



Ordinateur **IBM 1130**

Programas de Aplicação

- Civil Engineering Coordinate Geometry (COGO) for the IBM 1130, Model II
- Structural Engineering System Solver (STRESS) for the IBM 1130, Model 2B
- 1130 Project Control System
- 1130 Program for Optical System Design (POSD)
- 1130 Programs for Optical Systems Design (1130-EO-11X and 1130-EO-12X)
- 1130 Commercial Subroutine Package
- 1130 Commercial Subroutine Package (1130-SE-25X)
- 1130 Commercial Subroutine Package, Version 2
- 1130 Scientific Subroutine Package (1130-CM-02X)
- 1130 Route Accounting for Dairies and Bakeries (1130-DX-01X)
- 1130 Route Accounting System for Dairies and Bakeries
- 1130 Continuous System Modeling Program (1130-CX-13X)
- 1130 Statistical System (1130-CA-06X)
- 1130 Linear Programming Mathematical Optimization Subroutine System
- 1130 Mechanism Design System- Gears and Springs
- 1130 Synthetic Seismogram Program (1130-MP-11X)
- 1130 Synthetic Seismogram Program (1130-MP-11X)

ix cen-  
u pour  
lement  
larges  
résente  
emploi,  
utilisa-  
onique  
avaux.

\*\*\*\*\*

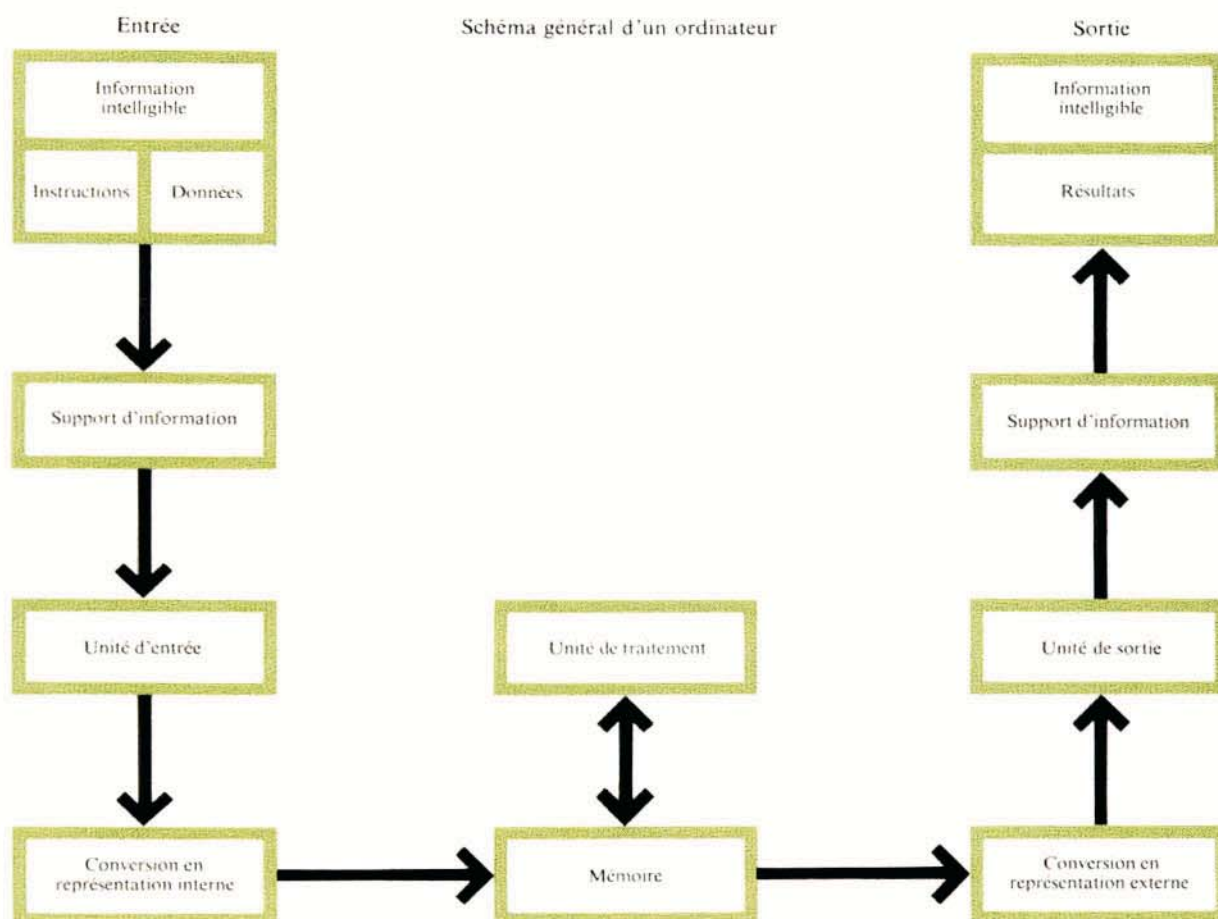
numérique  
présenta-

## IBM 1130

L'IBM 1130 est un ordinateur destiné aux bureaux d'étude, aux centres de recherche et aux établissements d'enseignement. Conçu pour résoudre les problèmes de statistique, de simulation, de dépouillement de mesures, de calcul scientifique et technique, il offre les plus larges possibilités d'application. D'un encombrement réduit, il ne présente aucune difficulté d'installation. Son coût modéré, sa facilité d'emploi, la simplicité de sa programmation le mettent à la portée d'utilisateurs qui jusqu'alors ne pouvaient disposer d'un matériel électronique adapté à la nature, au nombre et à l'importance de leurs travaux.

Quelques généralités sur le calcul numérique associé aux ordinateurs précéderont la présentation de l'ordinateur IBM 1130.

## organisation générale des ordinateurs





Lorsqu'un chercheur, un mathématicien ou un ingénieur se propose d'effectuer une série de calculs, il s'enquiert d'abord de la méthode à suivre et met noir sur blanc la suite des formules à utiliser. Il s'assure ensuite qu'il connaît toutes les hypothèses de départ, les coefficients numériques, les paramètres fixes et variables, les constantes, les variables, ainsi que l'ordre de grandeur des résultats cherchés.

Pour mener à bien son travail, il est amené à consulter des tables, des abaques, des réseaux de courbes dans lesquels il doit interpoler, extrapoler linéairement ou non. Il doit souvent résoudre des équations non linéaires par des méthodes graphiques, avoir accès aux tables de fonctions trigonométriques, hyperboliques, etc. Pour cela, il a besoin d'une règle à calcul, voire d'une machine à calculer pour améliorer la précision des calculs. La plupart du temps, il reprend son travail, les hypothèses de départ

devant être revues en fonction des résultats intermédiaires.

Pour résoudre un problème sur ordinateur, l'utilisateur décompose la suite des calculs en opérations élémentaires et affecte à chacune d'elles une instruction que l'ordinateur devra exécuter. Une suite d'instructions s'appelle un programme. Ce programme est enregistré dans la mémoire au moyen des unités d'entrée. Il peut être exécuté un nombre indéfini de fois sous le contrôle de l'unité de traitement. Il élabore les résultats et les restitue par l'intermédiaire des unités de sortie, le plus souvent une imprimante.

Une fois le programme établi, l'utilisateur peut recommencer les calculs autant de fois qu'il le désire en modifiant un ou plusieurs des paramètres; la partie la plus fastidieuse de son travail étant éliminée, il peut travailler avec le maximum d'efficacité.

## memoire

La mémoire centrale de l'ordinateur est généralement constituée de ferrites (élément magnétique capable d'enregistrer une information binaire 0 ou 1). Les ferrites groupées par caractère ou par mot (ensemble de caractères) emmagasinent les informations (données, tables, instructions, résultats intermédiaires, etc.). Chacun de ces caractères ou mots est identifiable individuellement par un nombre appelé adresse. Au moyen de ces adresses numériques, l'ordinateur peut localiser les informations qui, ainsi, peuvent être utilisées à l'infini et au besoin être remplacées. La vitesse des calculs dépend essentiellement du temps d'accès à la mémoire qui correspond au temps nécessaire pour rendre disponible une information unitaire enregistrée dans la mémoire, à une autre unité de l'ordinateur. L'opération de transfert se fait sous le contrôle de l'unité de traitement.

## unite de traitement

L'unité de traitement est capable d'identifier chacune des instructions du programme préalablement stocké dans la mémoire et d'effectuer les opérations correspondantes. Elle additionne, soustrait, multiplie, divise et compare des nombres. Elle prend des décisions en fonction de résultats intermédiaires dont elle peut reconnaître le signe. Elle transmet ces informations vers d'autres unités de l'ordinateur.

Ces quelques fonctions élémentaires suffisent pour effectuer les calculs les plus complexes, lesquels sont toujours décomposables en opérations arithmétiques simples.

## unites d'entree-sortie

Les données sont introduites dans l'ordinateur au moyen de cartes perforées, de bande de papier, de bandes magnétiques, de disques magnétiques, ou manuellement au moyen des boutons du pupitre de commande et de contrôle. En fait la fonction d'entrée se traduit presque toujours par l'enregistrement des informations en mémoire. L'ordinateur peut produire les résultats sous les formes les plus diverses : alphabétique, numérique, graphique, lumineuse, sonore, sur les supports les plus variés : papier (machine à écrire, imprimante, table traçante), bande perforée, carte perforée, bande, disque et tambour magnétiques.

Le processus suivi par l'ordinateur pour résoudre un problème est très proche de la méthode manuelle traditionnellement employée par l'ingénieur. *Le passage de l'un à l'autre constitue la programmation.*

## utilisation et programmation des ordinateurs

Un ordinateur traite essentiellement des données et des instructions, celles-ci exploitant celles-là. Utiliser un ordinateur revient donc, après avoir défini le format des données et des résultats, à établir des programmes d'instructions, en suivant pas à pas toutes les phases du calcul et en se conformant aux règles de programmation.

### représentation des données

Elle fait l'objet d'une convention à laquelle sont soumises les différentes unités de l'ordinateur. Les exemples suivants supposent que l'élément unitaire de mémoire de l'ordinateur est un mot comportant huit positions décimales.

*Nombre en virgule fixe.*

Un nombre enregistré dans un mot est calé à droite comme dans une opération arithmétique ordinaire mais indépendamment de la position de la virgule: par exemple : 23,148 est représenté par

00023148

L'addition de deux nombres n'ayant pas le même nombre de chiffres après la virgule pose un problème de cadrage. L'opération ne peut s'effectuer que si les positions fictives des virgules coïncident :

23,148	(00023148)		00023148
+ 32,4	(00000324)	doit s'écrire	+00032400

Tout au long des calculs, il faut connaître exactement la position de la virgule. En conséquence, cette représentation n'est pratiquement utilisée que pour les nombres entiers.

*Nombre en virgule flottante.*

Pour ne pas tenir compte de la virgule et pour gagner de la place en mémoire (élimination des zéros significatifs), un moyen pratique de représenter les nombres, dont l'ordre de grandeur peut varier entre de larges limites, consiste à les définir par une mantisse comprise entre 1 et 0,1 multipliée par une caractéristique égale à une puissance de 10 : 0,000238 s'écrit  $0,238 \cdot 10^{-3}$  et est représenté dans un mot par

+2380-03

et  $-0,000000004763$  donne

-4763-09

Le nombre de chiffres significatifs dépendra de la dimension du mot de mémoire et du nombre de mots utilisés pour cette représentation.



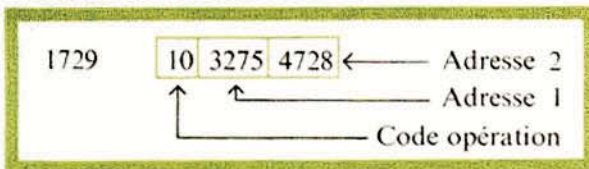
## représentation des instructions

Les exemples suivants supposent que la mémoire de l'ordinateur comporte 10 000 mémoires numérotées de 0000 à 9999, chacune d'elles comportant cette fois 10 positions décimales. Chaque instruction est repérée par une adresse exactement comme une donnée. Les instructions sont normalement exécutées les unes à la suite des autres, sauf dans le cas d'une instruction « de branchement ».

### Instruction arithmétique.

Une instruction de ce type doit contenir au moins deux types d'informations.

1. La nature de l'opération à effectuer (addition, multiplication, division...) : c'est le CODE OPERATION (par exemple 10 pour l'addition, 41 pour le branchement).
2. La ou les ADRESSES des facteurs à traiter.



Exemple : L'instruction contenue dans le mot 1729 de la mémoire, commande l'addition du nombre contenu dans le mot N° 3275 et du nombre contenu dans le mot N° 4728. Le résultat se trouve dans le mot N° 4728. Le nombre contenu dans le mot N° 3275 n'est pas effacé.

### Instruction logique ou « de branchement ».

Elle se représente de la même façon.

Exemple : 1221    41 3256 1729

L'instruction située en 1221 indique à l'ordinateur que la prochaine instruction à exécuter est située en 1729 à condition que le nombre contenu en 3256 soit nul. S'il ne l'est pas, l'ordinateur exécute l'instruction suivante située en 1222.

### Instruction d'entrée et de sortie.

Elle précise s'il s'agit d'une opération de lecture ou d'écriture, quel en est le mode, quelle est l'unité qui doit fonctionner (imprimante, bande magnétique, lecteur de cartes), etc.

## exécution d'une instruction

L'unité de traitement interprète chacune des instructions d'un programme enregistré dans la mémoire. Elle examine tout d'abord le code opération et lui fait correspondre une séquence de commandes de circuits déterminés. Elle analyse chacune des adresses, sachant en fonction du code opération, qu'elles représentent, par exemple, des facteurs arithmétiques. Enfin, elle exécute l'instruction et enregistre les résultats en mémoire.

## système de programmation

Il est bien évident que la conception, l'écriture et la mise au point d'un programme en un langage assimilable par la machine où chaque instruction est rédigée dans la représentation propre à l'ordinateur, sont des travaux fastidieux qui alourdissent l'utilisation de l'ordinateur.

Dans les bureaux d'étude et les centres de recherche, où la grande partie des calculs ne se présente qu'une fois, il faut un moyen rapide et pratique de transposer le problème en programme. De très nombreux systèmes de programmation ont été conçus dans ce sens. L'un d'entre eux, le système FORTRAN, est particulièrement adapté au langage mathématique.



## **le système FORTRAN**

Le système de programmation FORTRAN (FORmulation TRANsposée) est une méthode de programmation efficace. Il permet de transposer et de résoudre sans difficulté les problèmes scientifiques et techniques. Il transforme en quelque sorte les caractéristiques de l'ordinateur en simulant automatiquement les fonctions mathématiques.

La résolution numérique d'un calcul complexe revient à récrire, suivant une notation très proche de la notation mathématique usuelle, l'ensemble des relations et formules à utiliser. Une journée suffit à un utilisateur pour se familiariser avec la programmation.

Une fois le programme rédigé en langage FORTRAN, il est nécessaire de le transformer en un programme ne comprenant que les instructions élémentaires assimilables par l'ordinateur; c'est le rôle du traducteur. Ce travail est assuré par l'ordinateur lui-même, à l'aide de programmes de traduction fournis par IBM.

## le langage

Le langage FORTRAN est un ensemble de mots (vocabulaire) et de règles (grammaire) permettant de rédiger un texte qui reflète la suite des calculs à effectuer pour résoudre un problème. Dans la pratique, il est raisonnable de commencer par décrire l'organisation logique du problème au moyen d'un ORGANIGRAMME. Il suffit ensuite de transposer les formules mathématiques en instructions FORTRAN, dont l'ensemble constitue un programme appelé PROGRAMME ORIGINE. Comme on le verra dans l'exemple détaillé ci-après, les modifications imposées par la transposition sont simples.

Le vocabulaire international FORTRAN comprend une trentaine de mots clés qui correspondent à des ordres que le programmeur peut donner à la machine; de plus, pour désigner les quantités sur lesquelles portent ces ordres, le programmeur utilise des symboles qu'il définit lui-même.

Les règles essentielles de la grammaire sont les suivantes :

- Les signes + et - conservent leur représentation habituelle.
- Le signe de multiplication est représenté par un astérisque, le signe de division par une barre inclinée :  $(A + B) D / (D - E) F$   
s'écrit  $(A + B) * D / (D - E) * F$ .
- L'élevation à une puissance est indiquée à l'aide de deux astérisques juxtaposés suivis de l'ordre de la puissance  $(A - B)^2$   
s'écrit  $(A - B) ** 2$ .
- Les symboles SQRT, SIN, COS, EXP, LN, ABS, etc., remplacent les opérations racine carrée ( $\sqrt{\quad}$ ), sinus, cosinus, exponentielle, logarithme népérien, valeur absolue ( $|\quad|$ ). En fait ces fonctions sont réalisées au moyen de sous-programmes incorporés au traducteur.
- L'argument de la fonction peut être une expression arithmétique et apparaît entre parenthèses à droite du symbole, par exemple :  
 $\sqrt{B^2 - 4AC}$  s'écrit  $SQRT (B**2 - 4.*A.*C)$ .
- D'une manière générale les expressions qui doivent être évaluées comme un tout sont mises entre parenthèses.

Telles sont les règles essentielles de la formulation mathématique FORTRAN. Elles permettent de comprendre sans difficulté l'exemple qui suit.

### exemple

*Résolution d'un système d'équations non linéaires au moyen de FORTRAN.*

Le système proposé est le suivant :

$$G(x, y) = \sin x - \sin y - 0,2 = 0$$

$$H(x, y) = \cos x - \cos y - 1,2 = 0$$

L'objectif est de rechercher la solution la plus proche de zéro.

On calcule successivement :

$$G_x = -H_y = \frac{\delta G}{\delta x} = \frac{\delta H}{\delta y} = \cos x - \sin y$$

$$G_y = -H_x = \frac{\delta G}{\delta y} = \frac{\delta H}{\delta x} = \sin x - \cos y$$

$$\Delta = G_x H_y - G_y H_x = G_x^2 + G_y^2$$

La solution est obtenue par itérations successives au moyen des relations suivantes :

$$X_{k+1} = X_k + \frac{1}{\Delta} (HG_y - GH_y)$$

$$Y_{k+1} = Y_k + \frac{1}{\Delta} (GH_x - HG_x)$$

La solution est acceptable quand on a à la fois :

$$G(x, y) < 10^{-7} \text{ et } H(x, y) < 10^{-7}$$

On prendra comme valeurs de départ :

$$X_0 = 0,1 \quad Y_0 = 0,1$$

Afin de respecter les règles d'écriture du langage FORTRAN nous utiliserons les notations symboliques suivantes :

EXP (Y) pour  $e^y$       EXP (-Y) pour  $e^{-y}$

SIN (X) pour  $\sin x$       COS (X) pour  $\cos x$

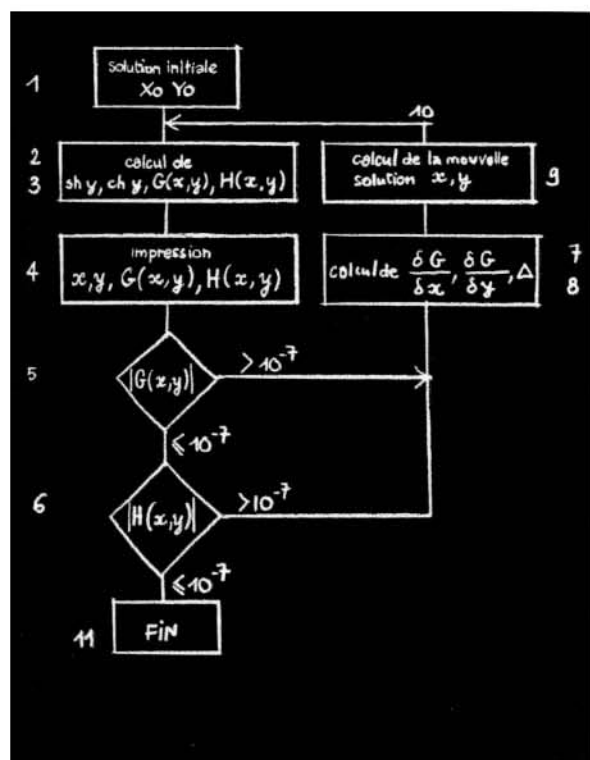
Ce sont des fonctions connues du traducteur FORTRAN.



Par contre les notations suivantes ne sont que des symboles mnémoriques équivalents aux variables mathématiques :

SHY pour $s_{hy}$	CHY pour $c_{hy}$
GXY pour $G(x,y)$	HXY pour $H(x,y)$
GX pour $G_x$	GY pour $G_y$
X pour $x$	Y pour $y$
DET pour $\Delta$	

A l'aide de ces notations le programme reproduit fidèlement le langage mathématique; il peut être lu et compris par une personne autre que l'auteur.



### Logique générale.

Les instructions numérotées 1 ne sont exécutées qu'une fois; elles fixent les valeurs de départ des variables  $x$  et  $y$ . Les instructions 2, 3, 4, 7, 8, 9 sont exécutées autant de fois qu'il faut jusqu'à la convergence du processus. Elles permettent de calculer une solution  $(x_{k+1}, y_{k+1})$  à partir de la solution précédente  $(x_k, y_k)$ . Les instructions 5 et 6 contrôlent le déroulement de la suite des itérations et permettent d'identifier la solution acceptable.

## Programme

Probleme		Date	Page
Type	N°	ORDRE FORTRAN	
1		X=0.1	
		Y=0.1	
2		SHY=EXP(Y)-EXP(-Y))/2.	
		CHY=EXP(Y)+EXP(-Y))/2.	
3		GXY=SIN(X)*SHY-0.2	
		HXY=COS(X)*CHY-1.2	
4		WRITE(1,100)X,Y,GXY,HXY	
5		IF (ABS(GXY)-0.000001) 6,6,7	
6		IF (ABS(HXY)-0.000001) 11,11,7	
7		GX=COS(X)*SHY	
		GY=SIN(X)*CHY	
8		DET=GX**2+GY**2	
9		X=X+(HXY*GY-GXY*GX)/DET	
		Y=Y-(GXY*GY+HXY*GX)/DET	
10		GO TO 2	
11		STOP	
100		FORMAT (4E16.8)	
		END	

## Résultats

x	y	G(x,y)	H(x,y)
.10000000E+00	.10000000E+00	-.19000002E+00	-.20001680E+00
.43416440E-01	.20499023E+01	-.34236140E-01	.27442244E+01
.84604611E-01	.13325400E+01	-.50983920E-01	.81998170E+00
.15841550E+00	.87307350E+00	-.44092580E-01	.18835720E+00
.24308641E+00	.69931198E+00	-.17617160E-01	.17760200E-01
.27207005E+00	.68706447E+00	-.49508000E-03	-.35120000E-03
.27244941E+00	.68773314E+00	.33000000E-06	.10000000E-06
.27244909E+00	.68773285E+00	.00000000E-99	.00000000E-99

x = 0,27244909  
y = 0,68773285

La solution est donc :

### Commentaires

Généralement on ne numérote que les instructions auxquelles on se réfère dans le programme (2, 6, 7, 11, 100). Des numéros ont été ajoutés pour faciliter la compréhension du programme. Les instructions 2 explicitent en fonction de  $e^x$  les fonctions hyperboliques SHY et CHY qui en général ne sont pas connues du traducteur FORTRAN.

Les instructions 3 permettent de calculer les deux fonctions  $G(x, y)$  et  $H(x, y)$  qui constituent le système.

L'instruction 4 provoque l'impression, par exemple sur machine à écrire (unité numérotée 1), des valeurs intermédiaires de la dernière solution obtenue  $(x, y)$  et des valeurs des fonctions  $G(x, y)$  et  $H(x, y)$ . Chacune des quatre valeurs sera imprimée selon le format défini à la ligne numérotée 100.

La ligne numérotée 100 définit la présentation des résultats imprimés. Le chiffre 4 indique que les quatre valeurs de la ligne 4 seront imprimées de la même façon. E 16.8 veut dire que le nombre sera représenté en virgule flottante, qu'il occupera 16 positions et qu'il comportera 8 chiffres significatifs.

L'instruction 5 examine le signe de l'expression située entre parenthèses immédiatement après le mot IF.

Les trois lettres ABS désignent la fonction valeur absolue de la quantité entre parenthèses qui suit. Si l'expression  $(ABS(GXY) - 0.0000001)$  est négative ou nulle, le programme exécute l'instruction 6, si elle est positive l'instruction 7.

Si l'instruction 6 remplit les conditions de convergence, le programme exécute donc l'instruction 7 qui examine la valeur absolue de la fonction  $H(x, y)$ . Si la condition  $|H(x, y)| \leq 10^{-7}$  est remplie, le programme exécute l'instruction 11 et le calcul s'arrête.

Si la condition n'est pas remplie, les instructions 7, 8, 9 déterminent une nouvelle solution  $(x, y)$  au départ d'une nouvelle itération commençant en 2.

De nombreuses possibilités offertes par FORTRAN ne figurent pas dans ce programme, telle la possibilité d'affecter aux variables jusqu'à trois indices pouvant varier indépendamment les uns des autres; la progression de ces indices peut être placée sous le contrôle d'un ordre spécial DO qui permet en outre de répéter plusieurs fois un groupe d'instructions. Le traitement des tableaux de nombres à deux ou trois dimensions se trouve ainsi simplifié. Des exemples d'utilisation de cette forme d'instruction figurent dans le chapitre suivant.

Exemple :

la valeur  $X = 0,27244909$  sera imprimée :

bbb. 27244909 E + 00

Puissance de 10  
8 chiffres significatifs  
Position blanche si le nombre est positif  
signe - si le nombre est négatif  
Deux blancs d'espacement entre nombres

16 positions d'impression



## le traducteur

Le PROGRAMME ORIGINE est traduit par l'ordinateur lui-même sous le contrôle d'un programme de traduction appelé traducteur ou COMPILATEUR. Il est spécialement conçu pour comprendre le langage FORTRAN et le transformer dans le langage de la machine à raison de plusieurs dizaines d'instructions machine pour une instruction FORTRAN. Le programme obtenu est appelé PROGRAMME RÉSULTANT. Il suffit de l'enregistrer en mémoire et de le faire exécuter par l'ordinateur pour obtenir les résultats cherchés.

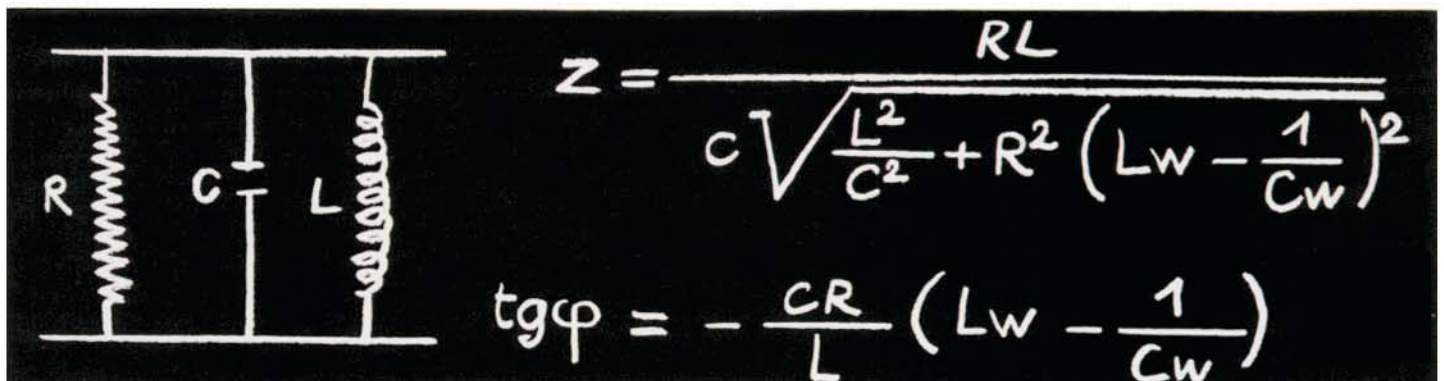
Au cours de la traduction, le compilateur décèle les erreurs formelles du programme origine et en fournit la liste; la mise au point du programme en est accélérée.

Dans la pratique chaque ligne du programme origine est perforée dans une carte. Le programme traducteur est alors chargé en mémoire et exécuté. Il lit les cartes du programme origine et restitue le programme résultant soit sur cartes perforées, soit sur bande perforée, soit sur disque magnétique. Il suffit de charger en mémoire ce programme résultant pour qu'il soit exécuté.

## exemple de calculs numériques sur ordinateur 1130

### calcul des éléments d'un circuit électrique élémentaire

Il s'agit de calculer l'impédance et le déphasage d'un circuit bouchon RLC en fonction de la fréquence (100 à 20 000 Hz) et pour plusieurs groupes de valeurs RLC.



### Programme

Probleme		Date	Page
Type	Suite	ORDRE FORTRAN	
N°		Identification	
1		<pre> FREQ=0. READ(2,100)RES,SELF,CAPA WRITE(3,100)RES,SELF,CAPA DO 2 I=1,200 FREQ=FREQ+100 PULS=2.*3.14159*FREQ PART=SELF*PULS-1./((CAPA*PULS) FMP=RES*SELF/(CAPA*SQRT(SELF**2/(CAPA**2+RES**2*PART**2)) DEPH=-CAPA*RES*PART/SELF 2 WRITE(3,100)FREQ,FMP,DEPH GO TO 1 100 FORMAT(3E16.8) END </pre>	

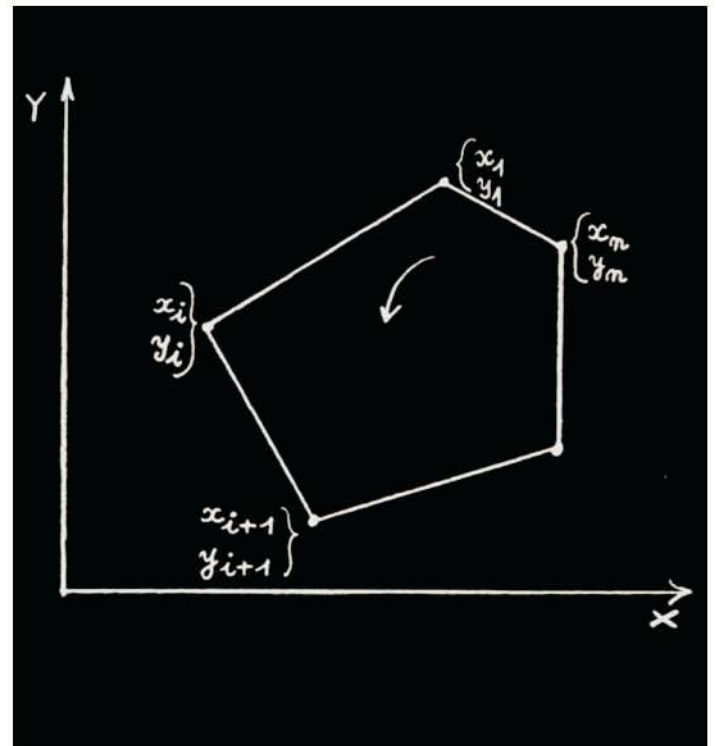
### Commentaires

L'instruction DO 2 I = 1,200 permet d'effectuer 200 fois la suite des calculs compris entre elle et l'instruction numérotée 2 avec impression des résultats pour chaque valeur de la fréquence.

Puis le programme, au moyen de l'instruction GO TO 1 provoque la lecture de nouveaux éléments RES, SELF, CAPA, réinitialise la valeur de la fréquence et ainsi de suite.

## calcul des éléments d'une section de poutre

Cet exemple n'est qu'une partie d'un programme plus important utilisé en mécanique, en génie civil pour calculer la surface, les moments statiques et les moments d'inertie d'une aire polygonale à partir des coordonnées de chacun des sommets suivant deux axes rectangulaires.



### Méthode

Le contour polygonal est décomposé en trapèzes; chaque trapèze est calculé et les résultats sont totalisés. Le calcul utilise les formules suivantes :

Surface 
$$F = \frac{1}{2} \sum_{i=1}^n (y_i + y_{i+1}) (x_i - x_{i+1})$$

Moment statique par rapport à oy 
$$S_y = \frac{1}{6} \sum_{i=1}^n (x_i - x_{i+1}) \left[ (x_i + 2x_{i+1}) y_{i+1} + (2x_i + x_{i+1}) y_i \right]$$

Moment statique par rapport à ox 
$$S_x = \frac{1}{6} \sum_{i=1}^n (x_i - x_{i+1}) (y_{i+1}^2 + y_i (y_i + y_{i+1}))$$

Moment d'inertie par rapport à ox 
$$I_x = \frac{1}{12} \sum_{i=1}^n (x_i - x_{i+1}) (y_i + y_{i+1}) (y_i^2 + y_{i+1}^2)$$

Moment d'inertie par rapport à oy 
$$I_y = \frac{1}{12} \sum_{i=1}^n (y_{i+1} - y_i) (x_i + x_{i+1}) (x_i^2 + x_{i+1}^2)$$



## Programme

Probleme		Date	Page
Type	Suite	ORDRE FORTRAN	
N°		Identification	
		SUBROUTINE POLYG (N,X,Y)	
		DIMENSION X (99), Y (99)	
		COMMON F, SX, SY, XI, YI	
		F=0.	
		SX=0.	
		SY=0.	
		XI=0.	
		YI=0.	
		DO 4 I=1, N	
		IF(I-N) 1,2,2	
1		J=I+1	
		GO TO 3	
2		J=1	
3		F=F+(Y(I)+Y(J))*(X(I)-X(J))/2.	
		SX=SX+(Y(I)**2+Y(I)*Y(J)+Y(J)**2)*(X(I)-X(J))/6.	
		SY=SY+((X(I)+2.*X(J))*Y(J)+Y(I)*(2.*X(I)+X(J)))*(X(I)-X(J))/6.	
		XI=XI+(Y(I)**2+Y(J)**2)*(Y(I)+Y(J))*(X(I)-X(J))/12.	
4		YI=YI+(X(I)**2+X(J)**2)*(X(I)+X(J))*(Y(J)-Y(I))/12.	
		SIGN=F/ABS(F)	
		F=ABS(F)	
		SX=SX*SIGN	
		SY=SY*SIGN	
		XI=ABS(XI)	
		YI=ABS(YI)	
		RETURN	
		END	

### *Commentaires*

L'instruction **SUBROUTINE** indique le début du sous-programme ; **POLYG (N, X, Y)** indique son nom et les trois variables indispensables à son exécution. Dans le programme principal qui fait appel de nombreuses fois à ce même sous-programme au moyen de l'instruction **CALL**, on utilise la même référence.

L'instruction **RETURN** marque la fin du sous-programme et prépare le retour au programme principal.

L'instruction **DIMENSION** indique les dimensions maxima des emplacements de mémoire réservés aux coordonnées  $(x_i, y_i)$  de chacun des sommets du polygone.

L'instruction **COMMON** assigne aux résultats du sous-programme les mêmes adresses que dans le programme principal.

L'instruction **DO 4 I = 1, N** permet d'effectuer les sommations  $F = \Sigma, SX = \Sigma, SY = \Sigma, XI = \Sigma, YI = \Sigma$  en fonction du nombre **N** de sommets du polygone.

Les six dernières instructions rétablissent les signes des différents résultats (surface **F** positive, moments d'inertie **XI** et **YI** positifs, etc.) qui dépendent du sens de numérotation des sommets.

## calcul d'écart-type

Le calcul d'écart-type porte sur N valeurs de la grandeur physique X.

L'écart-type est donné par la formule ci-contre :

$$\sigma = \sqrt{\frac{\sum x^2 - \frac{(\sum x)^2}{N}}{N-1}}$$

## Programme

Probleme		Date	Page	
Type	N°	ORDRE FORTRAN		Identification
			READ(2,100)N	
			SIGMA=0.	
			CARRE=0.	
			C=N	
			DO 1 I=1,N	
			READ(2,200)X	
			SIGMA=SIGMA+X	
	1		CARRE=CARRE+X**2	
			ECART=SQRT((CARRE-SIGMA**2/C)/(C-1.))	
			WRITE(3,300)ECART	
			PAUSE	
	100		FORMAT(I3)	
	200		FORMAT(F5.2)	
	300		FORMAT(7H ECART=,E15.8)	
			END	



### *Commentaires*

Le nombre de valeurs  $N$  est contenu dans la première carte suivant le FORMAT 100 (nombre entier perforé dans les trois premières colonnes de la carte).

Chaque valeur de  $X$  est contenue dans une carte suivant le FORMAT 200 (5 caractères en tout dont un signe en première colonne et la virgule en troisième colonne).

La boucle dont l'exécution est contrôlée par l'instruction DO effectue la lecture de  $N$  cartes et les cumuls correspondants  $\Sigma X$  et  $\Sigma (X^2)$  nécessaires au calcul de l'écart-type.

La valeur de l'écart est imprimée suivant le FORMAT 300 (15 caractères en tout dont 8 chiffres significatifs, 2 chiffres pour l'exposant et 1 signe pour l'exposant).

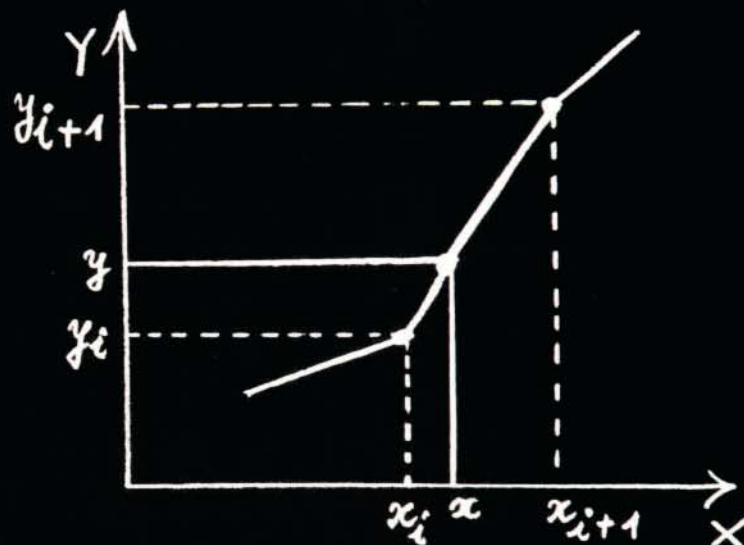
## consultation de table avec interpolation linéaire

La consultation de table est une technique de calcul numérique très utilisée dans tous les bureaux d'étude. Dans la mesure où l'intervalle de la table est petit, l'interpolation linéaire est presque toujours employée.

Le problème consiste à déterminer pour une valeur  $x$  de la variable, la valeur  $y$  de la fonction correspondante.

Si  $x_{i+1}$  et  $x_i$  encadrent la valeur  $x$ , la valeur de  $y$  est donnée par la formule ci-dessous :

$$y = y_i + \frac{(y_{i+1} - y_i)(x - x_i)}{x_{i+1} - x_i}$$



## Programme

Probleme		Date	Page
Type	№	ORDRE FORTRAN	
↓	↓	Identification	
		DIMENSION X(12),Y(12)	
1		READ(2,100)VAR	
		DO 6 I=1,12	
		IF(VAR-X(I))2,3,6	
6		CONTINUE	
		GO TO 1	
2		FONCT=Y(I-1)+(Y(I)-Y(I-1))*(VAR-X(I-1))/(X(I)-X(I-1))	
4		WRITE(3,200)FONCT	
		GO TO 1	
3		FONCT=Y(I)	
		GO TO 4	
100		FORMAT	
200		FORMAT	
		END	

## Commentaires

L'instruction DIMENSION suppose qu'une table de 24 nombres (12 pour  $X_1, X_2, \dots, X_{12}$  et 12 pour  $Y_1, Y_2, \dots, Y_{12}$ ) est enregistrée en mémoire avant l'exécution de ce programme.

La boucle initialisée par l'instruction DO cherche la valeur  $X(I)$  immédiatement supérieure à l'argument VAR. Si au bout de 12 recherches aucune valeur n'est trouvée, le programme lit sur une carte une nouvelle valeur de VAR pour une nouvelle consultation de la table.

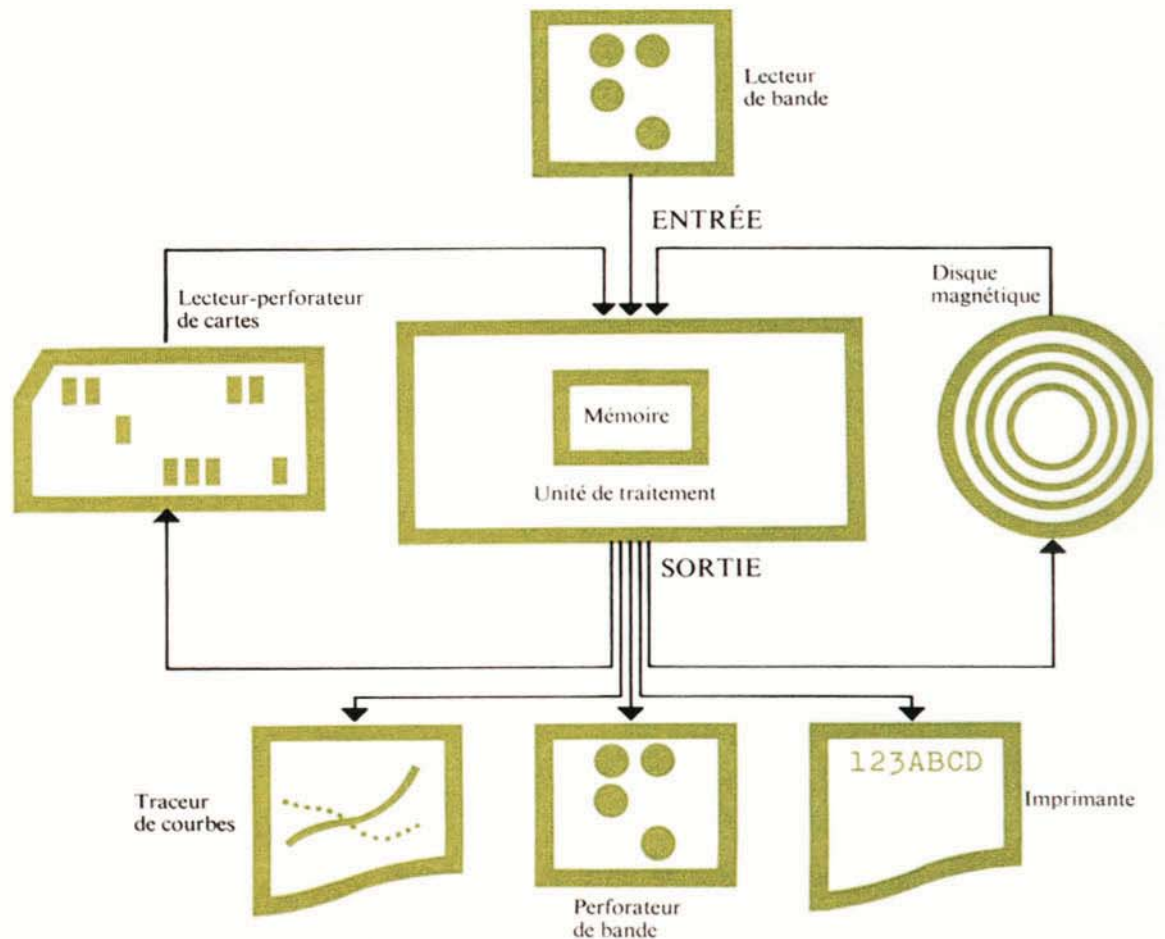
Dans les autres cas, après calcul avec ou sans interpolation, le résultat FONCT est imprimé. En fait ce programme type est rarement utilisé sous cette forme. La valeur VAR n'est pas lue sur une carte mais est en général le résultat d'un calcul intermédiaire au sein d'un programme principal important.

Les instructions FORMAT n'ont pas été détaillées. Elles dépendent essentiellement de la nature et de l'ordre de grandeur des nombres traités.



# ordinateur 1130

L'organisation de l'ordinateur 1130 ressemble celle de tout ordinateur. Il comprend une unité de traitement renfermant la mémoire et, connectées à elle, différentes unités d'entrée et de sortie. Ces unités peuvent faire l'objet d'un choix déterminé par la nature des problèmes à traiter et s'appliquer à plusieurs combinaisons. Elles sont sous le contrôle de l'unité de traitement qui, à tout instant, peut connaître leur état, savoir si elles sont occupées ou disponibles.





*Un système complet comprenant l'unité de traitement et toutes les unités périphériques d'entrée et de sortie dissipe 2 050 KCAL/heure, exige 2,9 KVA en monophasé, pèse 740 kg et peut être installé dans un local d'une superficie inférieure à 25 m<sup>2</sup>. La mise en place et l'utilisation de l'ordinateur 1130 ne nécessitent pas d'installations spéciales.*

## unité de traitement

L'unité de traitement dispose de 3 index (registres permettant de modifier automatiquement une adresse), de l'adressage indirect qui étend le rôle de la partie adresse d'une instruction et de 6 niveaux d'interruption qui permettent de faire fonctionner plusieurs unités d'entrée et de sortie en SIMULTANÉITÉ. La mémoire de l'unité de traitement est constituée de ferrites. Elles sont groupées par 18 : 16 ferrites utiles et 2 ferrites de contrôle. L'élément unitaire de mémoire est donc un MOT de 16 positions binaires. Le temps d'accès à un mot est de 3,6 microsecondes. La capacité de la mémoire est de 4096 ou 8192 mots

adressables. Les opérations arithmétiques (multiplication et division sont câblées) permettent de traiter les nombres en simple ou double précision. L'unité de traitement est munie également d'un pupitre avec indicateurs visuels et boutons de commande ainsi que d'un clavier à l'usage de l'opérateur, pour effectuer toute opération de contrôle ou entrée directe d'informations en mémoire.

Enfin, une imprimante de pupitre permet l'impression de messages ou de résultats de calculs à la cadence de 15,6 caractères par seconde.



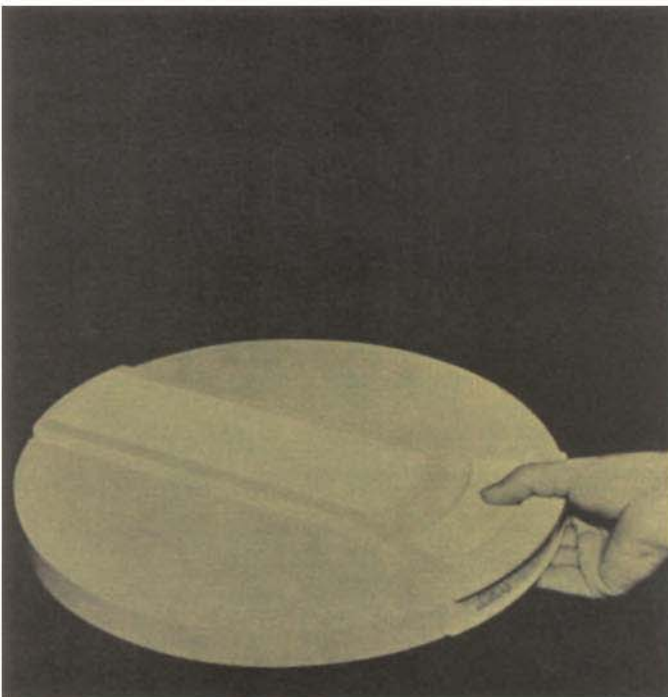


### mémoire à disque magnétique amovible

Elle peut être intégrée physiquement à l'unité de traitement. C'est une mémoire à accès sélectif constituée par un disque AMOVIBLE qui donne la possibilité à l'utilisateur d'organiser une « discothèque » de capacité pratiquement illimitée.

Un disque renferme 512 000 mots répartis sur deux faces en 400 pistes de 4 secteurs de 320 mots. Ce disque est accessible par un mécanisme comportant des têtes de lecture et d'écriture. Le mécanisme se déplace d'une piste ou de deux pistes en 15 millisecondes et, une fois positionné, permet l'accès à 2 560 mots.

Le transfert des données entre l'unité de disque et l'unité de traitement est placé sous le contrôle du système d'interruption automatique et s'effectue par l'intermédiaire d'un canal à raison de 35 000 mots à la seconde.



### programme moniteur

Le disque magnétique permet d'enregistrer le programme assembleur, le traducteur FORTRAN, les sous-programmes, les programmes utilitaires, les programmes de l'utilisateur, des tableaux de données, des fichiers, etc.

Le MONITEUR, également enregistré sur disque, assure la gestion automatique de l'ensemble de ces programmes et simplifie l'exploitation de l'ordinateur en éliminant tous les temps morts au cours des procédures opératoires.

Il charge directement en mémoire, à partir du disque, le programme approprié au moment de son utilisation.

Il enchaîne automatiquement plusieurs travaux sans aucune intervention manuelle de l'opérateur.

La mise au point des programmes est accélérée.

Les manipulations et la consommation de cartes perforées et de ruban perforé sont considérablement réduites.

Le MONITEUR rentabilise au mieux l'ordinateur 1130.

### traceur de courbes

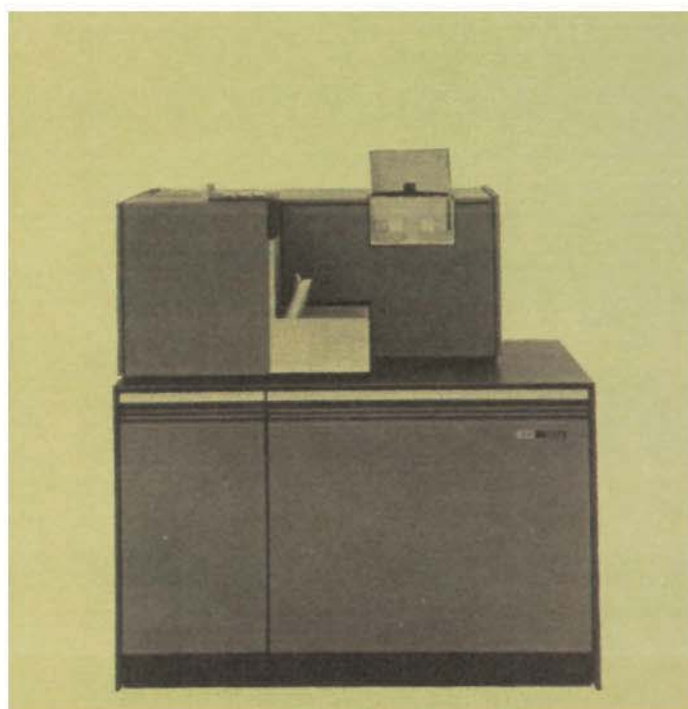
Il permet d'obtenir, sous forme graphique, les informations reçues de l'unité de traitement. Il se compose d'une plume se déplaçant latéralement sur un support (axe des y) et d'un tambour assurant le déplacement longitudinal du papier sous la plume (axe des x).

Le tracé est réalisé par une suite de déplacements élémentaires de la plume, suivant huit directions possibles (un déplacement élémentaire suivant l'axe des x ou des y a une longueur de 0,254 mm) à la cadence de 200 à 300 par seconde en fonction du modèle de traceur.



### lecteur-perforateur de cartes

Suivant le modèle, il lit 300 ou 400 cartes en 1 minute et perfore 80 ou 160 colonnes par seconde. Les informations lues ou perforées peuvent être exprimées à l'aide du code-carte standard IBM ou de tout autre code. Le lecteur-perforateur est muni d'un système automatique contrôlant l'exactitude de l'information lue ou perforée.



### perforateur de bande lecteur de bande perforée

Ces unités perforent et lisent à la vitesse de 888 caractères par minute. La largeur de la bande, qui peut être en papier ou en Mylar, est de 2,54 cm pour 8 canaux de perforation. Les informations lues ou perforées peuvent être exprimées à l'aide du code-bande perforée standard IBM ou de tout autre code.

### imprimante

C'est une imprimante à roues. Sur chaque roue sont disposés 48 caractères : 10 chiffres, 26 lettres et 12 caractères spéciaux. Sa vitesse d'impression à la minute est de 82 lignes en écriture alphabétique, de 110 lignes en écriture numérique, la ligne comprenant 120 caractères.

L'entraînement du papier est contrôlé au moyen d'une bande pilote.





## les programmes

IBM produit une BIBLIOTHÈQUE DE PROGRAMMES pour l'ordinateur 1130. Elle se compose de deux parties : l'une destinée à faciliter le travail de l'utilisateur dans l'écriture, la mise au point et l'exploitation de ses propres programmes, l'autre lui fournissant les solutions programmées à certains types d'applications.

### *Les systèmes de programmation :*

- Le programme ASSEMBLEUR permet l'écriture de programmes en un langage symbolique, souple et parlant, plus facile à utiliser que le langage machine.
- Le traducteur FORTRAN accepte un langage très proche de celui des mathématiques et est avant tout destiné aux calculs scientifiques et techniques.
- Le MONITEUR gère les programmes enregistrés sur disque magnétique et automatise l'exploitation de la machine.
- L'ensemble de sous-programmes assurant le contrôle des opérations d'entrée et de sortie, l'exécution de fonctions telles que SINUS, COSINUS, LOGARITHME... est utilisé par le programme ASSEMBLEUR et le traducteur FORTRAN.
- Les programmes UTILITAIRES chargent les programmes résultants, permettent l'analyse de la mémoire pendant les procédures de mise au point, assurent le transfert de l'information d'un support à un autre (carte à bande perforée, par exemple).

### *Les programmes d'applications :*

Dans tout programme écrit en FORTRAN, l'utilisateur peut appeler un ou plusieurs sous-programmes faisant partie d'un ensemble appelé MATHPAK. Ces sous-programmes utilisent le langage FORTRAN et donnent les solutions programmées de certaines fonctions et techniques mathématiques communément employées par les chercheurs et ingénieurs. Ils traitent des sujets suivants : Fonction de Bessel, Série de Fourier, Équations Différentielles, Intégration numérique, Calcul Matriciel, Polynôme de Legendre, Génération de nombres au hasard...

Les programmes d'applications comprennent également :

- un ensemble généralisé de programmes de statistiques
  - analyse de variance,
  - régression linéaire,
  - lissage polynomial de courbe,
  - analyse factorielle;
- des programmes d'études de surfaces utilisant le traceur de courbes;
- un programme COGO destiné aux applications de Génie Civil;
- etc...

**IBM**

Compagnie IBM France  
Division machines à cartes perforées et ordinateurs  
5, place Vendôme - Paris 1<sup>er</sup> - Tél. 073.17.90

