + P + C \leq 999$$

pour que le numéro d'un bloc de la table des opérandes soit un nombre de trois chiffres.

.../...



a et b dépendent du programme A P B

.../...

CHAPITRE III ORGANISATION DU TRADUCTEUR

GENERALITES

On peut distinguer trois phases de principe dans la production d'un programme en langage machine ou en langage objet , à partir d'un programme source écrit dans un langage symbolique un peu complexe.

- Edition

Cette phase assure la transformation du programme symbolique en une chaine codée et normalisée , équivalente syntactiquement.

- Génération de la notation post-fixée ou préfixée

Il s'agit de la transformation d'une écriture avec parenthèses en une écriture sans parenthèse , écriture qui indique la suite des opérations à effectuer.

- Génération du programme objet à partir de la notation pré ou post - fixée

Suivant le traducteur et la complexité du langage source , ces trois phases peuvent se trouver plus ou moins imbriquées , plus ou moins réduites.

Dans le cas de l'APB , la phase d'Edition est pratiquement réduite à la transformation des numéros de niveau de renvoi et des opérandes numériques de l'écriture en code alphanumérique à deux digits par caractère à l'écriture en code numérique.

.../...

La phase de génération de la notation préfixée n'existe pas. Le langage APB ne comporte pas de parenthèses. La structure des instructions est telle qu'un programme APB est déjà pratiquement en notation préfixée.

Seule la 3ème phase a reçu un certain développement. Encore est elle simplifiée par le fait que le langage objet choisi est très proche du langage APB.

GENERATION DU PROGRAMME - OBJET

La traduction d'une instruction symbolique APB ne nécessite que la connaissance de cette instruction et celle de la table des opérandes construite lors de la traduction des niveaux précédents. Il est donc possible de faire la traduction en carte à carte ; c'est à dire qu'une carte - programme est lue , l'instruction qu'elle porte est traduite , la pseudo-instruction objet rangée en mémoire, puis la carte-programme suivante est lue et détruit l'instruction symbolique précédente.

Les opérations de traduction d'une instruction symbolique sont les suivantes: (elles ne dépendent pratiquement pas du TOS.).

- Analyse du numéro de niveau

Le numéro de niveau est transposé du code alpha-numérique au code numérique , comparé au numéro de niveau de l'instruction précédente pour déceler un éventuel déclassement , puis mis en réserve.

- Fabrication du code opération correspondant au TOS

Le TOS est d'abord comparé au symbole FIN.
S'il n'y a pas identité , on recherche dans la table de TOS un TOS identique.
Si on en trouve un , le numéro du bloc de la table qui le contient forme le code opération.
S'il n'y en a pas un message d'erreur est imprimé sur machine à écrire.

.../...

- Analyse des opérandes

Le type de l'opérande est déterminé de la façon suivante :

- La zone de la carte programme correspondant à l'opérande analysé est vide ; il s'agit d'un opérande blanc. La pseudo-adresse est 00≠. L'analyse et l'affectation de l'opérande sont terminées.

- La zone correspondant aux huit dernières positions de l'opérande n'est pas vide : l'opérande est numérique.

- Cette zone est vide ; il y a un point dans les quatre premières positions de l'opérande : c'est un opérande numérique.

- Il n'y a pas de point : c'est un opérande symbolique ou indicé.

- Le symbole commence par RG : c'est un opérande indicé

- Le symbole ne commence pas par RG : c'est un opérande symbolique

- Affectation des opérandes et constructions de la table des opérandes

Une fois le type de l'opérande reconnu , il est affecté.

Le numéro du premier bloc libre de la table des opérandes est gardé en mémoire et augmenté de 1 chaque fois qu'un bloc , libre jusque là , est affecté à un opérande.

Si l'opérande est symbolique , il est comparé à la partie identificateur de tous les blocs de la table jusqu'à ce qu'une des deux éventualités se produise :

On trouve un bloc dont la partie identificateur est semblable. Le numéro de ce bloc forme la pseudo-adresse qui remplace l'opérande. On passe à l'opérande suivant.

.../...

On arrive sur un bloc libre. L'opérande est envoyé dans la partie identificateur de ce bloc qui lui est affecté. Son numéro devient la pseudo-adresse correspondant à l'opérande

- Si l'opérande est indicé , les opérations sont les mêmes que pour un opérande symbolique mais , s'il ne figure pas encore dans la table , on lui affecte les deux premiers blocs libres contigus. Celui qui a le plus petit numéro . reçoit dans sa partie identificateur l'opérande indicé ; son numéro forme la pseudo-adresse correspondante. Une marque d'enregistrement est mise dans la dernière position de ce bloc.

- S'il s'agit d'un opérande numérique , il est d'abord transposé du code alphanumérique au code numérique puis transformé en virgule flottante.

On balaye alors la table des opérandes pour rechercher tous les blocs contenant $\overline{6}0000000$ dans leur partie identificateur. L'opérande numérique est comparé avec la constante flottante qui y est stockée.

S'il y a égalité l'affectation est terminée.

Si la constante n'a pas encore été stockée en mémoire , on lui affecte le premier bloc libre. La constante $\overline{6}0000000$ est envoyée dans sa partie identificateur et l'opérande numérique dans les 11 positions suivantes. Le numéro du bloc forme la pseudo-adresse qui remplace dans l'instruction objet l'opérande numérique.

Le traitement des opérandes blancs est terminé dès que la recherche du type de l'opérande est finie.

- Analyse du numéro de renvoi

On teste la présence du numéro de renvoi. S'il existe , il est transposé du code alphanumérique au code numérique et envoyé en position A3 de la pseudo-instruction.

- Cas du TOS REC

Quand toutes les opérations précédentes sont terminées on regarde si le TOS est REC.

.../...

Si oui , le premier bloc libre de la table est affecté au compteur et l'indicatif $\overline{4}0000000$ envoyé dans sa partie identificateur.

La traduction de l'instruction APB est alors terminée et on se boucle sur l'instruction de lecture d'une carte programme.

MESSAGES D'ERREUR EN COURS DE TRADUCTION

Trois types d'erreur sont détectées pendant la traduction et provoquent l'impression d'un libellé sur la machine à écrire.

-Les numéros de niveau successifs ne sont pas dans un ordre croissant au sens strict. Le libellé suivant est imprimé.

XXX DECLASSE

XXX est le numéro du niveau déclassé

-Le TOS ne figure pas dans la table de 26 TOS permis

XXX TOS FAUX

XXX est le numéro du niveau incorrect.

Dans ces deux cas la traduction peut être reprise à son début en appuyant sur DEPART après rectification du programme source.

Le programme objet et la table des opérandes risquent de se chevaucher. Le message suivant est imprimé

DEP DE CAPACITE

Dans ce dernier cas la traduction doit être reprise comme s'il s'agissait d'un nouveau programme.

.../...

CHAPITRE IV EXECUTION DU PROGRAMME RESULTANT

PROGRAMME D'INTERPRETATION

Le programme objet , tel qu'il est fabriqué par le traducteur , est évidemment inéxécutable directement. Il existe donc en mémoire , un programme qui décode les pseudo-instructions du programme objet et assure leur exécution.

Le programme d'interprétation assure également toutes les initialisations , les appels et renvois de valeurs numériques en mémoires opérateur Z1 et Z2 , le passage en séquence, sur les zones vides correspondant à des niveaux vides d'instruction et effectue enfin le test d'arrêt du calcul.

La structure de ce programme d'interprétation est telle qu'un programme d'analyse pourrait facilement y être inclus.

INITIALISATIONS

En APB , il n'est pas nécessaire de définir la valeur numérique initiale d'un opérande avant de l'employer. On suppose qu'elle est égale à zéro. Or , sur l'ordinateur 1620 , le zéro flottant a la forme : $\overline{0}0000000\overline{99}$.

Il ne suffit donc pas de réserver des zones de mémoire pour les valeurs numériques des opérandes , il faut leur donner la valeur initiale zéro.

Avant de commencer l'exécution du programme objet , le programme d'interprétation balaye la table des opérandes et met la constante flottante $\overline{0}0000000\overline{99}$ dans la partie valeur numérique de tous les blocs affectés à des opérandes symboliques ou à des compteurs de TOS REC. Les blocs affectés à des opérandes numériques ou indicés sont épargnés dans ce balayage.

.../...

Puis toute la zone réservée aux tableaux de nombres , entre la fin de la table des opérandes et la fin du programme objet est systématiquement garnie avec la constante zéro flottant ,puisque on ne sait pas d'avance quelle fraction de cette zone sera occupée.

Enfin , l'adresse du premier élément du 1er tableau , c'est à dire l'adresse de la fin de la table des opérandes , adresse qui peut varier avec le programme APB , sert à initialiser un mot de 5 positions appelé IMPLT qui contiendra tout au long du calcul l'adresse de la première position libre de la zone de stockage des tableaux.

Ces initialisations , en particulier l'initialisation des opérandes symboliques , auraient pu se faire pendant la génération du programme-objet. Elles ont été groupées en une séquence du programme d'interprétation en vue de faciliter l'opération de reprise.

DECODAGE ET UTILISATION DU CODE OPERATION

Le rôle du programme d'interprétation dans l'exécution d'une pseudo-instruction , consiste essentiellement à appeler un sous-programme spécifique d'un TOS.

Le programme d'interprétation utilise deux zones de travail , l'une de 5 positions appelée AD qui contient l'adresse de la pseudo-instruction à exécuter , l'autre appelée ZI contient cette pseudo-instruction elle-même. AD contient 8510 en début d'exécution. On considère que l'adresse d'une pseudo-instruction est celle de sa position d'extrême droite.

Deux tests préliminaires sont d'abord effectués. On s'assure que l'adresse contenue dans AD n'est pas égale à l'adresse de fin de programme calculée pendant la traduction. Dans ce cas on imprime FIN et le calcul s'arrête On vérifie ensuite que la zone à l'adresse contenue dans AD n'est pas vide , ce qui pourrait arriver puisque 11 positions sont réservées dans le programme-objet aux niveaux vides. Dans ce cas AD est incrémenté de 11 et on recommence les tests.

.../...

Quand le résultat de ces tests est favorable , le programme d'interprétation calcule l'adresse du bloc de la table des TOS qui correspond au code opération de la pseudo-instruction. Cette adresse est donnée par la formule

977 - 13. C où C est le code opération

Dans ce bloc figure l'adresse du sous-programme adéquat. Cette adresse est utilisée pour compléter une instruction de liaison au sous-programme.

Le programme d'interprétation ne s'occupe pas directement des pseudo-adresses qui sont prises en charge et décodées par des sous-programmes d'appel et de renvoi d'opérande.

APPELS ET RENVOIS D'OPERANDES

Toutes les opérations sur les valeurs numériques des opérandes , sont effectuées dans les mémoires opérateur Z1 et Z2 et le résultat se retrouve en Z2 , ceci pour obtenir simplement des règles d'emploi d'opérandes blancs semblables à celles de la Notice d'Emploi APB. Il existe donc deux sous-programmes d'appel en Z1 et Z2 : S1 et S2 et un sous-programme de renvoi de Z2 : R1.

Ces sous-programmes tiennent compte, pour effectuer ces déplacements, du type d'opérande.

Une pseudo-adresse est en général un numéro de bloc de la table des opérandes. On commence par tester si c'est 00≠. Dans ce cas on a affaire à un opérande blanc. Aucun mouvement n'est effectué. Si la pseudo-adresse est différente de 00 ≠ on calcule l'adresse réelle A du bloc de la table des opérandes qui lui correspond.

A = 20018 - N * 19

où N est la pseudo-adresse.

.../...

En tenant compte du mode de fabrication de ces blocs pendant la traduction , si à l'adresse A figure une marque d'enregistrement , on a affaire à un opérande indicé. Les deux blocs contigus à l'adresse A forment le registre d'index associé. L'adresse de la valeur numérique appelée ou l'adresse de renvoi du résultat est en A - 1.

On vérifie que cette adresse est bien comprise entre l'adresse du premier et du dernier élément du tableau correspondant. Dans le cas favorable , le déplacement est réalisé puis on fait progresser le registre d'index.

Si la position à l'adresse A ne contient pas une marque d'enregistrement , la pseudo-adresse correspond à un opérande symbolique , numérique , ou à un compteur de TOS REC. Dans ce cas A est l'adresse de la valeur numérique à appeler ou l'adresse de rangement d'un résultat.

UTILISATION DES SOUS - PROGRAMMES D'APPEL ET DE RENVOI

C'est en principe dans les sous-programmes correspondant aux TOS que doivent figurer les instructions de liaison aux sous-programmes d'appel et de renvoi d'opérandes. On a cependant tiré parti du fait que pour plusieurs TOS les mêmes déplacements sont à réaliser.
Pour les TOS , ADD , SOU , MUL , DIV par exemple , il faut appeler en Z1 la valeur numérique du premier opérande , en Z2 la valeur numérique du second opérande. Le résultat est en Z2 et doit être renvoyé de cette mémoire.

Pour utiliser ce fait , l'instruction de liaison au sous-programme spécifique d'un code opération , dans le programme d'interprétation , est précédée d'une instruction I1 de liaison à S1 , d'une instruction I2 de liaison à S2 et suivie d'une instruction I3 de liaison à R1.

La présence ou l'absence d'une marque de mot sur les positions 2,3,4 des adresses de sous-programmes dans la table de TOS permet de savoir s'il faut ou non exécuter I1 , I2 , I3.

.../...

Ainsi pour ADD , SOU , MUL , DIV , qui ont deux opérandes donnée et un opérande résultat , I1 , I2 , et I3 sont exécutées. L'adresse à la forme $\overline{M}\,\overline{M}\,\overline{M}\,\overline{M}\,M$.

Pour ALG et ABS qui ont un opérande donnée et un opérande résultat , I1 et I3 sont exécutées. L'adresse est de la forme $\overline{M}\,\overline{M}\,M\,\overline{M}\,M$.

Pour certains TOS , les mouvements d'opérandes ne peuvent être réalisés par l'utilisation partielle ou totale de la séquence I1 , I2 , I3. Les sous-programmes correspondant comportent leurs propres instructions de liaison à S1 , S2 et R1. C'est le cas pour LIRE , IMPR , DMRG , INRG , LIRG et IPRG.

PROBLEMES D'EXECUTION PARTICULIERS A CERTAINS TOS

- TOS à rupture de séquence

Dans le cas des TOS à rupture de séquence , la troisième pseudo-adresse est en fait le numéro de niveau de l'instruction dont l'exécution doit suivre.

Le programme objet étant rangé séquentiellement , il est facile de calculer l'adresse A de la pseudo-instruction qui lui correspond.

Cette adresse , est donnée par la formule

$$A = 8499 + 11.N$$

où N est le numéro de renvoi qui forme la troisième pseudo-adresse.

A remplace dans AD l'adresse de la pseudo-instruction en séquence , après quoi on se renvoie au programme d'interprétation.

.../...

IMPLANTATION DES TABLEAUX

L'APB permet de traiter des tableaux à une dimension dont l'une des bornes au moins est variable. Aussi le traducteur se contente - t - il de réserver des blocs de la table des opérandes pour les registres d'index. Leur contenu n'est précisé que pendant l'exécution d'instructions comportant des TOS INRG et DMRG.

Un mot de 5 positions appelé IMPLT contient en début d'exécution l'adresse de la fin de la table des opérandes +1. Le contenu de IMPLT est modifié à chaque DMRG relative à un nouveau registre en tenant compte du second opérande de la DMRG de façon que IMPLT fournisse toujours l'adresse du premier élément du tableau suivant.

DMRG calcule , à partir de l'adresse contenue dans IMPLT , l'adresse du dernier élément du tableau associé. Ces deux adresses réunies , fournissent la dimension et l'implantation du tableau.

INRG précise , pour le tableau associé , l'adresse du premier élément à considérer dans les calculs et le pas de progression de cette adresse.

De toutes les quantités contenues dans un registre d'index , une seule : l'adresse du premier élément c'est à dire l'implantation est fixée définitivement. Quand on exécute une DMRG , on vérifie donc si le registre correspondant ne possède pad déjà une adresse d'implantation. Dans ce cas celle ci n'est pas modifiée.

.../...

ENTREES ET SORTIES

L'introduction de données lors de l'exécution des TOS LIRE et LIRG , se fait uniquement par cartes qui doivent être perforées suivant le même format que pour le Gamma Tambour.

Le seul organe de sortie est la machine à écrire. Les résultats sont imprimés selon une mise en page identique à celle de l'APB sur Gamma Tambour.

L'impression , facultative , du contenu des cartes données , est obtenue par un renvoi aux sous programmes correspondant aux TOS IMPR et IPRG , suivant le cas , après que les données aient été transposées du Code alphanumérique au Code numérique , transformées en virgule flottante et rangées dans la zone de stockage. C'est pourquoi les données sont imprimées sous forme flottante même si elles sont perforées avec point décimal sur les cartes données. Les libellés LIRE RGXX et INTRO permettent de distinguer les impressions de données de celles de résultats.

MESSAGES D'ERREUR EN COURS D'EXECUTION

Deux types d'erreurs , concernant toutes deux l'utilisation des registres d'index , sont détectées pendant l'exécution du programme résultant et provoquent l'impression de messages par la machine à écrire.

Pendant l'exécution de chaque DMRG, on vérifie que la dimension du tableau associé est compatible avec la place disponible en mémoire. S'il y a risque de chevauchement avec le programme objet , le message

DEP DE CAPACITE

est imprimé et le calcul s'arrête.

.../...

On vérifie également , lors de l'exécution des sous-programmes d'appel et de renvoi d'opérande que l'indice correspondant à un élément de tableau à déplacer , est inférieur ou égal à la dimension du tableau. Sinon le message suivant est imprimé :

HORS LIMITES $\overline{X}XX$ YYY $\overline{Z}ZZZ$

$\overline{X}XX$ est le numéro de niveau de l'instruction dont l'exécution a provoqué l'erreur.

YYY est l'opérande indicé.

$\overline{Z}ZZZ$ est l'indice erroné.

REPRISE ET INTERCALATION

La reprise consiste simplement en un branchement à l'adresse du début de programme d'interprétation. Elle ne comprend aucune séquence de programme particulière.

Pour l'intercalation , toutes les instructions à intercaler en une même opération , sont supposées rangées dans l'ordre des numéros de niveau croissants. On entre par machine à écrire le numéro de niveau de la première instruction et celui de la dernière.

La traduction d'une instruction à intercaler comprend les mêmes opérations que celle de n'importe qu'elle instruction d'un programme APB ; mais après chaque traduction d'une instruction , le traducteur se renvoie à une séquence d'intercalation qui vérifie s'il y a encore des instructions à intercaler. La séquence d'intercalation assure également la transformation du traducteur nécessaire à ce test et sa remise en état.

.../...

CHAPITRE V OPERATIONS PRATIQUES

MANIPULATIONS

Les manipulations à effectuer pour traduire et exécuter un programme APB sont à peu près les mêmes que sur le Gamma Tambour.

1°) - Introduire le traducteur APB s'il n'est pas encore en mémoire

2°) - Mettre dans le lecteur de cartes le programme source précédé d'une carte traduction portant dans ses 7 premières colonnes 4905736 et suivi des cartes donnée. Il ne faut pas mettre de carte vierge entre le programme source et les données.

3°) - Appuyer sur "Chargement". Le programme est introduit et traduit. Quand la traduction est terminée le message FIN DE TRADUCTION est imprimé et la machine s'arrête sur une halte programmée.

4°) - Appuyer sur "Départ". Le programme résultant est exécuté. Quand l'exécution est terminée le message FIN est imprimé et l'ordinateur 1620 s'arrête sur une halte programmée.
Suivant la position de la clé 1 on peut : soit faire une nouvelle traduction , soit reprendre l'exécution du programme résultant.

L'opération de reprise peut s'effectuer de la manière suivante :

1°) - Mettre dans le lecteur de cartes les cartes donnée précédées d'une carte reprise portant dans ses 7 premières colonnes 4900964

2°) - Appuyer sur "Chargement".

.../...

Pour l'intercalation :

1°) - Mettre dans le lecteur de carte les niveaux à intercaler dans l'ordre des numéros croissants , précédés d'une carte intercalation portant dans ses 7 premières colonnes 4907912

2°) - Appuyer sur "Chargement"

3°) - Entrer par machine à écrire le numéro du premier niveau et celui du dernier niveau

4°) - Quand l'intercalation est terminée , effectuer une reprise.

REGLAGE DE LA MACHINE A ECRIRE

Pour que les résultats et les données soient imprimés selon une mise en page identique à celle de l'APB sur Gamma Tambour et pour qu'on obtienne une liste correcte des programmes sources , le règlage de la machine à écrire doit être le suivant : Si on suppose que la marge gauche est en M

- Taquets de tabulation en M + 8 , M + 24 , M + 36 , M + 52 , M + 64
- Marge droite au delà de M + 80

Ce règlage doit être le même pour la phase de traduction et la phase d'exécution.

POSITION DES CLES DE PUPITRE

Clés d'incident

Pendant la traduction et l'exécution des programmes les clés d'incident doivent être sur "ARRET" sauf la clé de dépassement de capacité qui doit être sur "PROGRAMME".

.../...

Clés de programme

La position de ces clés peut être choisie par l'opérateur.

Pour la traduction :

- clé 1
  - Haute : introduction du programme source par machine à écrire
  - Basse : introduction du programme source par lecteur de carte
- clé 2
  - Haute : annulation de l'introduction d'ume instruction par machine à écrire
  - Basse : traduction de l'instruction introduite
- clé 3
  - Haute : impression d'une table des symboles en fin de traduction
  - Basse : pas d'impression des données
- clé 4
  - Haute : le programme source est imprimé sur machine à écrire
  - Basse : pas de liste du programme source

Pour l'exécution du programme résultant

- clé 3
  - Haute : déroulement instruction par instruction appuyer sur "Départ" après chaque instruction
  - Basse : déroulement sans interruption
- clé 4
  - Haute : impression des données
  - Basse : pas d'impression

.../...

clé 1 :
- Haute : renvoi au début du programme d'interprétation en fin de calcul
- Basse : renvoi au début du programme de traduction en fin de calcul

clé 2 : inutilisée ; position quelconque

Les adresses de début du traducteur et du programme d'interprétation sont 05736 et 00964.

En cas d'incident se terminant sur un "ARRET ERREUR" il vaut mieux introduire à nouveau le traducteur APB.

Pour faciliter le ré-emploi des programmes APB déjà écrits pour le Gamma Tambour , il existe un programme de transcodage qui transforme les cartes perforées suivant le code BULL en cartes perforées dans le code IBM

.../...

CHAPITRE VI TEMPS DE TRADUCTION ET DE CALCUL COMPARAISON AVEC LE GAMMA TAMBOUR

TEMPS DE TRADUCTION

Il comprend le temps de traduction proprement dit et le temps d'impression d'une liste du programme source. Ce dernier est imprimé sur la machine à écrire de l'ordinateur 1620 , à la vitesse de 10 caractères par seconde. C'est donc une opération très lente. Elle a été un peu accélérée en utilisant des taquets de tabulation. Au lieu d'imprimer un opérande blanc , on effectue une tabulation. Mais le temps d'impression d'une instruction reste de 4 secondes en moyenne , alors que la tabulatrice connectée au Gamma Tambour permet d'imprimer 150 lignes par minute.

Par contre , si on n'imprime pas la liste du programme source , les temps de traduction de l'ordinateur 1620 sont comparables à ceux du Gamma Tambour pour les programmes courts : 2 à 3 instructions par seconde. Pour les programmes longs , ayant beaucoup d'opérandes différents , l'ordinateur 1620 est nettement plus rapide. Pour un programme de 342 niveaux et 183 opérandes

- Sur l'ordinateur 1620 : 6 minutes 30 secondes
- Sur le Gamma Tambour : 36 minutes

Ceci est du au temps d'accès plus long de la mémoire à tambour, qui se fait sentir davantage quand le nombre d'opérandes différents augmente.

.../...

TEMPS D'EXECUTION DES INSTRUCTIONS OBJET SUR L'ORDINATEUR 1620

Le temps d'exécution d'une instruction peut se décomposer ainsi :

- Temps d'exécution de l'interpréteur :
  Ce temps est de 7 millisecondes
- Temps d'appel et de renvoi des opérandes :
  Ce temps dépend du type d'opérande
  1 milliseconde pour un opérande blanc
  5,3 millisecondes pour un opérande symbolique ou numérique
  11,5 millisecondes pour un opérande indicé
- Temps d'exécution du sous programme correspondant au TOS de l'instruction

  Voici le temps d'exécution pour les principaux TOS :

ADD	1,5	millisecondes
SOU	2,7	millisecondes
MUL	15,5	millisecondes
DIV	52,7	millisecondes

  Ce temps est principalement celui de l'exécution de l'instruction machine correspondante.

ABS	1	milliseconde
ALG	0,8	milliseconde
DMRG	22,5	millisecondes
INRG	16,7	millisecondes
ALL	1,9	millisecondes
SISU, SIDI	3,3	millisecondes si on passe en séquence
SIIN, SIEG	5,3	millisecondes si on effectue la variante

.../...

Pour les TOS d'entrée et sortie, le temps d'exécution dépend essentiellement des organes d'entrée et sortie

- 260 millisecondes environ pour la lecture d'une carte et le rangement des données.
- 12 secondes environ pour l'impression d'une ligne complète sur la machine à écrire, selon la même mise en page que sur le GAMMA TAMBOUR.

On peut faire deux remarques sur ces temps. Le temps d'exécution des TOS IMPR et IPRG et également LIRE et LIRG quand on imprime les données, est considérable par rapport à celui des autres TOS. Il diminue énormément la vitesse de calcul, et on a intérêt à réduire les sorties au maximum.

Le temps d'exécution d'une instruction objet comprend, pour une part importante, celui de l'interpréteur et des appels et renvois d'opérande.. Dans le cas d'une instruction ayant comme TOS ADD et trois opérandes indicés, le temps d'exécution du sous-programme correspondant à ADD atteint à peine 4 % du précédent. C'est une conséquence de l'organisation du programme résultant. Les essais d'exécution de programmes APB ont montré que, malgré tout, la vitesse de calcul est plus élevée sur l'ordinateur 1620 que sur le Gamma Tambour.

COMPARAISON DES TEMPS D'EXECUTION DE PROGRAMMES SUR L'ORDINATEUR 1620 ET SUR LE GAMMA TAMBOUR.

Voici les temps d'exécution de quelques programmes sur l'ordinateur 1620 et sur le Gamma Tambour.

PROGRAMME I

Ce programme calcule par la méthode de Hörner la valeur d'un polynôme de degré 14. Le calcul est recommencé 100 fois. La méthode de Hörner permet d'employer de très nombreux opérandes blancs.

- Temps d'exécution sur Gamma Tambour : 143 secondes ce qui correspond à un temps moyen de 0,06 seconde par instruction.
- Temps d'exécution sur 1620 : 50 secondes
  Temps moyen d'une instruction : 0,02 seconde.

.../...

PROGRAMME II

Ce programme est le même que le précédent mais les opérandes blancs ont été supprimés.

- Temps d'exécution sur Gamma Tambour : 230 secondes
  Temps moyen d'une instruction : 0,10 seconde
- Temps d'exécution sur 1620 : 62 secondes
  Temps moyen d'une instruction : 0,03 seconde.

Le temps de l'introduction des données et celui de la sortie des résultats n'a pas été compté dans les programmes I et II.

PROGRAMME III

Il s'agit d'un programme séquentiel calculant une série d'expressions arithmétiques compliquées pour diverses valeurs des arguments. Ce programme comporte 342 instructions et 183 opérandes différents.

Ordinateur 1620

Temps de traduction avec impression du programme source : 28 minutes

Temps de traduction sans impression du programme source : 6 minutes 30 secondes

Temps de calcul total pour 22 groupes de données : 32 minutes

Temps de calcul pour un groupe de données : 87 secondes en tenant compte des entrées et sorties
6 secondes de calcul pur
ce qui correspond à un temps moyen par instruction de 0,017 secondes.

Gamma Tambour

Temps de traduction 36 minutes

Temps de calcul total pour 22 groupes de données : 17 minutes

Temps de calcul pour un groupe de données : 46 secondes en tenant compte des entrées et sorties

.../...

31 secondes de calcul pur
0,09 seconde de temps moyen par instruction

PROGRAMME IV

Calcul des valeurs propres et des vecteurs propres d'une matrice symétrique par la méthode du gradient. Le programme comporte 211 niveaux et permet de traiter des matrices jusqu'à la dimension 24 sur 1620, jusqu'à la dimension 47 sur Gamma Tambour. Le volume des sorties de résultats est assez réduit par rapport à celui du calcul : impression des valeurs propres et des vecteurs propres correspondants.

Le calcul des valeurs propres maximum et minimum d'une matrice de dimension 4 a demandé :

- Avec l'ordinateur 1620 : 6 minutes 20 secondes
- Avec le Gamma Tambour : 26 minutes

PROGRAMME V

Programme d'intégration numérique d'une équation différentielle par la méthode de PICARD.

Le calcul d'une valeur approchée de la solution de l'équation différentielle $YY' = 1/2$ pour $X = 2$ a pris :

- Sur 1620 : 170 secondes
- Sur Gamma Tambour : 510 secondes

PROGRAMME VI

Intégration numérique d'une équation différentielle par la méthode de MILNE.

L'intégration de l'équation $Y'Y = 1/2$ pour X variant entre 0,1 et 5,4 par pas de 0,1 avec impression de la valeur de Y pour $X = 5,4$ seulement ; a demandé :

160 secondes sur 1620
et 540 secondes sur Gamma Tambour.

.../...

PROGRAMME VII

Analyse de covariance.
Sur ordinateur 1620 le programme a traité 600 données en 180 secondes.
Sur Gamma Tambour les 600 données ont été traitées en 940 secondes.
Le temps d'entrée des données n'est pas compté.

De cette série d'essais se dégagent quelques conclusions relatives à l'emploi du traducteur APB sur ordinateur 1620.

La vitesse d'exécution des calculs proprement dits est trois à cinq fois plus élevéé sur l'ordinateur 1620 que sur le Gamma tambour.

Cependant la lenteur d'impression de la machine à écrire utiliséé comme organe de sortie sur l'ordinateur 1620 rend illusoire cette augmentation de vitesse pour les programmes qui comportent un volume d'impression de résultats important par rapport au temps de calcul. Dans le cas du programme III par exemple, les opérations d'entrée et sortie occupent 29 minutes sur un total de 32 minutes d'exécution.

Dans le cas du programme IV, au contraire, les sorties de résultats sont peu nombreuses : impression des valeurs propres et des vecteurs propres correspondants. Le volume de calcul est important et comporte de nombreuses opérations sur des registres d'index. Le rapport des temps d'exécution est alors très favorable à l'ordinateur 1620.

Il sera intéressant d'utiliser le traducteur APB sur l'ordinateur 1620 d'une part dans le cadre de travaux pratiques où le problème du temps d'exécution se pose peu, d'autre part, pour une catégorie de programmes qui comprennent un nombre restreint d'impressions de résultats.