DIRECTION DES SERVICES D'ENSEIGNEMENT

ÉCOLE NATIONALE SUPÉRIEURE
DES TÉLÉCOMMUNICATIONS

INITIATION A L'EMPLOI DES CALCULATEURS ELECTRONIQUES

par J.THURIN
Ingénieur en Chef des Télécommunications

F.MONTEILLER
Inspecteur Principal

R.VOLT
Inspecteur Principal Adjoint
des Services d'Enseignement

1967

ÉCOLE NATIONALE SUPÉRIEURE
DES TÉLÉCOMMUNICATIONS

INITIATION A L'EMPLOI DES CALCULATEURS ELECTRONIQUES

par J.THURIN
Ingénieur en Chef des Télécommunications

F.MONTEILLER
Inspecteur Principal

R.VOLT
Inspecteur Principal Adjoint
des Services d'Enseignement

Imprimé et Edité par la
DIRECTION DES SERVICES D'ENSEIGNEMENT
46, Rue Barrault - PARIS 13ème

C.C.P. 9041-84 PARIS

Tél. 707-89-59

PREMIERE PARTIE

LES CALCULATEURS ELECTRONIQUES

Chapitre I

LE CALCUL SCIENTIFIQUE

1.1,1. UTILISATION DES NOMBRES POUR CARACTERISER LES PHENOMENES PHYSIQUES

L'étude scientifique des phénomènes naturels est basée sur les éléments suivants :

1° - Observation des phénomènes et choix d'une ou plusieurs grandeurs qui les caractérisent ;

2° - Etablissement d'une correspondance entre une suite d'états de la grandeur et une suite de nombres.

L'étude de l'état ou de l'évolution d'un phénomène physique se traduit par un ensemble de relations logiques entre les grandeurs qui le caractérisent, ou plus précisément par un ensemble de relations entre les nombres qui caractérisent l'état de ces grandeurs.

Dans certains cas, ces relations sont des inégalités et traduisent un classement. Généralement, ces relations sont métriques ; on définit l'addition algébrique de deux grandeurs et par suite les quatre opérations.

Dans ce qui suit, nous nous intéresserons plus particulièrement au traitement mathématique des nombres qui jouissent de la propriété d'additivité.

1.1,2. MATHEMATIQUES ET CALCUL NUMERIQUE

Les lois physiques qui permettent d'analyser et de prévoir l'évolution d'un phénomène traduisent des relations entre grandeurs. La formulation mathématique d'une loi physique consiste à établir des relations entre les nombres qui caractérisent l'état d'un phénomène.

Pour donner aux raisonnements qui interviennent dans le traitement mathé-matique des grandeurs, un caractère de généralité, le mathématicien raisonne sur de symboles. Cela lui permet par ailleurs de condenser son écriture puisqu'il n'a pas à expliciter l'ensemble des nombres qui caractérisent un état. Mais, si le but n'es pas une simple spéculation de l'esprit mais l'utilisation concrète des lois de la physique en vue d'une action, il est nécessaire en fin de compte de revenir aux val numériques. Cette dernière phase est souvent fastidieuse. Elle peut nécessiter un temps considérable et être entachée d'erreur du fait de l'imperfection des calculat humains. Le but des machines à calculer est essentiellement de déterminer rapidemen et sans fautes la suite des valeurs numériques de problèmes pour lesquels les mathé maticiens ont défini une méthode de résolution.

1.1.3. REPRESENTATION CONTINUE ET REPRESENTATION DISCRETE DE L'EVOLUTION D'UN PHENOMENE

Pour représenter l'évolution d'un phénomène faut-il considérer une suite continue ou une suite discrète de nombre ?

Pour tenter de répondre à cette question, il faut prendre en considératio un certain nombre de points.

A priori, deux états voisins d'un phénomène ne peuvent être discernés que si l'écart dépasse un certain seuil, que l'on se place au niveau de la particule élémentaire régie par le principe d'indétermination ou au niveau macroscopique pour lequel il existe un "bruit" c'est-à-dire une fluctuation aléatoire. L'ensemble des nombres caractérisant un état d'un phénomène est donc défini avec une certaine appr ximation. Puisqu'il y a approximation, le phénomène peut être décrit par un ensembl discret de nombres.

Pour les besoins l'analyse mathématique, il est utile de considérer les phénomènes comme continus de façon à pouvoir définir la notion de dérivée. Ceci n'altère pas la description des phénomènes physiques dont on ne connaît que des valeurs discrètes.

Lorsqu'on sort du formalisme mathématique et qu'on revient aux valeurs numériques, on retrouve pour décrire l'évolution escomptée d'un phénomène ou bien une suite continue de valeurs connues avec une certaine incertitude, ou bien une su discrète de valeurs. Finalement, le premier des éléments qui fixe la qualité du cal fait par une machine, est soit l'approximation lors d'une lecture ou d'une inscript soit le pas entre valeurs successives. Les qualités d'une machine à calculer ne son donc pas liées au fait qu'elles donnent des résultats sous forme continue ou sous f discrète.

Ce qui fixe la précision avec laquelle on peut prévoir l'évolution d'un phénomène, c'est d'une part la précision avec laquelle sont connues les données ini-tiales, d'autre part, la précision avec laquelle sont obtenus les calculs faits à partir de ces données.

Le fait qu'une petite erreur sur les données, ce qui modifie le problème, n'entraîne pas une variation très importante de la solution numérique d'un problème est une affaire d'analyse numérique, c'est-à-dire de méthode mathématique et non une affaire de calculateur.

1.1.4. CALCUL ANALOGIQUE ET CALCUL NUMERIQUE

Un calcul peut être conduit à partir de données fournies sous formes con-tinue ou sous forme discrète. Egalement les résultats peuvent être fournis sous forme continue ou sous forme discrète. Ceci n'implique en principe rien sur la nature du calculateur mis en oeuvre, car on peut passer de la forme continue à la forme discrète par échantillonnage et de la forme discrète à la forme continue par lissage.

Toutefois, du fait de leur structure, certains calculateurs manipulent des grandeurs variant de façon continue simultanément : ce sont les calculateurs analo-giques alors que d'autres manipulent des grandeurs variant de façon discrète : ce sont les calculateurs numériques.

Dans le fonctionnement des calculateurs analogiques, on met en jeu des ten-sions et courants variant continuement en fonction du temps. Ces calculateurs sont de ce fait bien adaptés pour effectuer des opérations différentielles. Par contre, ils ne peuvent pas stocker un grand nombre de données.

Dans le fonctionnement des calculateurs numériques, on met en jeu des éléments agissant par tout ou rien. Ces calculateurs sont bien adaptés à la résolution de problèmes algébriques. Ils sont doués de mémoires importantes assurant le stockage des données intermédiaires ou des résultats partiels. Ils peuvent par suite mettre en oeuvre des problèmes très importants.

1.1.5. CALCUL EN TEMPS REEL

Les problèmes mis en oeuvre dans les calculateurs sont de deux sortes : d'une part les problèmes statiques dans lesquels les résultats sont des grandeurs indépen-dantes du temps, d'autre part, les problèmes dans lesquels les résultats sont des grandeurs fonction du temps.

En général, on choisit pour la résolution du problème, une échelle des temps différente du temps d'évolution propre du problème. Ceci est indispensable si le problème réel évolue très lentement (par exemple la simulation de réactions chi-miques) ou très rapidement (par exemple l'étude du mouvement d'une particule élec-trisée).

Dans les problèmes dans lequel le calculateur est associé à un système qu'il commande, le calculateur doit avoir comme échelle des temps celle d'évolution du phénomène réel. On dit qu'il fonctionne en "temps réel".

Dans un fonctionnement en temps réel le temps d'exécution d'un calcul s'intègre dans la durée d'évolution du système. Les données sont fournies à des instants imposés par le système et les résultats doivent être disponibles lorsqu'ils sont nécessaires au système pour qu'il puisse poursuivre son évolution.

Chapitre II

LES CALCULATEURS ANALOGIQUES

2.1. GENERALITES

2.1.1. PRINCIPE DES CALCULATEURS ANALOGIQUES

Un phénomène physique évoluant en fonction du temps est décrit lorsqu'on connaît un certain nombre de grandeurs qui le caractérisent et qui évoluent en fonction du temps. Les relations entre les mesures de ces grandeurs se traduisent généralement par des équations algébriques ou des équations différentielles.

Des phénomènes totalement différents peuvent être caractérisées par le même ensemble d'équations. Dans ces conditions, la connaissance de l'évolution d'un phénomène entraîne la connaissance de l'évolution des autres.

Cette analogie est mise à profit dans les calculateurs analogiques. On fait correspondre au système à étudier un système électrique caractérisé par des courants et des tensions régis par les mêmes équations. Aux conditions initiales du système à étudier, on fait correspondre des conditions initiales pour le système électrique analogue et on mesure en fonction du temps les grandeurs électriques analogues des grandeurs caractéristiques du système à étudier.

Partant d'une part pour le système à étudier, d'autre part pour le système électrique analogue de grandeurs qui n'ont aucun rapport entre elles, on ne peut avoir l'identité des mesures pour ces deux ensembles de grandeurs que par un choix convenable des unités.

2.1.2. ORGANES CONSTITUTIFS D'UNE MACHINE ANALOGIQUE

Dans une machine analogique, on mesure des tensions qui évoluent dans le temps conformément à un système d'équations qui constitue les données du problème. Ces tensions peuvent être "continues", c'est-à-dire plus précisément très lentement variables ou "alternatives" ; elles sont alors caractérisées par leur amplitude et leur phase.

Dans ce qui suit, nous nous intéresserons uniquement aux calculeurs anal
giques à "courant continu".

Un calculateur analogique est constitué par un ensemble d'éléments com-
prenant :

1. <u>Les éléments linéaires</u>

Ces éléments permettent de faire en permanence :

a) la somme de grandeurs ou d'intégrales de grandeurs évoluant en fonct
temps

b) le produit par une constante numérique d'une grandeur évoluant en fo
du temps

2. <u>Les éléments non linéaires</u>

Ces éléments permettent de faire en permanence :

a) le produit de deux grandeurs

b) la génération d'une fonction du temps.

3. <u>Les organes de sortie</u>

Les éléments assurent la lecture ou l'enregistrement du résultat.

4. <u>Les alimentations</u>

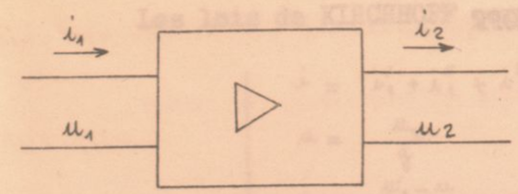
2.2. LES ORGANES LINEAIRES ET LEUR UTILISATION

2.2.1. <u>L'AMPLIFICATEUR</u>

L'amplificateur est l'organe essentiel du calculateur. Il est utilisé
pour faire le changement de signe d'une grandeur, la sommation de plusieurs grand
l'intégration d'une grandeur.

La technologie des amplificateurs sera vue à propos de l'étude du calcu
lateurs NADAC 20 dans un autre chapitre. Les caractéristiques essentielles sont l
suivantes :

L'amplificateur à courant continu d'un calculateur analogique est un él
actif linéaire caractérisé par une très grande résistance d'entrée et un très gra
gain. A tubes dans les anciennes machines, il est actuellement à transistors dans
toutes les machines modernes. L'amplificateur constitue un quadripôle caractéris
par les équations suivantes :



$$\left| \begin{aligned} i_1 &= g_{11} u_1 - g_{12} i_2 \\ u_2 &= g_{21} u_1 - g_{22} i_2 \end{aligned} \right.$$

Celles-ci du fait de la très grande résistance d'entrée et du gain très élevé se réduisent pratiquement à

$$\left| \begin{aligned} i_1 &= 0 \\ u_2 &= g\, u_1 \end{aligned} \right.$$

Une des bornes d'entrées et une des bornes de sortie du quadripôle sont à la masse. Pour simplifier les schémas, dans ce qui suit, on ne représentera pas ces bornes. L'amplificateur sera représenté par le symbole suivant :

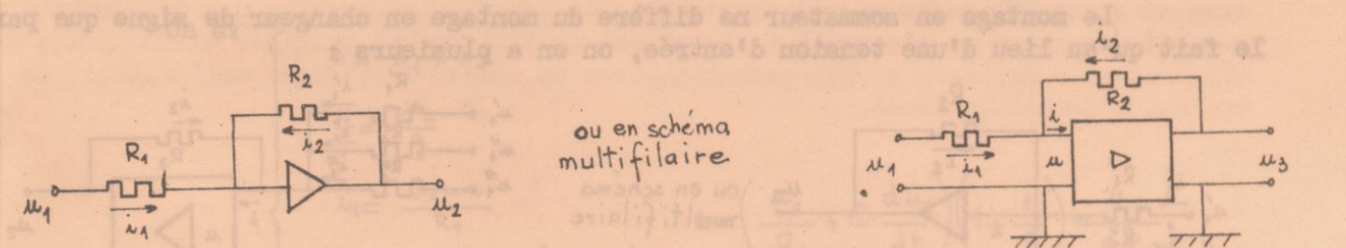
L'amplificateur n'est jamais un organe autonome ; il est associé à un ensemble de résistances ou de condensateurs et on a l'habitude de désigner sous le vocable général "amplificateur", l'ensemble formé par l'amplificateur proprement dit et les composants passifs résistances et condensateurs qui lui sont liés.

Pour éviter les confusions de notations, on désignera par u et i la tension et le courant à l'entrée de l'amplificateur proprement dit et par des lettres indicées les grandeurs électriques tension et courant soit à l'entrée, soit à la sortie de l'ensemble amplificateur et composants liés.

Nous allons voir maintenant les utilisations de l'amplificateur :

2.2.1.1. FONCTIONNEMENT DE L'AMPLIFICATEUR EN CHANGEUR DE SIGNE

Le montage est le suivant :



On a, en appliquant les lois de KIRCHHOFF

$$\begin{cases} i = i_1 + i_2 \\ u = \dfrac{u_2}{g} \\ i_1 = \dfrac{u_1 - u}{R_1} \\ i_2 = \dfrac{u_2 - u}{R_2} \\ i = \dfrac{u}{R_E} \end{cases} \qquad \text{ou} \qquad \frac{u_1}{R_1} + \frac{u_2}{R_2} = \frac{u_2}{g}\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_E}\right)$$

Si, comme nous le supposons, l'amplificateur a une très grande résistanc d'entrée, le courant i peut être considéré comme nul. Si l'amplificateur a un gai très grand, la tension u_2 restant limitée dans le fonctionnement, on a pratiqueme $\frac{u_2}{g}\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_E}\right) = 0$.

Par suite : $\frac{u_1}{R_1} + \frac{u_2}{R_2} = 0$ ou $u_2 = -\frac{R_2}{R_1}u_1$

La tension u_2 se déduit de la tension u_1 par multiplication par une con tante et changement de signe.

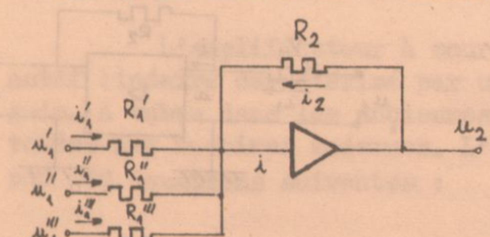
Habituellement $R_2 = R_1$, et par suite $u_2 = -u_1$

Par exemple si $R_1 = 10^6\,\Omega$, $R_2 = 10^6\,\Omega$, $R_E \neq 10^6\,\Omega$ $g = 10^7$ $u_2 = -u_1 + \frac{3}{10^7}u_2$

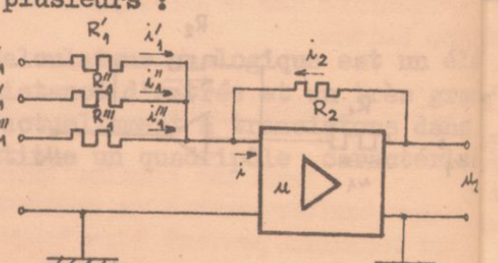
En négligeant le courant i , on commet l'erreur $\frac{\Delta U_2}{U_2} = 3.10^{-7}$

2.3.1.2. FONCTIONNEMENT DE L'AMPLIFICATEUR EN SOMMATEUR

Le montage en sommateur ne diffère du montage en changeur de signe que pa le fait qu'au lieu d'une tension d'entrée, on en a plusieurs :



ou en schéma multifilaire



Les lois de KIRCHHOFF permettent d'écrire :

$$\begin{aligned} i &= i'_1 + i''_1 + i'''_1 + \cdots + i_2 \\ u &= \frac{u_2}{g} \\ i_1 &= \frac{u'_1 - u}{R'_1} \\ i''_1 &= \frac{u''_1 - u}{R''_1} \\ i'''_1 &= \frac{u'''_1 - u}{R'''_1} \\ i &= \frac{u_2}{g R_E} \end{aligned}$$

ou $\frac{u'_1}{R'_1} + \frac{u'_2}{R''_1} + \cdots + \frac{u_2}{R_2} = \frac{u_2}{g}\left(\frac{1}{R'_1} + \frac{1}{R''_1} + \cdots + \frac{1}{R_2} + \frac{1}{R_E}\right)$

Les conditions R_E très grand et g très grand entraînent :

$$\frac{u'_1}{R'_1} + \frac{u''_1}{R''_1} + \cdots + \frac{u_2}{R_2} = 0$$

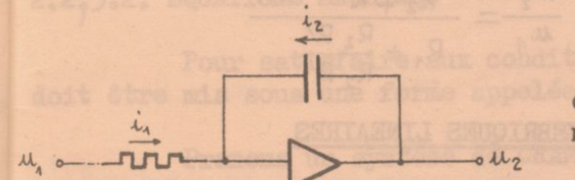
Généralement, on choisit les résistances égales et on a :

$$u_1 + u''_1 + u'''_1 + \cdots + u_2 = 0$$

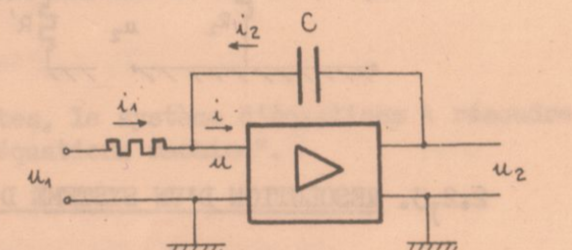
En reprenant le calcul d'erreur fait précédemment avec le changeur de signe, on trouve avec trois entrées dans le sommateur une erreur relative de 5.10^{-7}

2.2.1.3. FONCTIONNEMENT DE L'AMPLIFICATEUR EN INTEGRATEUR

Le montage est le suivant :



ou en schéma multifilaire



On a:

$$\begin{cases} i = i_1 + i_2 \\ u = \dfrac{u_2}{g} \\ i_1 = \dfrac{u_1 - u}{R_1} \\ i_2 = C\dfrac{d(u_2 - u)}{dt} \\ i = \dfrac{u}{R_E} \end{cases} \qquad \text{ou} \qquad \left(\frac{u_1}{R_1} + C\frac{du_2}{dt}\right)\left(1 - \frac{1}{g}\right) = \frac{u_2}{g R_E}$$

Compte tenu que R_E et g sont très grands, on a :

$$\frac{u_1}{R_1} + C\frac{du_2}{dt} = 0$$

où

$$u_2 = -\frac{1}{R_1 C}\int u_1\,dt$$

On choisit généralement les valeurs de R et C de façon que RC soit égal à 1. (par exemple $R_1 = 10^6\,\Omega$, $C = 1\,\mu F$)

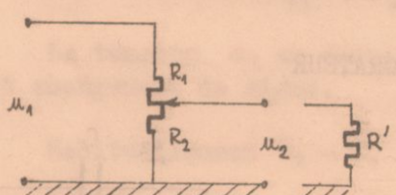
On a alors : $$u_2 = -\int u_1\,dt$$

2.2.2. LES POTENTIOMETRES

Les potentiomètres sont des éléments constitués par une résistance à laquelle on accède par trois points : le pied mis à la masse, le sommet où on appl la tension d'entrées, le curseur sur lequel on recueille la tension de sortie.

Le rapport $\frac{u_2}{u_1} = a$ est fonction de la position du curseur.

On a $$a = \frac{R_2}{R_1 + R_2}$$



Si le potentiomètre débite sur une rési tance R' , le rapport se modifie et c

$$\frac{u_2}{u_1} = \frac{\dfrac{R_2 R'}{R_2 + R'}}{R_1 + \dfrac{R_2 R'}{R_2 R'}}$$

2.2.3. RESOLUTION D'UN SYSTEME D'EQUATIONS ALGEBRIQUES LINEAIRES

Pour résoudre un système d'équations algébriques linéaires, on fait correspondre aux inconnues et aux constantes des tensions mesurables par des nombres proportionnels et on construit à l'aide des amplificateurs utilisés en chargeurs d signe ou en sommateurs et des potentiomètres un réseau électrique satisfaisant au système d'équations donné.

2.2.3.1. CONDITIONS IMPOSEES PAR LA TECHNOLOGIE

Avant la construction du réseau, il y a à tenir compte des considérations suivantes :

1 - Les tensions inconnues, qu'on se donne à priori sur des bornes, sont normalement utilisées comme sources et fournissent de l'énergie soit à d'autres éléments du système, soit à l'organe de mesure qui permet de connaître le résultat. Dans ces conditions, ces bornes doivent obligatoirement être les bornes de sortie d'un amplificateur.

2 - La tension sur le curseur d'un potentiomètre est inférieure à la tension à son sommet (le pied est à la masse). Le potentiomètre servant à faire correspondre à une tension u_1 une tension proportionnelle $u_2 = a\,u_1$ le coefficient multiplicatif est nécessairement inférieur à 1.

Pour des raisons de précision, il est souhaitable qu'il soit supérieur à 0,1.

3 - Aux constantes des équations algébriques correspondent des tensions aux bornes de potentiomètres alimentés par une tension de référence, (par exemple 20 volts pour un calculateur à transistors, 100 volts pour un calculateur à tubes).

On affecte par convention la valeur 1 à cette tension de référence. Il en résulte que tous les termes constants des équations doivent nécessairement être inférieurs à 1.

4 - Les tensions de sortie des amplificateurs doivent rester comprises dans une certaine plage correspondant au fonctionnement correct de cet organe. On choisit normalement la tension de référence (valeur 1) à l'intérieur de cette plage près de la limite supérieure. Les variables inconnues qui apparaissent à la sortie des amplificateurs doivent donc normalement avoir des valeurs inférieures à 1.

2.2.3.2. EQUATIONS MACHINES

Pour satisfaire aux conditions précédentes, le système d'équations à résoudre doit être mis sous une forme appelée "système" d'équations machine".

Prenons un système de CRAMER

$$\left|\begin{array}{l} a_1^1 x_1 + a_1^2 x^2 + \cdots\cdots + a_1^n x_n + b_1 = 0 \\ a_2^1 x_1 + a_2^2 x^2 + \cdots\cdots \qquad + b_2 = 0 \\ a_n^1 x_1 + a_n^2 x^2 + \cdots\cdots \qquad + b_n = 0 \end{array}\right.$$

Examinons d'abord les coefficients des variables. Pour satisfaire à la dition au 2°, on divise chacune des équations par le plus grand des coefficients multiplicatif des variables, de la ligne correspondante

A la place de la matrice

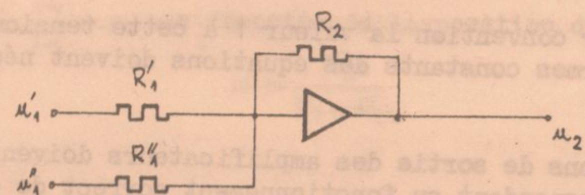
$$\begin{vmatrix} a'_1 \dots a^n_1 \\ \dots\dots \\ a'_n \dots a^n_n \end{vmatrix}$$

on a la matrice

$$\begin{vmatrix} A'_1 \dots A^n_1 \\ \dots\dots \\ A'_n \dots A^n_n \end{vmatrix}$$

dont tous les coefficients sont inférieurs ou égaux à 1

Si aucun de ces coefficients n'est très petit (inférieur à 0,1 par exem la matrice est utilisable pour effectuer ensuite la résolution avec une bonne pré sion en utilisant pour les sommateurs (ou les inverseurs) des résistances toutes égales. Sinon, il faut que certains sommateurs soient montés avec des résistances inégales. Pour des raisons de simplicité, on donne à la résistance de bouclage R_i une valeur uniforme (par exemple 1 MΩ) et on laisse la possibilité de donner au résistances R'_1 R''_1 .... des valeurs différentes. Pratiquement, on se limite à deu (1 MΩ ou 0,1 MΩ par exemple). Dans ces conditions, remplacer une résistance de 1 par une résistance de 0,1 MΩ revient à multiplier par 10 le coefficient affiché le potentiomètre.



En effet, la relation $\frac{u'_1}{R'_1} + \frac{u''_1}{R''_1} + \frac{u_2}{R_2} = 0$ s'écrit avec des résista

$R'_1 = R''_1 = R_2 = 1 M\Omega$ $\qquad u'_1 + u''_1 + u_2 = 0$

avec $R'_1 = 0{,}1 M\Omega$ $\qquad R''_1 = R_2 = 1 M\Omega$ $\qquad 10u'_1 + u''_1 + u_2 = 0$

La règle est maintenant la suivante. Après avoir établi la matrice $\| A^j_i \|$ avec tous les coefficients inférieurs ou égaux à 1, si certains sont très petits (peut multiplier par 10 tous les coefficients de l'équation correspondante.

On a alors des coefficients compris entre 0 et 1 et des coefficients co entre 1 et 10.

Examinons maintenant le traitement des coefficients constants tels que Normalement les coefficients obtenus après le traitement vu précédemment (divisio de la ligne par le plus grand coefficient des variables,) doivent tous être infér à 1. Pour réaliser cette condition, on divise toutes les variables par le plus gr de ces coefficients, par exemple B'_k et on fait le changement de variable $X_i = \frac{x_i}{B'_k}$

Toutefois si les ordres de grandeur des termes constants sont très différents, au lieu de diviser par B'_k on peut diviser par $0{,}1 B'_k$ et appliquer certains termes sur des résistances de 0,1 R .

On a maintenant un système

$$\begin{vmatrix} A^1_1 X_1 + \dots\dots + A^n_1 X_n + B_1 = 0 \\ \\ A^1_n X_1 + \dots\dots + A^n_n X_n + B_n = 0 \end{vmatrix}$$

<u>REMARQUE</u> : On aurait pu penser à priori qu'il serait plus simple de diviser tous les coefficients par le plus grand des nombres de l'ensemble formé par a^j_i et b_i mais en faisant cela on lie obligatoirement d'une manière immuable d'une part la correspondance entre l'échelle des grandeurs fixes et la tension de référence d'autre part la correspondance entre l'échelle des grandeurs variables et la plage utilisable des amplificateurs.

Il n'est plus possible alors de satisfaire à la condition du 4° qui peut se révéler non satisfaite lors de la résolution.

2.2.3.3. SCHEMA DE CABLAGE

On peut maintenant faire le schéma du réseau traduisant les équations machines. Les changements de signe se font à l'aide d'amplificateurs changeurs de signe. Les sommes se font à l'aide d'amplificateurs sommateurs qui sont justement ceux à la sortie desquels apparaissent les variables x , y , z ....

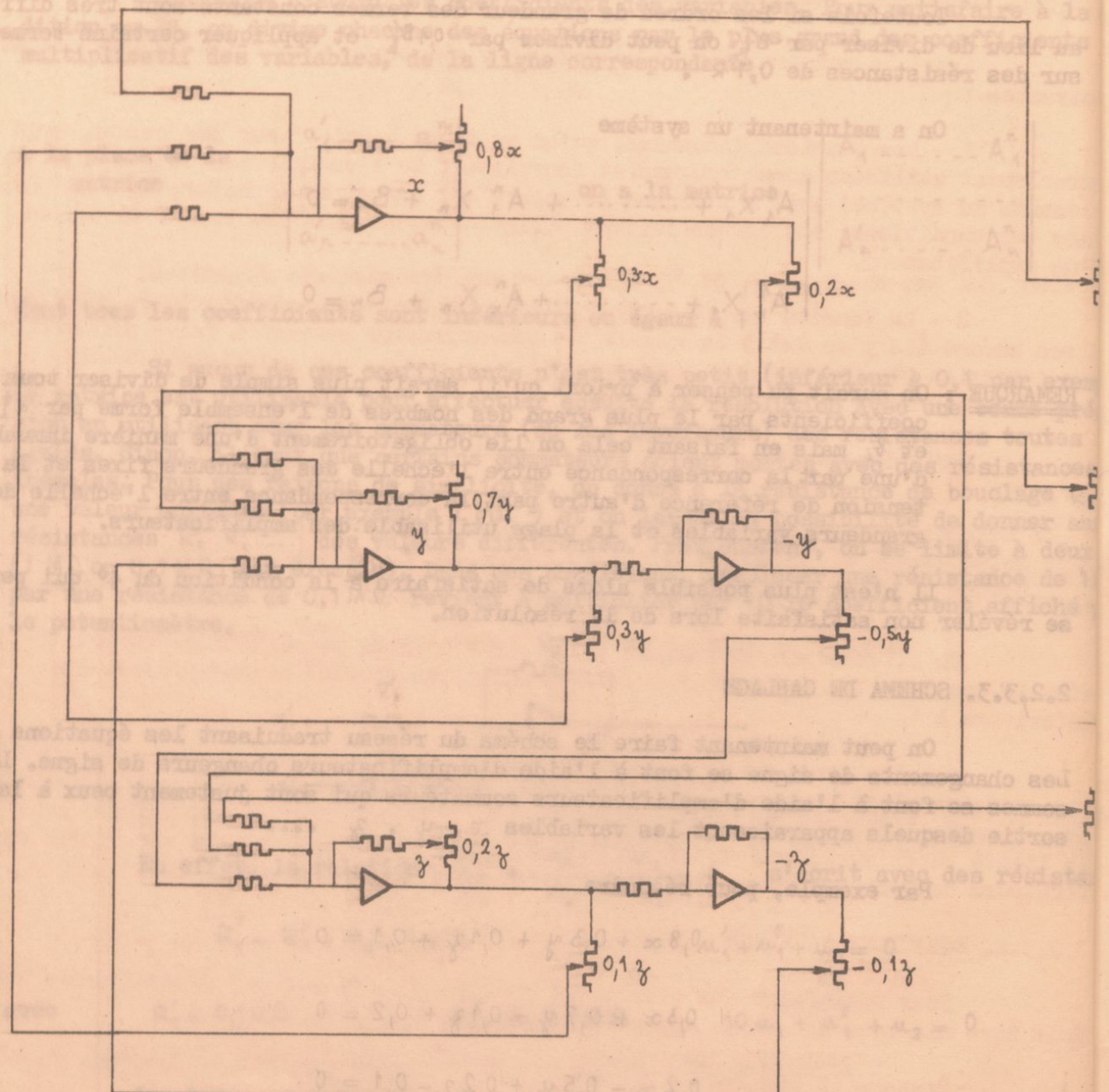
Par exemple, pour résoudre

$$0{,}8x + 0{,}3y + 0{,}1z + 0{,}1 = 0$$

$$0{,}3x + 0{,}7y - 0{,}1z + 0{,}2 = 0$$

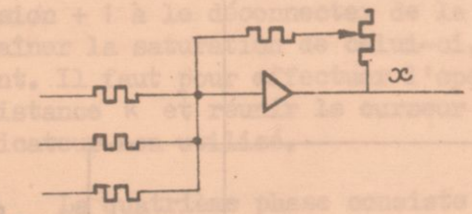
$$0{,}2x - 0{,}5y + 0{,}2z - 0{,}1 = 0$$

On fait le schéma suivant :



2.2.3.4. AFFECTATION DES AMPLIFICATEURS AUX INCONNUES

Nous avons vu qu'au départ on se fixait un certain nombre de bornes sur lesquelles apparaîtront lorsque le problème sera mis en résolution les valeurs des inconnues x, y, z.... , Ces bornes devant être des sorties d'amplificateurs.



Normalement, le bouclage d'un amplificateur doit se faire avec un potentiomètre conformément au schéma ci-contre.

On cherche à éviter chaque fois qu'on le peut de placer un potentiomètre sur un bouclage. Pour cela, on établit la matrice

$$\begin{vmatrix} A_1^1 \ldots\ldots A_1^n \\ A_n^1 \ldots\ldots A_n^n \end{vmatrix}$$

en mettant le maximum de 1 dans la diagonale et on distribue les amplificateurs de façon que la variable qui apparaît en sortie corresponde à un terme de la diagonale de la matrice.

Pour réduire le nombre d'éléments on peut aussi considérer qu'une sortie d'amplificateur représente l'opposé de la variable. La diagonale principale de la matrice comporte alors le maximum de termes égaux à 1 ou − 1.

En reprenant l'exemple précédent on aurait pu écrire

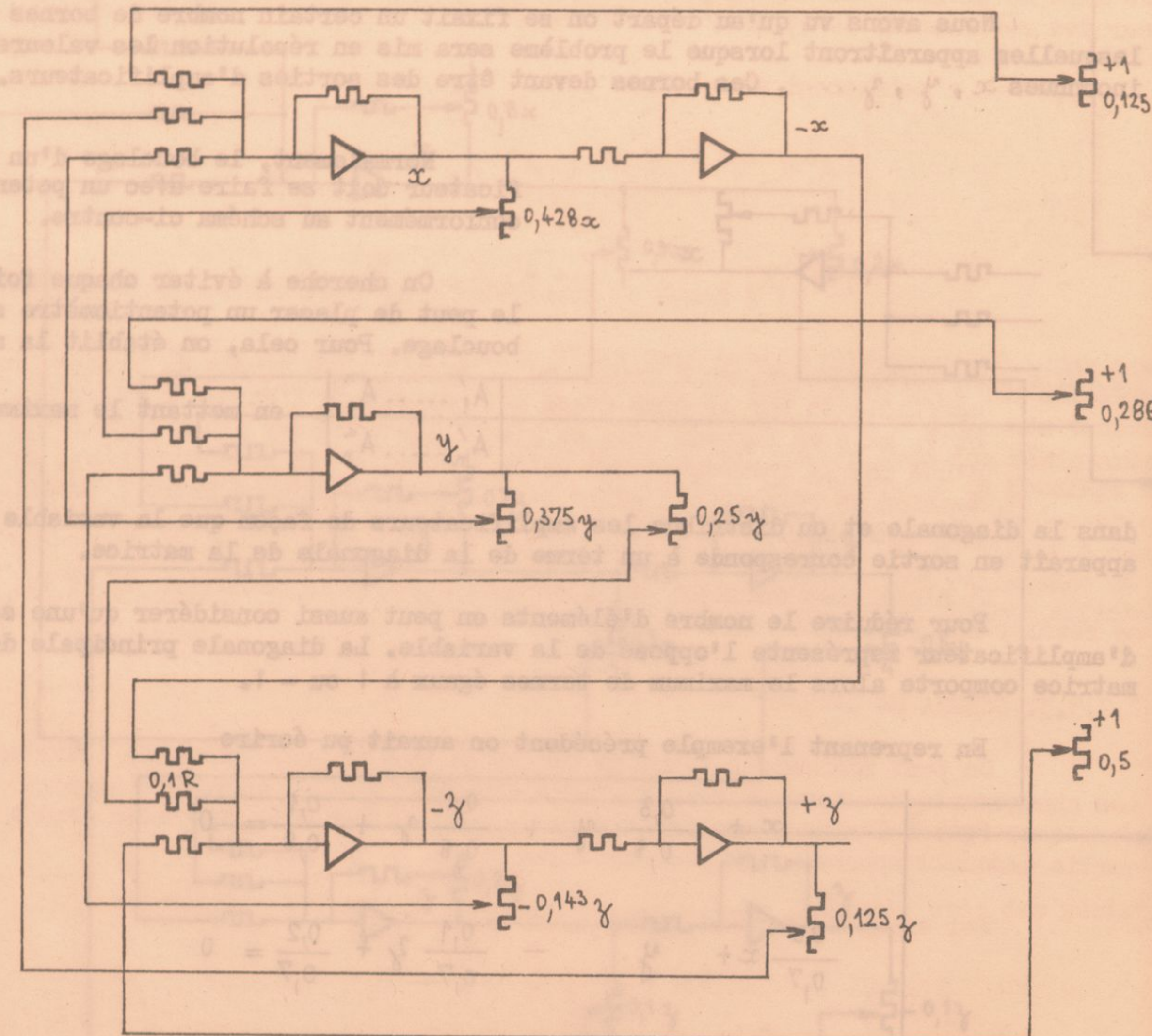
$$\left|\begin{array}{l} x + \dfrac{0,3}{0,8}\,y + \dfrac{0,1}{0,8}\,z + \dfrac{0,1}{0,8} = 0 \\[2ex] \dfrac{0,3}{0,7}\,x + y - \dfrac{0,1}{0,7}\,z + \dfrac{0,2}{0,7} = 0 \\[2ex] -\dfrac{0,2}{0,1}\,x + \dfrac{0,5}{0,2}\,y - z + \dfrac{0,1}{0,2} = 0 \end{array}\right.$$

ou

$$\left|\begin{array}{l} x + 0,375\,y + 0,125\,z + 0,125 = 0 \\ 0,428\,x + y - 0,143\,z + 0,286 = 0 \\ -x + 2,5\,y - z + 0,5 = 0 \end{array}\right.$$

On est amené à utiliser une entrée avec résistance $0,1R$ (entre 10)

Le schéma est alors le suivant :



2.2.3.5. MISE EN RESOLUTION

La mise en oeuvre d'un calculateur pour la résolution d'un système d'équa linéaires se fait en plusieurs temps :

1) élaboration des équations machines
2) câblage des éléments
3) réglage des potentiomètres
4) mise en résolution

Au cours des deux premières phases aucune tension n'est appliquée.

Au cours de la troisième phase on applique la tension + 1 successivement sur chacun des potentiomètres à régler même si celui-ci doit être lors de la mise en résolution alimenté par une sortie d'amplificateur. Le réglage de la tension se fait ainsi en tenant compte des éléments normalement reliés au curseur.

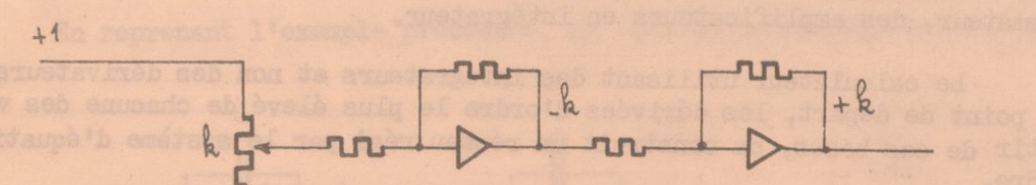
Si un potentiomètre est placé sur un bouclage ont est amené pour lui appliquer la tension + 1 à le déconnecter de la sortie de l'amplificateur ce qui a pour effet d'entraîner la saturation de celui-ci, le bouclage qui assure le rétroacteur disparaissant. Il faut pour effectuer l'opération de réglage boucler l'amplificateur avec la résistance R et réunir le curseur du potentiomètre à la résistance R d'entrée d'un amplificateur non utilisé.

La quatrième phase consiste à appliquer la tension de référence sur les potentiomètres d'affichage des constantes.

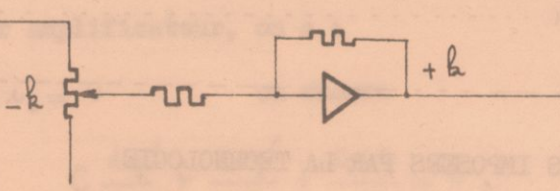
2.2.3.6. SATURATION DES AMPLIFICATEURS

Si l'une des variables dépasse la valeur 1, l'amplificateur n'est plus dans ses conditions normales de fonctionnement. On dit qu'il vient "en saturation". On peut alors réduire proportionnellement toutes les constantes. Toutes les variables sont alors réduites dans le même rapport.

Pour cela, on génère d'autres tensions d'alimentation des potentiomètres d'affichage des constantes à l'aide d'un potentiomètre de deux amplificateurs



On peut s'arranger d'ailleurs dans l'élaboration des équations machines pour que tous les termes constants soient de même signe par exemple positifs. On a alors simplement le montage



2.2,4. RESOLUTION D'UN SYSTEME D'EQUATIONS DIFFERENTIELLES LINEAIRES

Un système d'équations différentielles linéaires à coefficients constan
est un système de la forme

$$a_{1,1,n} \frac{d^n y_1}{dt^n} + a_{1,1,n-1} \frac{d^{n-1} y_1}{dt^{n-1}} + \dots + a_{1,i,n} \frac{d^n y_i}{dt^n} + \dots = f_1(t)$$

- -

$$a_{j,1,n} \frac{d^n y_1}{dt^n} + \dots \dots \dots a_{j,i,n} \frac{d^n y_i}{dt^n} + \dots = f_j(t)$$

- -

L"ordre de dérivation le plus élevé n est l'ordre du système. Certains coefficients peuvent être nuls.

Les fonctions $f_i(t)$ peuvent être identiquement nulles. Le système est alors sans second membre. Dans le cas contraire, il faut générer ces fonctions.

Le système est défini lorsqu'on se donne à l'instant $t = 0$ les valeurs variables et de leurs $n-1$ premières dérivées.

La résolution du système d'équations différentielles est le domaine par excellence des calculateurs analogiques. On utilise en plus des éléments employés pour le calcul algébrique : potentiomètres amplificateurs en changeur de signe ou en sommateur, des amplificateurs en intégrateur.

Le calculateur utilisant des intégrateurs et non des dérivateurs on pre comme point de départ, les dérivées d'ordre le plus élevé de chacune des variable A partir de ces bases, on construit un réseau régi par le système d'équation à résoudre.

Si par exemple on pose $u_1 = \frac{d^n y_1}{dt^n} \dots\dots u_i = \frac{d^n y_i}{dt^n}$

le système précédent s'écrit

$$a_{1,1,n} \; u_1 + a_{1,1,n} \int u_1 \, dt + a_{1,1,n-2} \int dt \int u_1 \, dt + \dots + a_{i,1,n} \, u_i \dots = f_1(t)$$

- -

$$a_{j,1,n} \; u_1 \dots\dots\dots\dots\dots\dots\dots\dots\dots + a_{j,1,n} \, u_i \dots = f_j(t)$$

- -

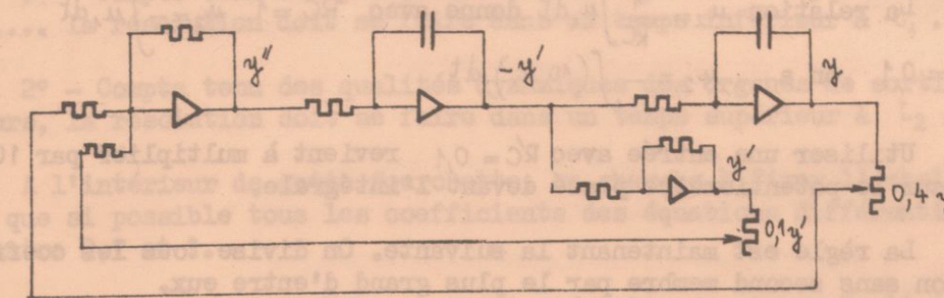
2.2.4.1.. CONDITIONS IMPOSEES PAR LA TECHNOLOGIE

Les conditions imposées par la technologie du calculateur et vues à pr de la résolution des systèmes algébriques se retrouvent de la même façon.

1° - Tous les coefficients affichés à l'aide de potentiomètres doivent être inférieurs à 1. Il est de ce fait indispensable de commencer avant tout affichage, par établir les "équations machines"en divisant chaque équation par le plus grand coefficient qu'elle contient.

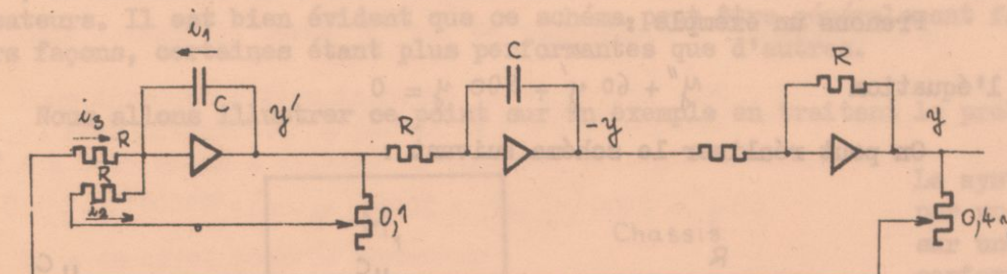
2° - Les bornes de départ du schéma sont des sorties d'amplificateur. On peut leur affecter les dérivées d'ordre le plus élevé des variables.

Si on veut par exemple résoudre l'équation $y'' + 0{,}1\,y' + 0{,}4\,y = 0$
on peut faire le montage suivant :



On peut aussi affecter aux bornes de départ les dérivées d'ordre le plus élevé moins un. Dans ces conditions, l'amplificateur à la sortie duquel elles apparaissent sert à la fois d'intégrateur et de sommateur.

En reprenant l'exemple précédent, on réalise le montage suivant :



Dans le premier amplificateur, on a :

$i_1 + i_2 + i_3 = 0$ ou encore

$$C \frac{dy'}{dt} + \frac{0{,}1 y'}{R} + \frac{0{,}4 y}{R} = 0 .$$

ou, compte tenu que l'on choisit R et C de façon que $RC = 1$

$$y'' + 0{,}1\,y' + 0{,}4\,y = 0$$

Cette méthode permet généralement de diminuer le nombre d'amplificateu
utilisés dans un montage.

Pour obtenir une bonne précision, nous avons indiqué à propos de la ré
des équations algébriques qu'il fallait que les coefficients affichés sur les po
tiomètres ne soient pas trop petits (supérieurs à 0,1 par exemple). Nous avons v
également que dans ce but on utilisait dans les sommateurs des résistances inégal
Un procédé analogue peut être utilisé pour les intégrateurs. Normalement, on choi
la constante de temps $RC = 1\ (1\,M\Omega \times 1\,\mu F)$.

La relation $u_2 = -\frac{1}{RC}\int u_1\,dt$ donne avec $RC = 1$ $u_2 = -\int u_1\,dt$

avec $RC = 0,1$ on a $u_2 = -\int (10\,u_1)\,dt$

Utiliser une entrée avec $R'C = 0,1$ revient à multiplier par 10 le coeffi
affiché sur le potentiomètre placé devant l'intégrale.

La règle est maintenant la suivante. On divise tous les coefficients de
l'équation sans second membre par le plus grand d'entre eux.

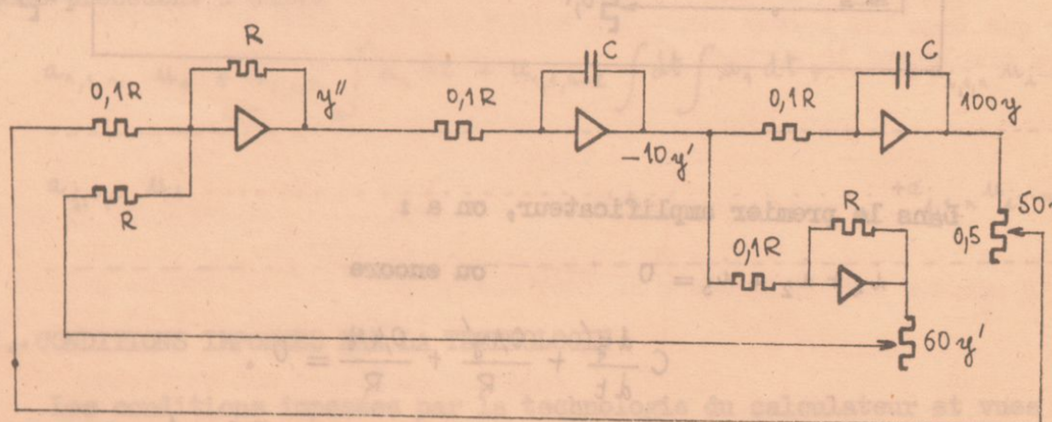
Si les ordres de grandeur des coefficients sont très différents, on peu
utiliser les entrées 0,1 R soit dans les sommateurs (ou chargeurs de signe), so
dans les intégrateurs de façon à multiplier par des puissances de 10 les coeffici
affichés sur les potentiomètres.

Cette possibilité n'est évidemment appliquée qu'après fixation de l'éch
des temps.

Prenons un exemple :

Soit l'équation $y'' + 60\,y' + 500\,y = 0$

On peut réaliser le schéma suivant :



L'équation des noeuds du premier sommateur donne $\frac{y''}{R} + \frac{60\,y'}{R} + \frac{50\,y}{0,1R} = 0$

soit $y'' + 60\,y' + 500\,y = 0$

2.2.4.2. ECHELLE DES TEMPS

Pour fixer l'échelle des temps, on doit prendre en considération les points suivants :

1° - Compte tenu des performances des composantes amplificateurs, condensateurs..... la résolution doit se faire dans un temps inférieur à t_1 .

2° - Compte tenu des qualités dynamiques des organes de sortie tels qu'enregistreurs, la résolution doit se faire dans un temps supérieur à t_2 .

A l'intérieur de cette fourchette, on cherche à fixer l'échelle des temps de façon que si possible tous les coefficients des équations différentielles soient compris entre 0,1 et 1.

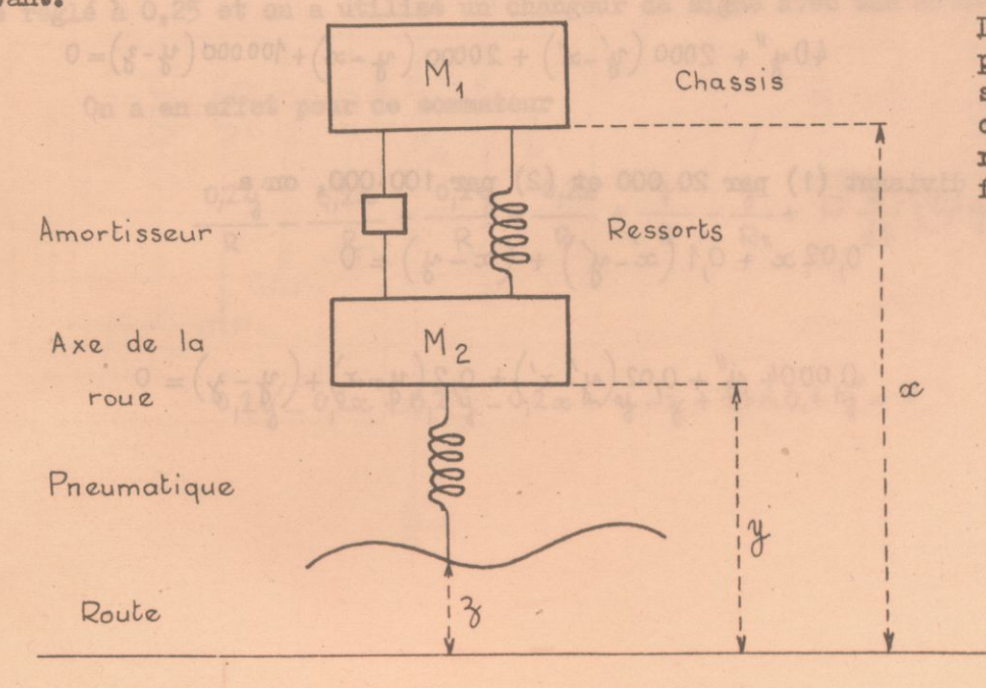
Si en effet on fait le changement d'échelle des temps $\theta = kt$ (θ étant le temps machine), on a $\frac{d^n x}{dt^n} = k^n \frac{d^n x}{d\theta^n}$. Il en résulte une modification de tous les coefficients.

2.2.4.3. CHOIX DU SCHEMA

Après avoir, compte tenu des règles précédentes, établi les "équations machine ", on construit le schéma de connexion des différents organes : potentiomètres, amplificateurs. Il est bien évident que ce schéma peut être généralement fait de plusieurs façons, certaines étant plus performantes que d'autres.

Nous allons illustrer ce point sur un exemple en traitant le problème suivant.

Le système formé par une automobile sur une route est conforme au modèle représenté sur la figure ci-contre :



Le système est régi par les équations suivantes :

$$M_1 \frac{d^2 x}{dt^2} + C_1\left(\frac{dx}{dt} - \frac{dy}{dt}\right) + K_1(x-y) = 0$$

$$M_2 \frac{d^2 y}{dt^2} + C_1\left(\frac{dy}{dt} - \frac{dx}{dt}\right) + K_1(y-x) + K_2(y-z) = 0$$

dans lesquelles :

- M_1 : Quart du poids de l'automobile
- M_2 : Poids de la roue et de l'axe
- K_1 : Coefficent d'élasticité du ressort
- C_1 : Coefficient d'amortissement
- K_2 : Coefficient d'élasticité du pneumatique
- x : Déplacement du chassis par rapport à une référence plane.
- y : Déplacement de l'axe de la roue par rapport à une référence plane
- z : Profil de la route.

Les constantes ont les valeurs suivantes :

$M_1 = 400$ kg, $M_2 = 40$ kg, $C_1 = 2000$ kg/s, $K_1 = 20\,000$ $K_2 = 100\,000$ kg

On a le système d'équations

(1) $$400x'' + 2000(x'-y') + 20000(x-y) = 0$$

(2) $$40y'' + 2000(y'-x') + 20000(y-x) + 100000(y-z) = 0$$

En divisant (1) par 20 000 et (2) par 100 000, on a

(1)' $$0{,}02x'' + 0{,}1(x'-y') + (x-y) = 0$$

(2)' $$0{,}0004y'' + 0{,}02(y'-x') + 0{,}2(y-x) + (y-z) = 0$$

Le changement d'échelle des temps $\theta = kt$ donne

(1)" $$0{,}02k^2 x'' + 0{,}1k(x'-y') + (x-y) = 0$$

(2)" $$0{,}0004k^2 y'' + 0{,}02k(y'-x') + 0{,}2(y-x) + y - z = 0$$

Dans (1)", on pourrait rendre le coefficient de x'' égal à 1, en prenant $0{,}02\,k^2 = 1$. Pour avoir un rapport d'échelle des temps simples, on prendra $k^2 = 100$ c'est-à-dire $k = 10$.

On a alors le système

(1)"' $$x'' + 0{,}5(x'-y') + 0{,}5(x-y) = 0$$

(2)"' $$0{,}04y'' + 0{,}2(y'-x') + 0{,}2(y-x) + (y-z) = 0$$

Une solution est constituée par le schéma de la page suivante.

Il est à noter que dans l'intégrateur sommateur relatif à la deuxième équation, on a pris une constante de temps $RC = 0{,}1$ en utilisant des entrées $0{,}1R$

Pour obtenir le terme $0{,}04y''$ on est parti de $0{,}4y'$

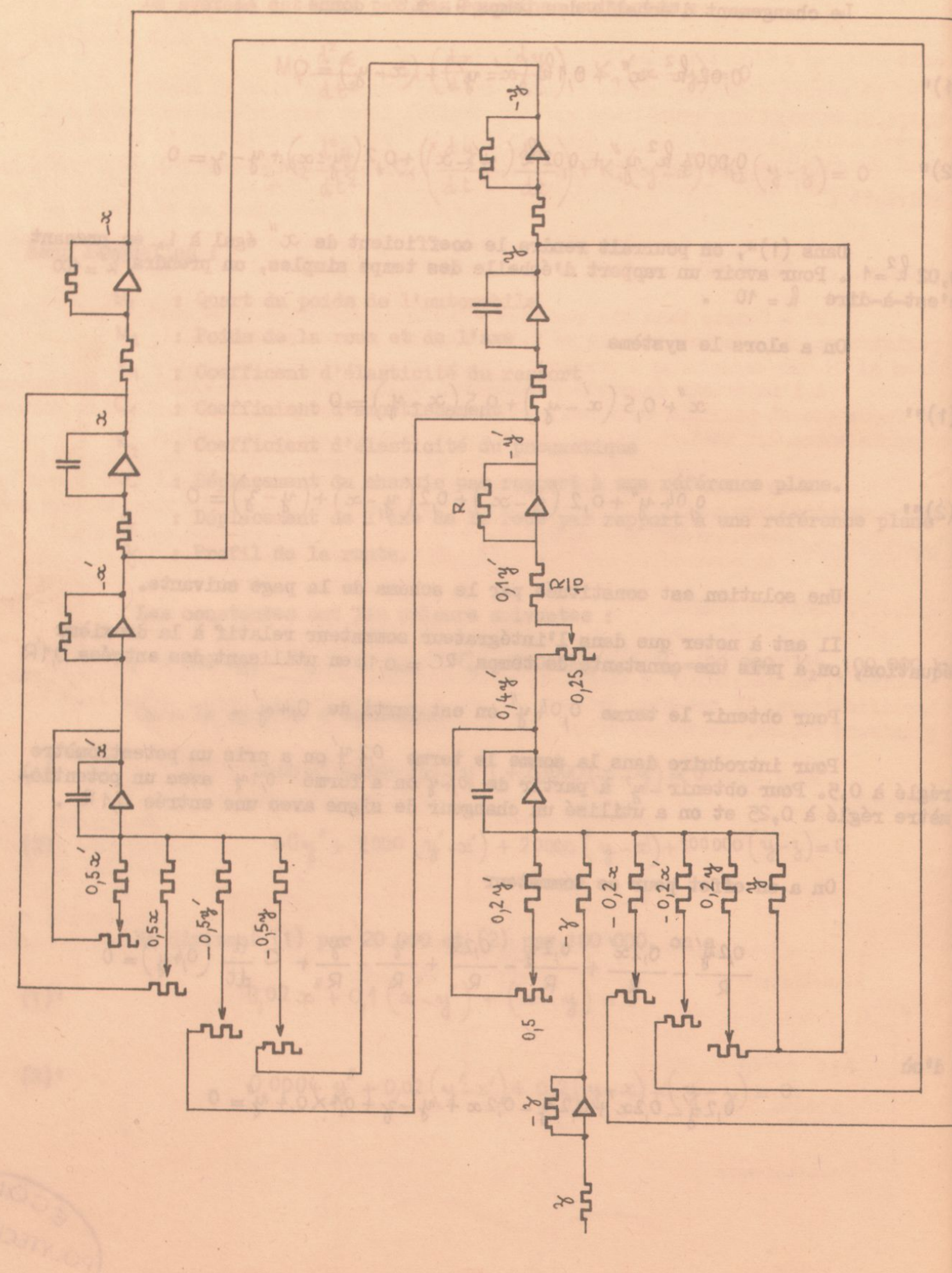
Pour introduire dans la somme le terme $0{,}2y'$ on a pris un potentiomètre réglé à 0,5. Pour obtenir $-y'$ à partir de $0{,}4y'$ on a formé $0{,}1y'$ avec un potentiomètre réglé à 0,25 et on a utilisé un changeur de signe avec une entrée $0{,}1R$.

On a en effet pour ce sommateur

$$\frac{0{,}2y'}{R} - \frac{0{,}2x'}{R} + \frac{0{,}2y}{R} - \frac{0{,}2x}{R} + \frac{y}{R} - \frac{z}{R} + C\frac{d}{dt}(0{,}4y') = 0$$

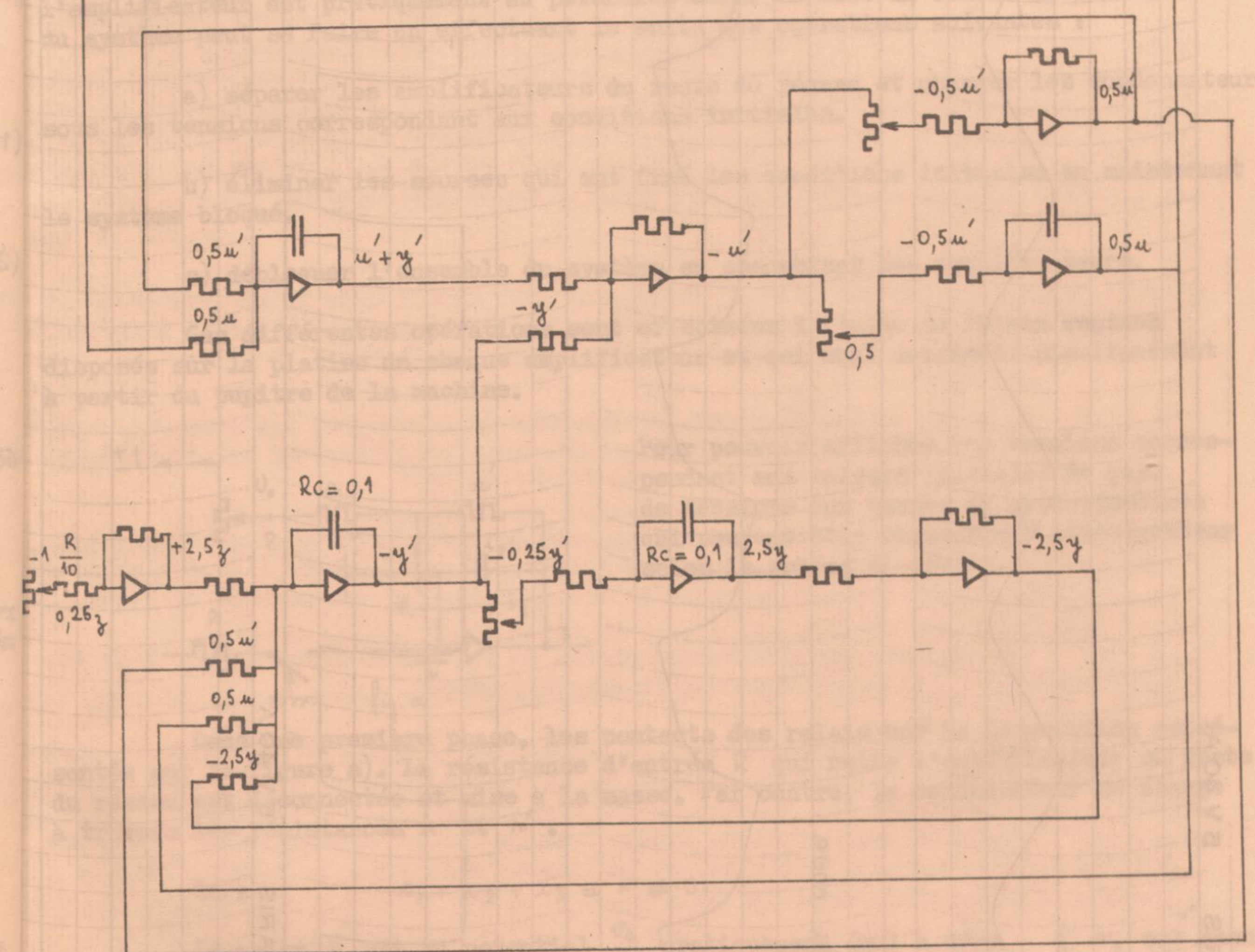
d'où

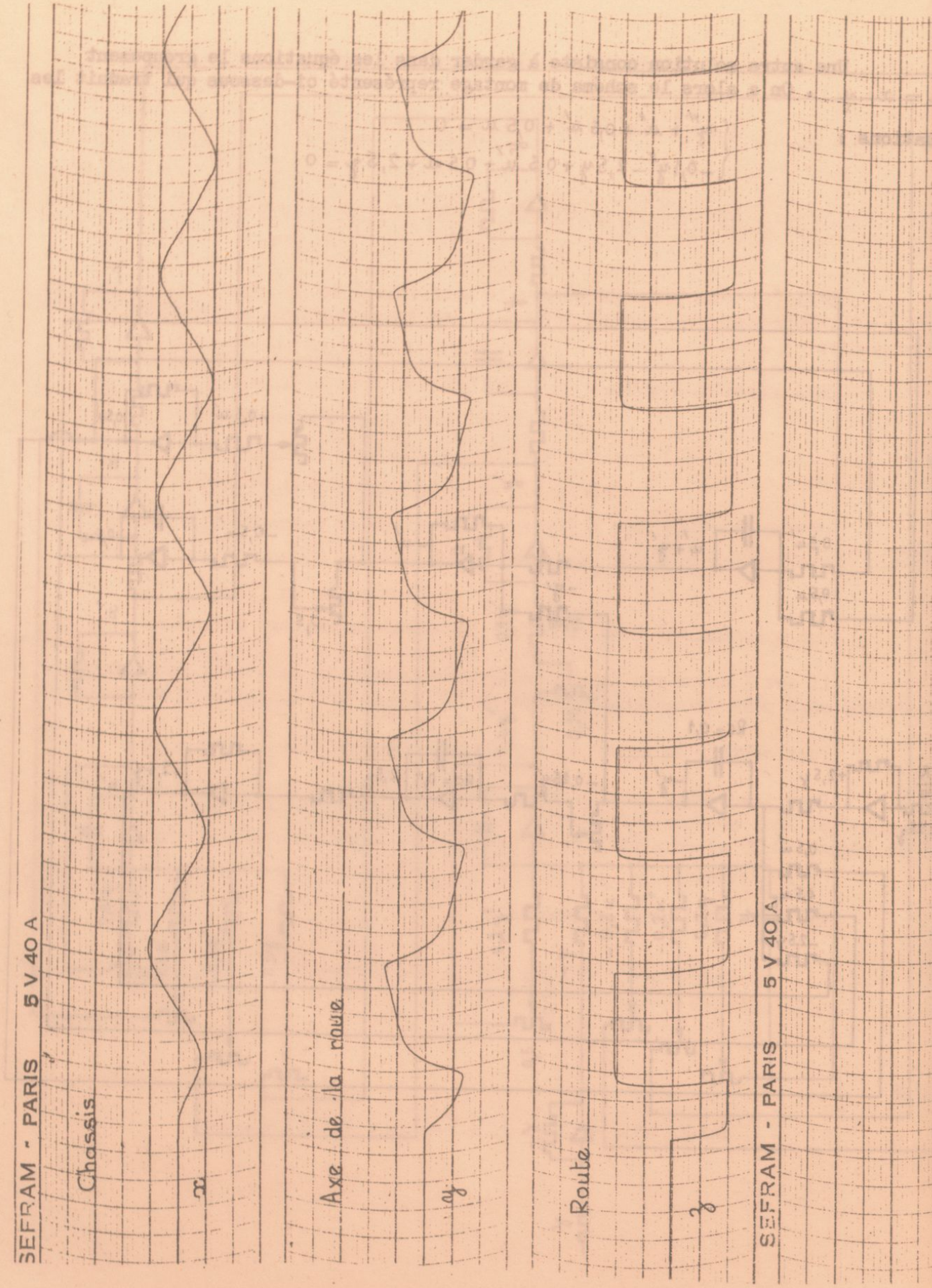
$$0{,}2y' - 0{,}2x' + 0{,}2y - 0{,}2x + y - z + 0{,}1 \times 0{,}4\,y'' = 0$$



Une autre solution consiste à garder dans les équations le groupement $u = x - y$. On a alors le schéma de montage représenté ci-dessous qui traduit les équations :

$$\begin{cases} y'' + u'' + 0{,}5\,u' + 0{,}5\,u = 0 \\ -0{,}1\,y'' - 2{,}5\,y + 0{,}5\,u' + 0{,}5\,u + 2{,}5\,z = 0 \end{cases}$$





2.2.4.4. APPLICATION DES CONDITIONS INITIALES

La solution d'un système d'équations différentielles est constitué par un ensemble de tensions évoluant en fonction du temps à partir d'un instant pris comme origine des temps et auquel correspondent des conditions initiales. Celles-ci sont constituées pour chacune des variables par la donnée de sa valeur et de celles de ses $n-1$ premières dérivées (n est l'ordre maximal de la dérivée de la variable dans l'équation différentielle).

Les variables et leurs $n-1$ premières dérivées apparaissent sous forme de tensions à la sortie d'amplificateurs montés en intégrateurs. Ces tensions sont celles qui existent aux bornes des condensateurs du montage intégrateur puisque l'entrée de l'amplificateur est pratiquement au potentiel zéro. La mise en oeuvre de la résolution du système peut se faire en effectuant la suite des opérations suivantes :

a) séparer les amplificateurs du reste du réseau et charger les condensateurs sous les tensions correspondant aux conditions initiales.

b) éliminer les sources qui ont fixé les conditions initiales en maintenant le système bloqué.

c) débloquer l'ensemble du système en connectant les amplificateurs.

Ces différentes opérations sont effectuées à l'aide de relais rapides disposés sur la platine de chaque amplificateur et qui sont commandés simultanément à partir du pupitre de la machine.

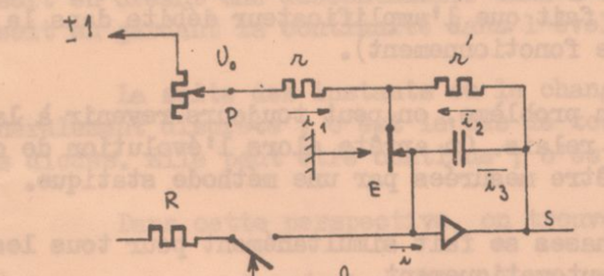


fig. a

Pour pouvoir afficher les tensions correspondant aux valeurs initiales on part de tensions aux bornes de potentiomètres qui peuvent être connectés à l'intégrateur selon le schéma ci-contre.

Dans une première phase, les contacts des relais ont la disposition représentée sur la figure a). La résistance d'entrée R qui relie l'amplificateur au reste du réseau est déconnectée et mise à la masse. Par contre, le condensateur se charge à travers les résistances r et r'.

On a $\qquad i_1 + i_2 + i_3 = i \# 0$

Le point E est au potentiel $\frac{v_s}{g}$ pratiquement égal à zéro (v_s est borné supérieurement et g est très grand). On a donc

$$C\frac{dv_s}{dt} + \frac{v_s}{r'} + \frac{V_0}{r} = 0$$

ou encore $$v_s = -\frac{r'}{r}\, U_0 \left(1 - e^{-\frac{t}{r'c}}\right)$$

La constante de temps $r'c$ étant assez faible (ordre de 0,1 s) très rap
dement, on a la tension $-\frac{r'}{r}\, U_0$ (ou la valeur à 1/1000 au bout de 7 fois la const
de temps.

En agissant sur le potentiomètre, on règle la valeur $-\frac{r'}{r}\, U_0$ (c'est
tension que l'on mesure et non pas U_0).

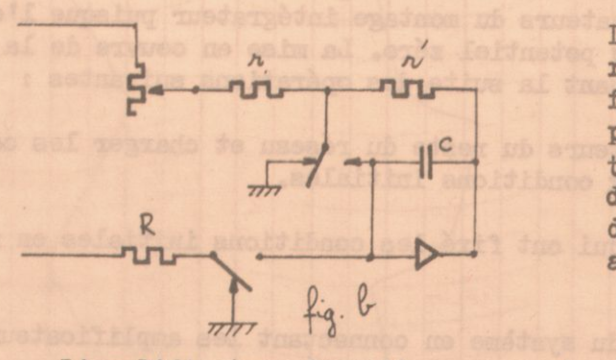


fig. b

Dans une deuxième phase, les contac
relais ont la disposition représent
figure b). Le courant i d'entrée d
plificateur étant extrêmement faibl
très grand), le condensateur ne peu
décharger et garde à ses bornes la
de sortie v_s (l'entrée est pratique
au potentiel zéro).

L'amplificateur fournit un courant qui circule dans la résistance r' et
dans les résistances qui peuvent être connectées à la sortie mais cela n'influe
rien sur la tension v_s aux bornes du condensateur.

Dans une troisième phase, on connecte la résistance d'entrée R . L'
lution du problème se produit alors (le fait que l'amplificateur débite dans la
résistance r' , n'influe en rien sur le fonctionnement).

Au cours de la résolution d'un problème, on peut toujours revenir à la
disposition précédente des contacts des relais. On arrête alors l'évolution de c
Les grandeurs sont "figées" et peuvent être mesurées par une méthode statique.

La commande des différentes phases se fait simultanément pour tous les
amplificateurs, soit manuellement soit automatiquement.

Dans la commande manuelle, on utilise sur le pupitre une suite de bout
correspondant aux fonctions suivantes.

Attente : on n'applique aucune tension (ni la tension du potentiomètre
d'affichage des valeurs initiales, ni la liaison d'entrée), le condensateur se
décharge dans la résistance r' .

Valeurs initiales : on met en oeuvre la première phase décrite précéde

Mémoire : on met en oeuvre la deuxième phase.

Résolution : on met en oeuvre la troisième phase.

Dans la commande automatique, on répète cycliquement ces opérations au bout
'un temps auquel on peut donner différentes valeurs. Le problème se trouve répété à
n rythme compatible avec l'utilisation d'un oscilloscope comme organe de sortie.

.3. LES ORGANES NON LINEAIRES ET LEUR UTILISATION

.3,1. SYSTEMES NON LINEAIRES

Un grand nombre de systèmes sont régis par des équations linéaires ou plus
précisément on peut considérer avec une approximation suffisante que les relations
ntre les différentes grandeurs qui les caractérisent sont régies par des équations
lgébriques linéaires ou des équations différentielles linéaires à coefficients con-
tants.

Il existe aussi une classe très importante de systèmes où cette approxima-
ion ne peut être faite. On considère alors que le domaine d'évolution des grandeurs
peut être fractionné en intervalles de temps élémentaires au cours desquels les gran-
leurs sont régies par des systèmes d'équations algébriques ou différentielles linéai-
res. Aux instants qui séparent les différents domaines, selon le type de problème, on
considère ou bien que les grandeurs subissent une discontinuité ou bien qu'elles res-
tent continues.

Pour adapter un calculateur à la résolution de problèmes non linéaires, il
est nécessaire d'adjoindre des organes de commutation qui changent la structure des
circuits, c'est-à-dire les équations qui régissent le système. Cette opération peut
se faire :

- soit en créant une discontinuité dans l'évolution des grandeurs,
- soit en gardant la continuité dans l'évolution des grandeurs.

La suite des instants où le changement de structure du circuit a lieu est
généralement discrète : c'est le cas de tous les systèmes électroniques qui utilisent
les diodes. Elle peut être continue ; c'est le cas des systèmes à servomécanismes.

Dans cette perspective, on trouve donc comme organes non linéaires.

1°) les organes qui créent une discontinuité dans l'évolution des grandeurs

Les composants qui entrent dans leur constitution sont essentiellement des
diodes et des relais. Ils permettent de simuler des phénomènes tels que les seuils,
les jeux d'engrenages, les vitesses limites de moteurs, les hystérésis et les satu-
rations magnétiques.

2°) les organes qui ne créent pas de discontinuité dans l'évolution des grandeurs
mais pour lesquels les intervalles de temps entre deux commutations sont dis-
crets.

Les composants sont essentiellement des diodes.

Ils permettent de générer des fonctions par exemple $y = x^2$, $y = \sin x$
$y = \log x$ ou d'une façon générale une fonction $y = f(x)$ donnée par une courbe.

Les multiplieurs à diodes qui font l'opération $z = xy$ appartiennent à
classe d'organes. On réalise en effet la suite d'opérations

$$z = \frac{1}{4}\left[(x+y)^2 - (x-y)^2\right]$$

3° - les organes qui ne créent pas de discontinuité dans l'évolution d
grandeurs et qui assurent une modification continue des circuits.

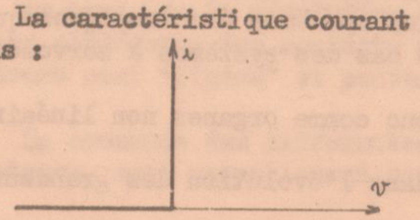
Ces organes sont essentiellement des servomécanismes qui à une tension
correspondre un angle θ proportionnel et permettent ainsi de positionner un org
un potentiomètre dans le cas d'un multiplieur à servomécanismes, un codeur dans l
cas où on veut passer à une grandeur numérique définie par exemple dans un code b

2.3.2. LES DIODES

La diode (symbole —▶|—) est un dipôle qui a une tension fait correspon
un courant non proportionnel. La relation de correspondance dépend essentiellemen
de la technologie de l'élément.

Pour étudier commodément les circuits qui comportent des diodes, on déf
la diode idéale qui pour une tension négative (on définit l'entrée et la sortie d
la diode) est équivalent à un interrupteur ouvert et pour une tension positive es
équivalente à un interrupteur fermé.

La caractéristique courant tension de cette diode idéale est représenté
ci-dessous :



Pour $v < 0$ la diode est bloquée

Pour $v > 0$ la diode est passante

Les diodes que l'on construit effectivement n'ont pas cette caractérist
idéale. Les diodes à vide ne sont plus utilisées actuellement dans les calculateu
Les diodes à semi-conducteur ont une caractéristique ayant l'allure représentée c
dessous, sur la figure a). On peut l'approximer par les caractéristiques représen
sur la figure b).

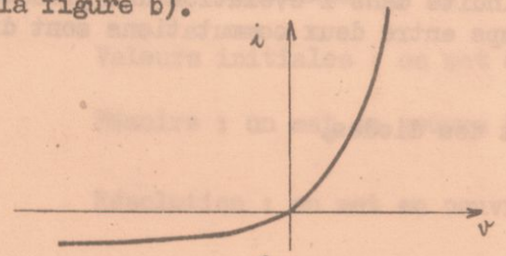


fig. a

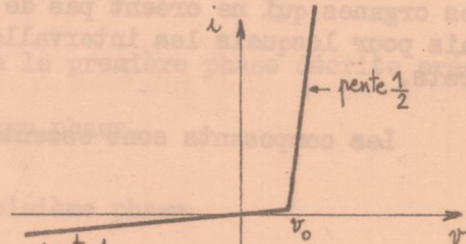


fig. b

Cette caractéristique peut être obtenue également à partir d'une diode idéale et d'éléments linéaires conformément au schéma de la figure c).

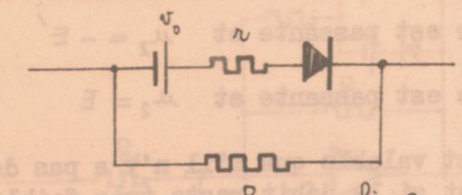
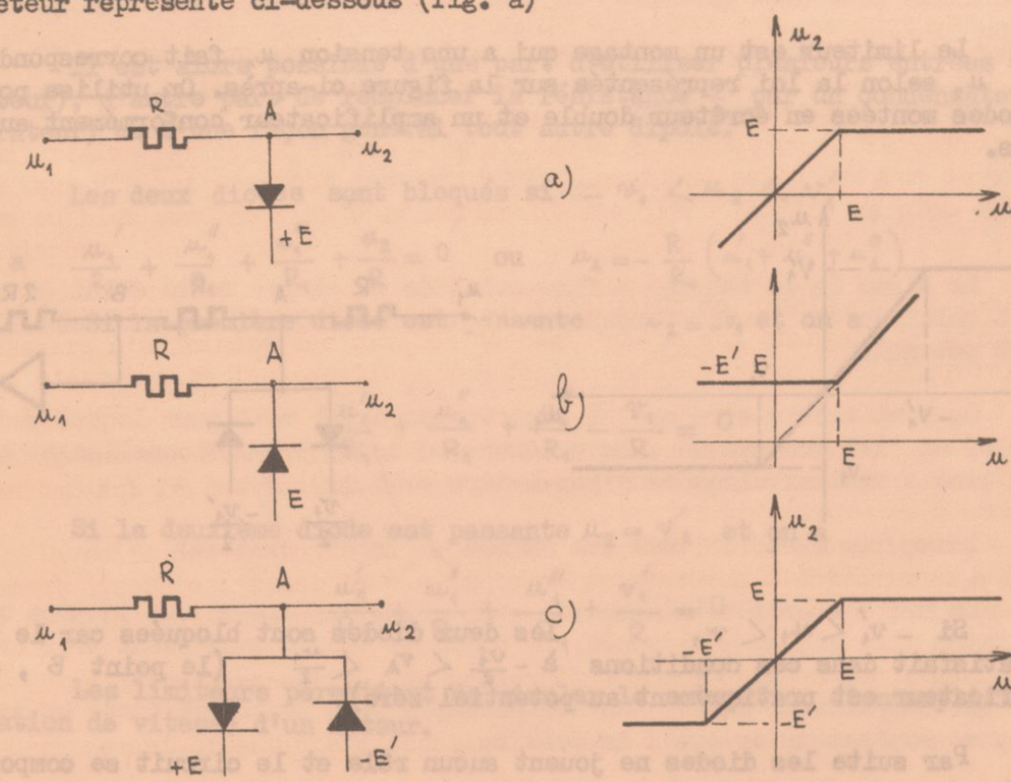


fig. c

La diode idéale correspond au cas où la tension v_0 appelée tension de déchet est nulle la résistance directe r est nulle, la résistance inverse R est infinie.

2.3.3. ECRETEURS

Une utilisation importante de la diode dans les calculateurs est le montage en écrêteur représenté ci-dessous (fig. a)



Si $u_1 \leqslant E$ la diode est bloquée et $u_2 = u_1$. Si $u_1 \geqslant E$ la diode est passante et $u_2 = E$.

En inversant le sens de la diode on a le montage de la figure b).

Si $u_1 \geqslant E$ la diode est bloqué et $u_2 = u_1$. Si $u_1 \leqslant E$ la diode est passante et $u_2 = E$

On peut associer les deux montages pour former un écrêteur double. (fig. c)

Si $-E' \leq u_1 \leq E$ les diodes sont bloquées et $u_2 = u_1$

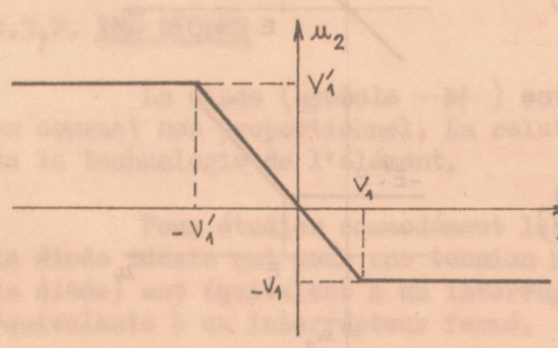
Si $u_1 \leq -E'$ la diode de droite est passante et $u_2 = -E'$

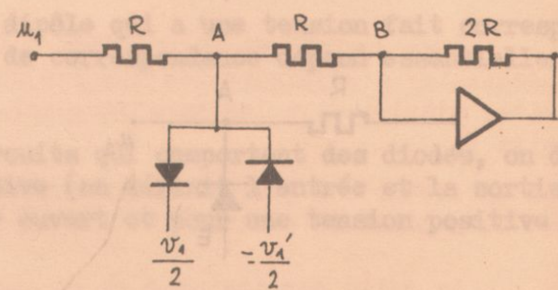
Si $u_1 \geq E$ la diode de gauche est passante et $u_2 = E$

Naturellement, le raisonnement n'est valable que s'il n'y a pas débit la sortie du montage. Il le reste pratiquement si le débit reste très faible. Ce montages se trouvent donc placés devant des amplificateurs.

2.3,4. LIMITEUR

Le limiteur est un montage qui a une tension u_1 fait correspondre une tension u_2 selon la loi représentée sur la figure ci-après. On utilise pour cel deux diodes montées en écrêteur double et un amplificateur conformément au schém ci-après.





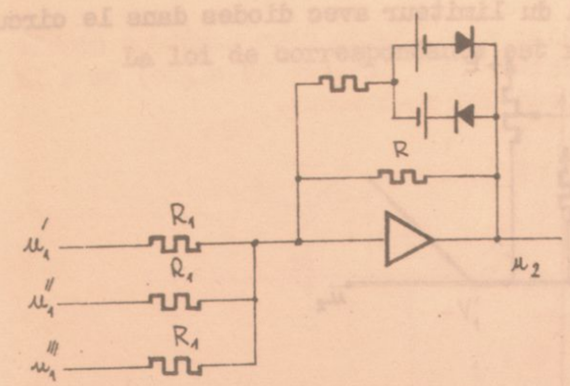
Si $-v_1' < u_1 < v_1$ les deux diodes sont bloquées car le potent de A satisfait dans ces conditions à $-\frac{v_1'}{2} < v_A < \frac{v_1}{2}$ (le point B, entrée l'amplificateur est pratiquement au potentiel zéro).

Par suite les diodes ne jouent aucun rôle et le circuit se comporte co un simple changeur de signe.

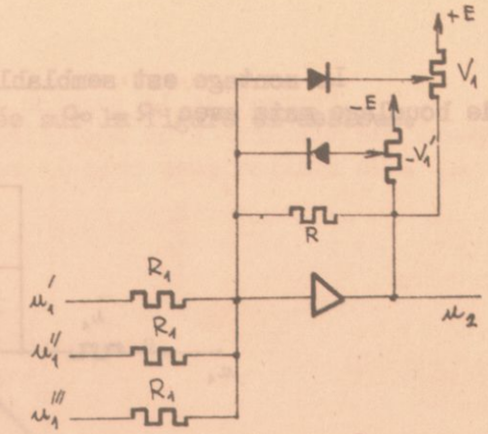
Si $u_1 < -v_1'$ la diode de droite est passante. Le potentiel de A est fixé à $-\frac{v_1'}{2}$ et le potentiel u_2 est égal à $+v_1'$.

De même, si $u_1 > \frac{v_1}{2}$ la diode de gauche est passante ; le potentiel est fixé à $\frac{v_1}{2}$ et le potentiel u_2 est égal à $-v_1$,

Une variante de ce montage consiste à placer les diodes dans le circuit bouclage de l'amplificateur conformément au schéma ci-après.



ou



Il est alors possible d'une part d'utiliser plusieurs entrées (montage en sommateur), d'autre part de remplacer la résistance R par un condensateur (montage intégrateur) ou d'une façon général tout autre dipôle.

Les deux diodes sont bloqués si $-v_1 < u_2 < v_1'$

et on a $\frac{u_1'}{R_1} + \frac{u_1''}{R_1} + \frac{u_1'''}{R_1} + \frac{u_2}{R} = 0$ ou $u_2 = -\frac{R}{R_1}\left(u_1' + u_1'' + u_1'''\right)$

Si la première diode est passante $u_2 = -v_1$ et on a

$$\frac{u_1'}{R_1} + \frac{u_1''}{R_1} + \frac{u_1'''}{R_1} - \frac{v_1}{R} = 0$$

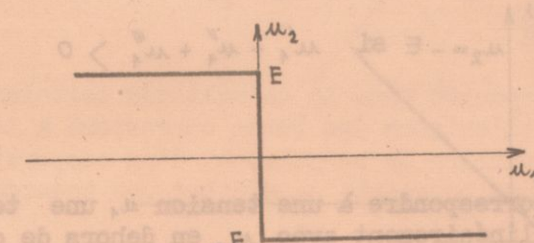
Si la deuxième diode est passante $u_2 = v_1'$ et on a

$$\frac{u_1'}{R_1} + \frac{u_1''}{R_1} + \frac{u_1'''}{R_1} + \frac{v_1'}{R} = 0$$

Les limiteurs permettent de simuler la saturation d'un amplificateur, la limitation de vitesse d'un moteur.

2.3,5. COMPARATEUR

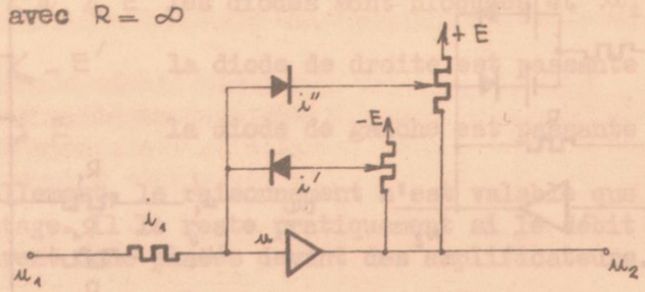
Le comparateur établit la correspondance suivante



$u_2 = E$ si $u_1 < 0$

$u_2 = -E$ si $u_1 > 0$

Le montage est semblable à celui du limiteur avec diodes dans le circui
de bouclage mais avec $R = \infty$



Une seule diode peut être passante, le choix entre les deux étant lié au signe de
la tension u_1.

La loi des neouds donne

soit $\begin{cases} i_1 + i' = 0 \\ i'' = 0 \end{cases}$ soit $\begin{cases} i_1 + i'' = 0 \\ i' = 0 \end{cases}$

Le signe de la tension u_1 fixe celui de u et par suite celui de u_2
(opposé à celui de u_1). En conséquence il détermine celui des courants i' ou i
qui n'est pas nul.

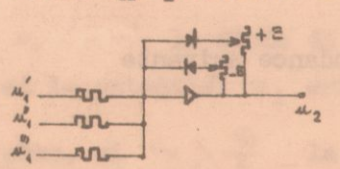
On peut encore dire que si à partir du point $u_1 = 0$ pour lequel aucun cou
ne circule, on fait croitre u_1 , la tension $-u_2$ croît proportionnellement tant qu
les deux diodes restent bloquées c'est-à-dire tant que $-E < u_2 < E$

Lorsqu'on atteint l'une des bornes u_2 reste constant.

Le montage permet de comparer avec grande précision une tension variabl
à une tension fixe réglée à l'avance, le temps de basculement étant très bref.

Le comparateur entre dans des montages plus complexes comme un basculeu
bistable, un générateur de dents de scie...

En utilisant plusieurs entrées on a de la même façon



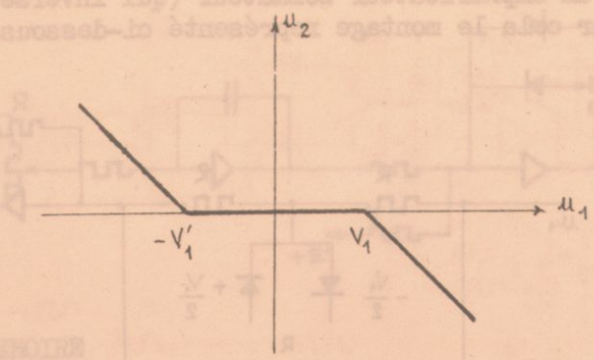
$u_2 = E$ si $u'_1 + u''_1 + u'''_1 < 0$

$u_2 = -E$ si $u'_1 + u''_1 + u'''_1 > 0$

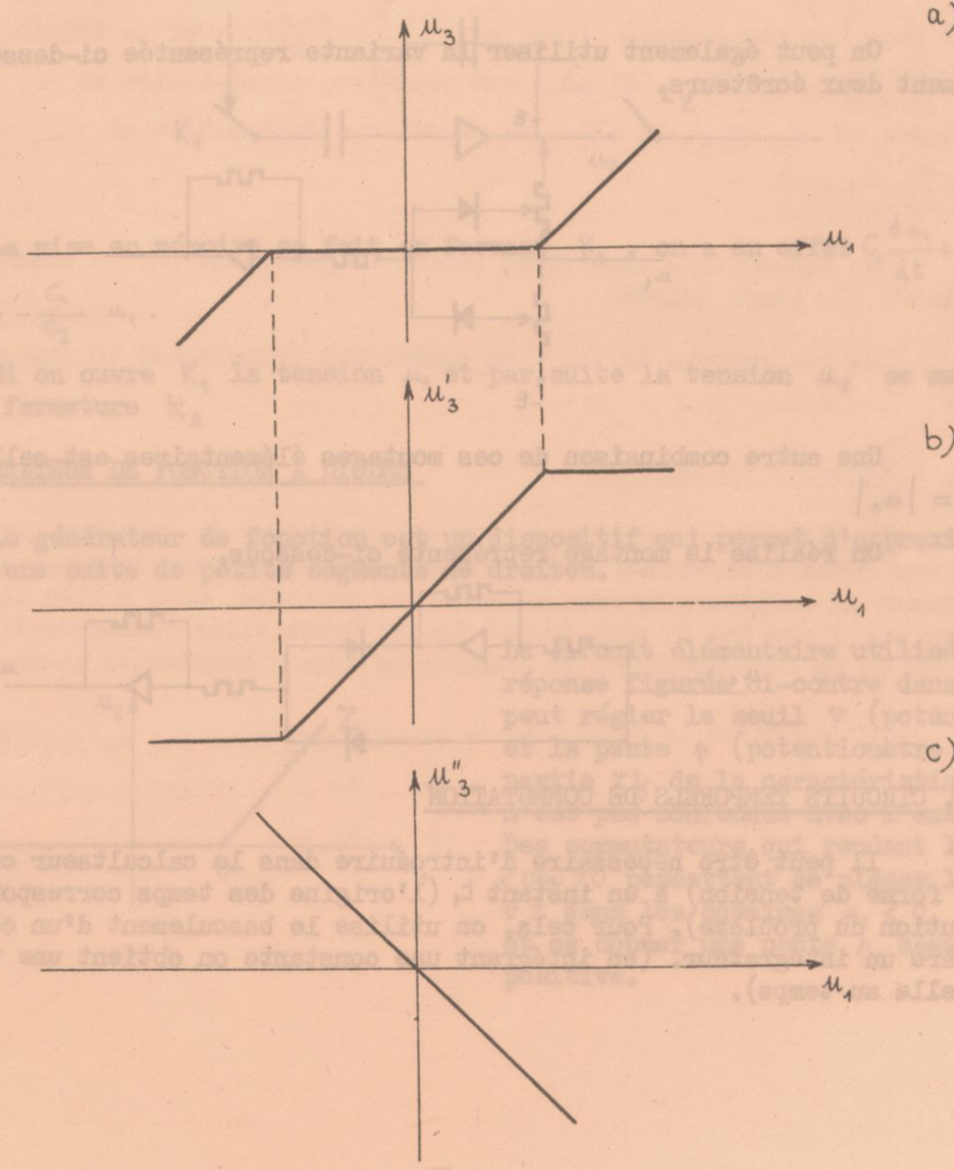
2.3.6. SEUIL

Etablir un seuil, c'est faire correspondre à une tension u_1 une tensio
u_2 nulle si $v'_1 < u_1 < v_1$ et variant linéairement avec u_1 en dehors de cet i
valle.

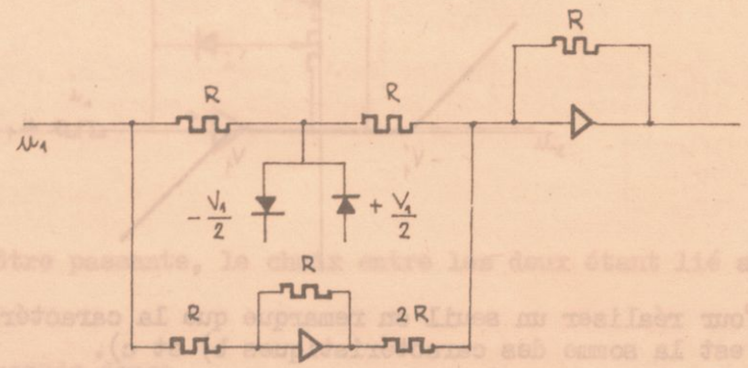
La loi de correspondance est représentée sur la figure ci-dessous.



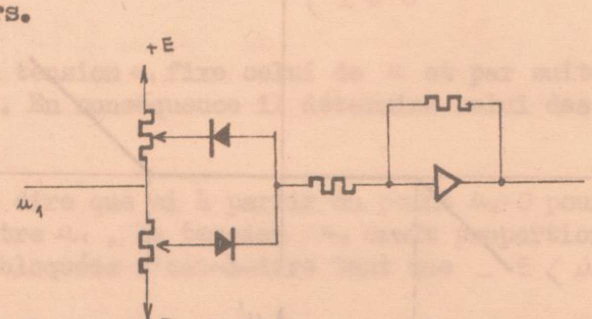
Pour réaliser un seuil on remarque que la caractéristique a) représentée ci-dessous est la somme des caractéristiques b) et c).



Partant de la tension u_1 on réalise u_3 à l'aide d'un écrêteur double e
quant ces tensions à un amplificateur sommateur (qui inverse le signe) on a la t
u_2 . On réalise pour cela le montage représenté ci-dessous.

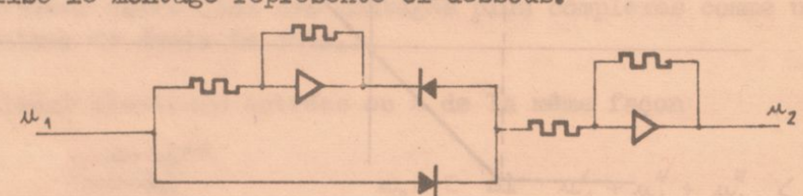


On peut également utiliser la variante représentée ci-dessous qui util
seulement deux écrêteurs.



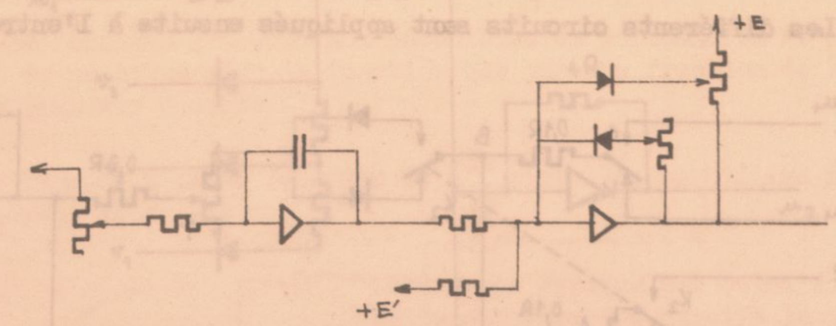
Une autre combinaison de ces montages élémentaires est celle qui donn
$u_2 = |u_1|$

On réalise le montage représenté ci-dessous.



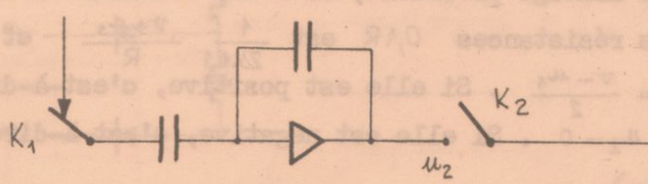
2.3.7. CIRCUITS TEMPORELS DE COMMUTATION

Il peut être nécessaire d'introduire dans le calcultaeur certains sig
(sous forme de tension) à un instant t_1 (l'origine des temps correspond à la mis
résolution du problème). Pour cela, on utilise le basculement d'un comparateur
derrière un intégrateur. (en intégrant une constante on obtient une tension prop
tionnelle au temps).



2.3.8. CIRCUIT DE MEMOIRE

On réalise également des circuits de mémoire avec un changeur de signe capacitif



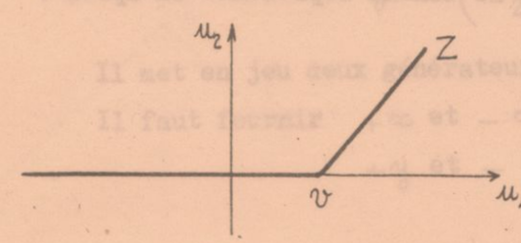
La mise en mémoire se fait en fermant K_1 , on a en effet $C_1\frac{du_1}{dt}+C_2\frac{du_2}{dt}=0$

d'où $u_2 = \frac{C_1}{C_2} u_1$.

Si on ouvre K_1 la tension u_1 et par suite la tension u_2 se maintiennent même après fermeture K_2

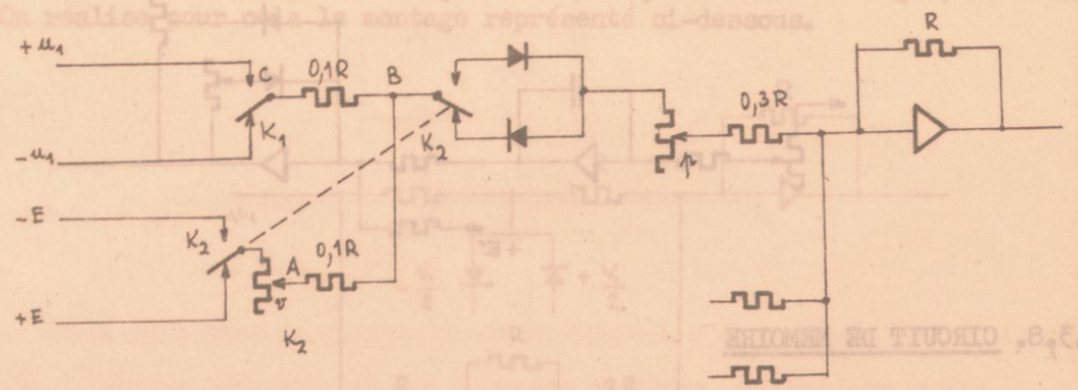
2.3.9. GENERATEUR DE FONCTION A DIODES

Le générateur de fonction est un dispositif qui permet d'approximer une courbe par une suite de petits segments de droites.



Le circuit élémentaire utilisé donne la réponse figurée ci-contre dans lequel on peut régler le seuil v (potentiomètre v) et la pente p (potentiomètre p) de la partie vz de la caractéristique qui n'est pas confondue avec l'axe des u_1 . Des commutateurs qui rendent le seuil $\langle 0$ ou $\rangle 0$ permettent de placer la partie vz dans les domaines $u_1 < 0$ ou $u_1 > 0$ et de donner une pente p négative ou positive.

Les différents circuits sont appliqués ensuite à l'entrée d'un sommate



Sur le montage précédent, on a les tensions v en A et $-u_1$ en C :
courant dans les résistances 0,1R est $\frac{1}{2\times 0,1}\;\frac{v+u_1}{R}$ et la tension en B est
$v - \frac{0,1R(v+u_1)}{2\times 0,1R} = \frac{v-u_1}{2}$. Si elle est positive, c'est-à-dire si $u_1 < v$ la dio
est bloquée et $u_2 = 0$. Si elle est négative, c'est-à-dire si $u_1 > v$, on a

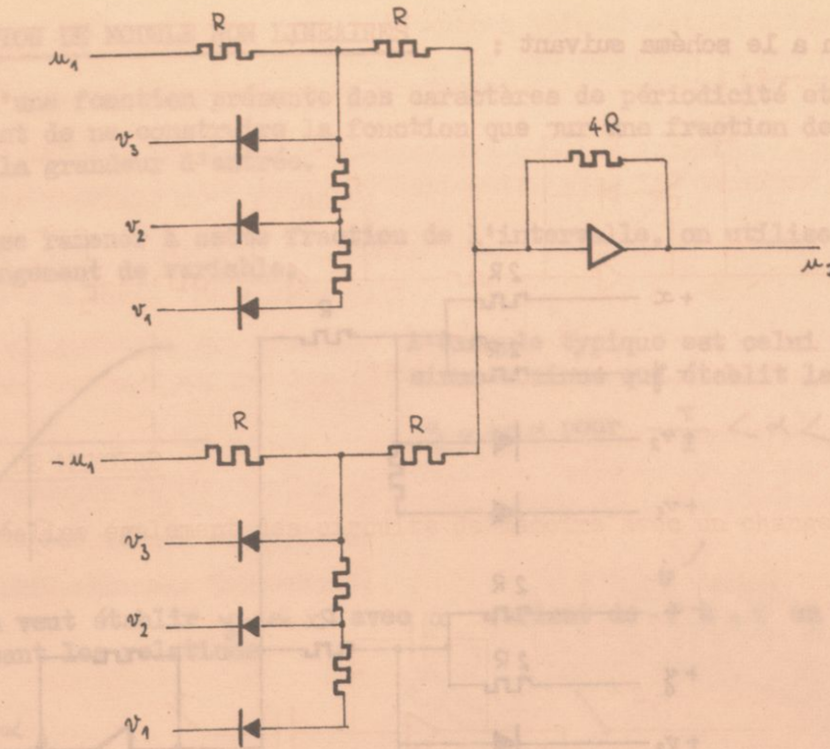
$$u_2 = \frac{1}{0,6}(u_1 - v).$$

Des dispositifs identiques montés en parallèle sur un sommateur permet
de réaliser une ligne brisée.

Les générateurs de fonction comportent normalement 10 dispositifs élém
taires.

2.3,10. GENERATEUR PARABOLIQUE

Le générateur parabolique est un générateur de fonction particulière u
lisant une suite d'écrêteurs et pour lequel on a recherché la disposition des se
qui donnent la meilleure approximation d'une parabole. Ceci a lieu lorsque les p
jections sur l'axe des x sont toutes égales entre elles (démonstration à partir
propriété de la sous-normale). On réalise le schéma représenté ci-après :



$v_1 < v_2 < v_3$

2.3,11. MULTIPLIEUR

Le multiplieur à diodes utilise l'identité

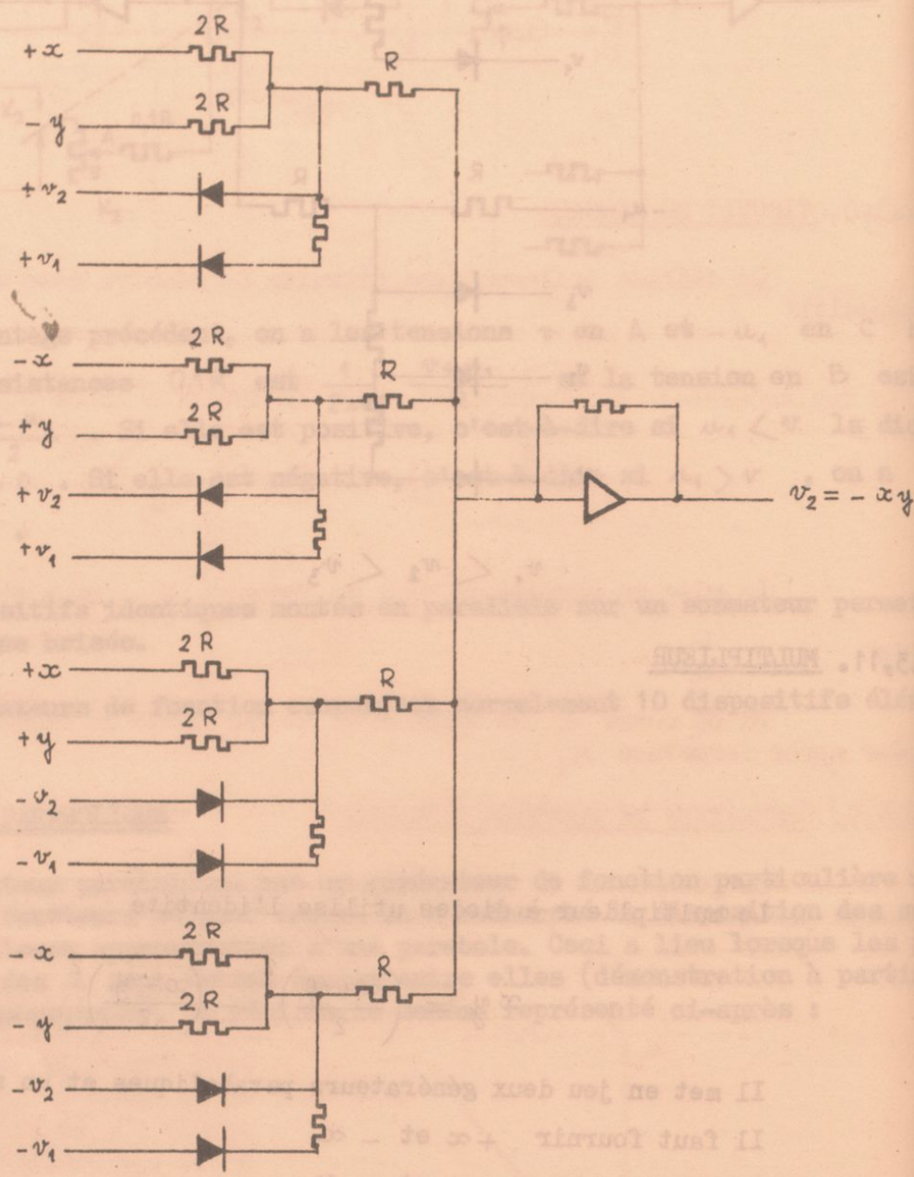
$$xy = \left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2$$

Il met en jeu deux générateurs paraboliques et un sommateur.

Il faut fournir $+x$ et $-x$

$+y$ et $-y$

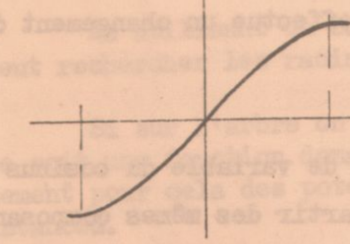
On a le schéma suivant :



2.3,12. ASSOCIATION DE MODULE NON LINEAIRES

Lorsqu'une fonction présente des caractères de périodicité et de symétrie il est intéressant de ne construire la fonction que sur une fraction de l'intervalle de variation de la grandeur d'entrée.

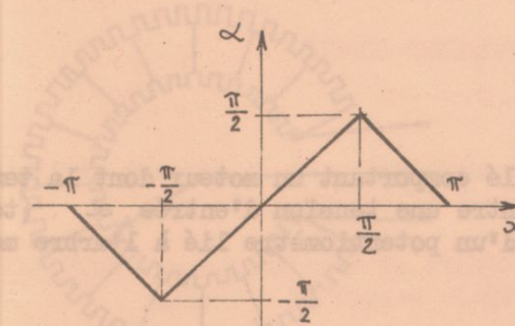
Pour se ramener à cette fraction de l'intervalle, on utilise un module qui effectue un changement de variable.



L'exemple typique est celui du module sinus-cosinus qui établit la relation

$y = \sin \alpha$ pour $-\frac{\pi}{2} < \alpha < \frac{\pi}{2}$

Si on veut établir $y = \sin x$ avec x variant de $-\pi$ à $+\pi$ on adjoint un module établissant les relations

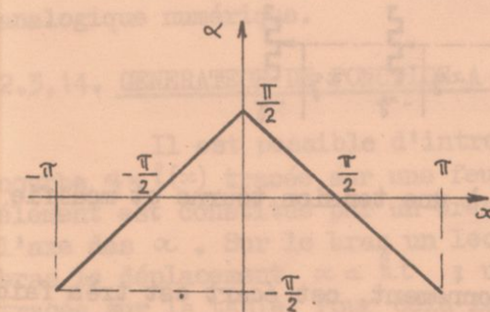


$\alpha = -\pi - x$ si $-\pi \leqslant x \leqslant -\frac{\pi}{2}$

$\alpha = x$ si $-\frac{\pi}{2} \leqslant x \leqslant \frac{\pi}{2}$

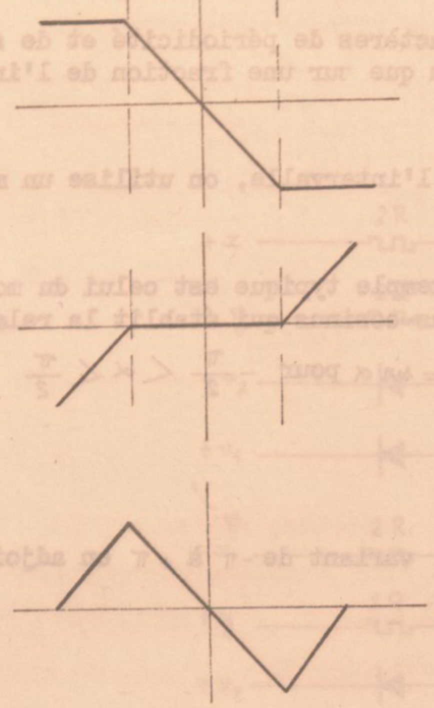
$\alpha = \pi - x$ si $\frac{\pi}{2} \leqslant x \leqslant \pi$

Si on veut établir $y = \cos x$ avec x variant de $-\pi$ à $+\pi$ on adjoint un module établissant les relations



$\alpha = \frac{\pi}{2} + x$ si $-\pi \leqslant x \leqslant 0$

$\alpha = \frac{\pi}{2} - x$ si $0 \leqslant x \leqslant \pi$



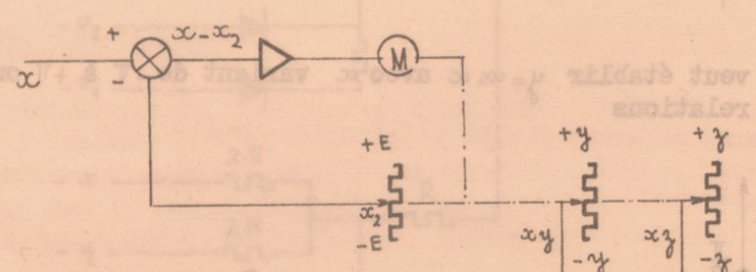
Le module "changement de variable" du sinus est construit par addition de de éléments : un limiteur et un seuil.

Ce module est placé à l'entrée d'un am ficateur qui effectue un changement de signe.

Le changement de variable du cosinus e constitué à partir des mêmes composant (écrêteurs) associés différemment

2.3,13. SERVOMECANISMES

Un servomécanisme est un système bouclé comportant un moteur dont la t d'alimentation est proportionnelle à l'écart entre une tension d'entrée x (d'affichage) et une tension x_2 sur le curseur d'un potentiomètre lié à l'arbre



Tant que $x-x_2 \neq 0$ le moteur soumis à une tension tourne et modifi de façon à annuler $x-x_2$.

Dans les conditions normales de fonctionnement, cet écart est très fai et $x_2 \neq x$.

La position θ de l'arbre de sortie du moteur est proportionnelle à On a $\theta = kx$.

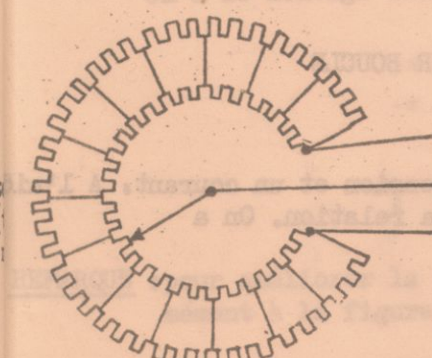
Si sur le même arbre on dispose d'autres potentiomètres alimentés par les tensions $+y, -y$; $+z, -z$; ... les tensions sur les curseurs de ces potentiomètres ont xy, xz ... on réalise ainsi un multiplieur.

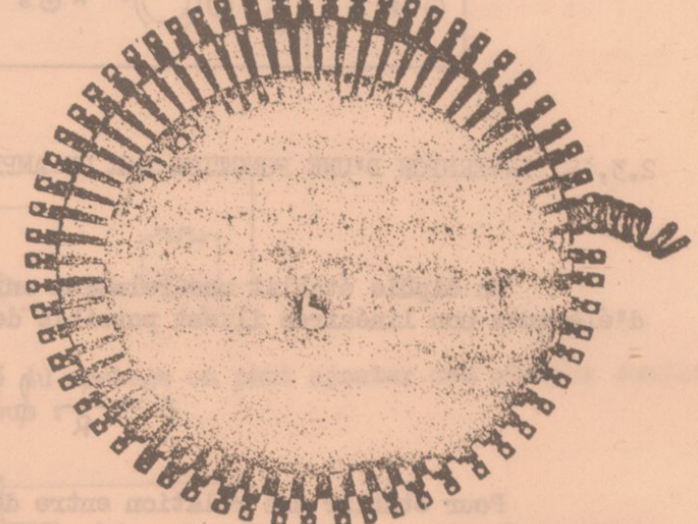
En alimentant le deuxième potentiomètre par les tensions $+x$ et $-x$, n a x^2 . En alimentant le troisième potentiomètre avec les tensions $+x^2$ et $-x^2$ n a x^3

On peut ainsi avoir les puissances successives de x et réaliser la fonction polynome $y = a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x + a_0$.

En utilisant un comparateur qui arrête le calculateur lorsque y s'annule n peut rechercher les racines réelles de l'équation $a_n x^n + a_{n-1} x^{n-1} + \cdots a_1 x + a_0 = 0$

Si sur l'arbre on place un potentiomètre ayant un profil tel que sa résistance soit une fonction donnée de θ , on réalise un générateur de fonction. On emploie également pour cela des potentiomètres à prises multiples entre lesquels on câble des résistances.

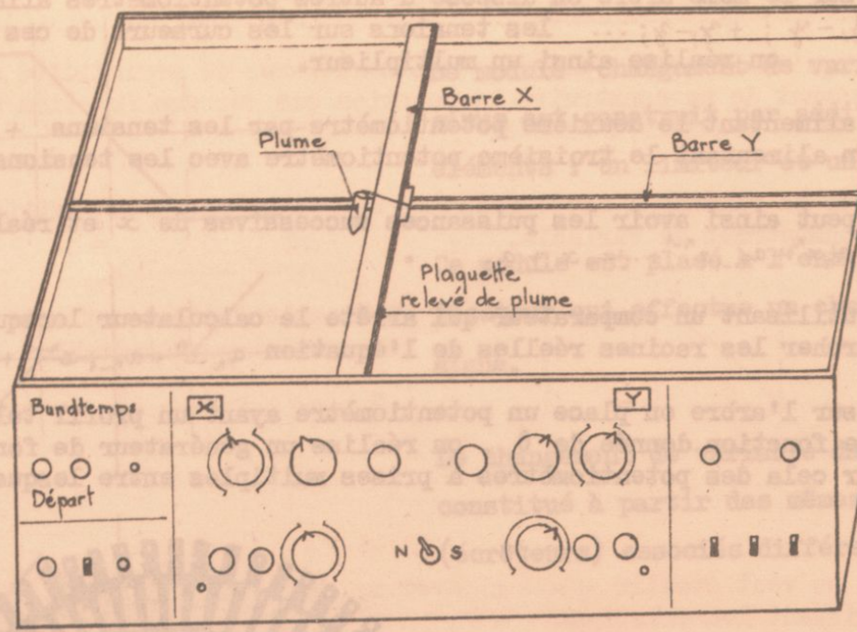




Enfin, en plaçant sur l'arbre moteur un codeur on réalise une conversion analogique numérique.

2.3,14. GENERATEUR DE FONCTION A SUIVEUR

Il est possible d'introduire dans un calculateur une fonction à partir d'une courbe $y = f(x)$ tracée sur une feuille de papier à l'aide d'une table traçante. Cet élément est constitué par un bras qui peut avoir un mouvement de translation selon l'axe des x . Sur le bras un lecteur peut se déplacer selon l'axe y . On donne au bras le déplacement $x = kt$; un servomécanisme fait suivre au lecteur la courbe tracée sur la table. Pour cela la courbe est tracée avec une encre à l'argent qui constitue un conducteur que l'on fait parcourir par un courant haute-fréquence, (ordre de 50 KHz) ce qui produit un champ magnétique. La tête de lecture comprend deux bobines ; l'écart des signaux induits dans les deux bobines est utilisé pour centrer la tête de lecture sur la courbe.

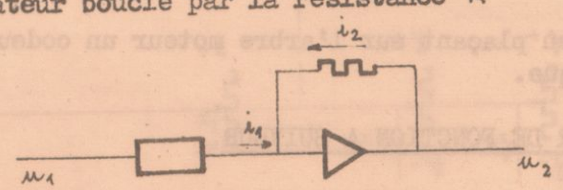


2.3,15. INVERSION D'UNE FONCTION PAR UN AMPLIFICATEUR BOUCLE

Un dipôle établit une relation entre une tension et un courant. A l'a
d'éléments non linéaires il est possible de fixer la relation. On a

$$i = \frac{1}{R}\, f(u)$$

Pour établir une relation entre deux tensions u_1 et u_2 on place l'élé
à l'entrée d'un amplificateur bouclé par la résistance R

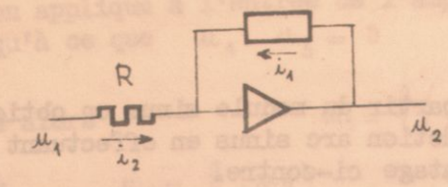


La relation $i_1 + i_2 = 0$

donne $\frac{1}{R} f(u_1) + \frac{u_2}{R} = 0$ ou encore

$$u_2 = -\, f(u_1)$$

Pour obtenir la fonction inverse, on place l'élément dans le bouclag
l'amplificateur



La relation $i_1 + i_2 = 0$

donne

$$\frac{u_1}{R} + \frac{1}{R} f(u_2) = 0$$

c'est-à-dire

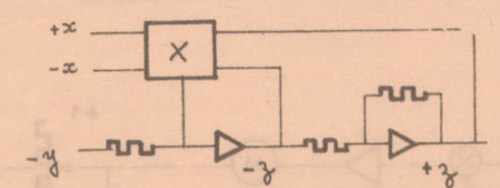
$$u_2 = -\, f^{-1}(u_1)$$

Comme application on peut avoir :

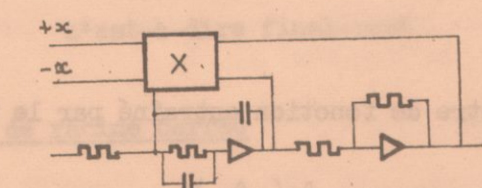
2.3,15.1. Division à partir d'un multiplieur

La méthode précédente est applicable avec un multiplieur à diode mais pour faire apparaître $y = \frac{z}{x}$ à partir de $z = x \cdot y$ il faut disposer de $+x$ et $-x$ ainsi que de $+z$ et $-z$. De plus on doit avoir $|z| < 1$ ce qui impose $|x| > |y|$. Enfin x ne soit pas s'annuler.

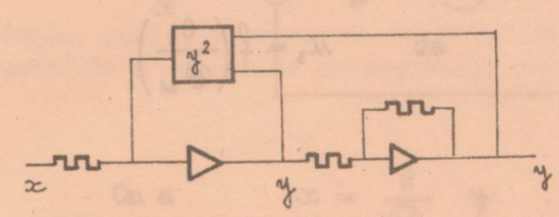
On a le montage suivant



REMARQUE : pour améliorer la stabilité du montage on peut ajouter des réseaux conformément à la figure ci-dessous :

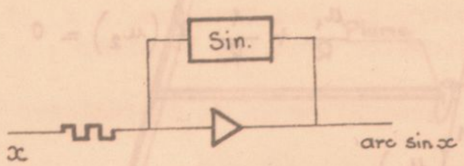


2.3,15.2. Extraction d'une racine carrée à partir d'un module "carré"



Pour faire apparaître $y = \sqrt{x}$ il faut en plus de y qui apparait en sortie de l'amplificateur bouclé par le module carré, diposer de $-y$ que l'on élabore avec un inverseur

2.3,15.3. Fonction arc sinus

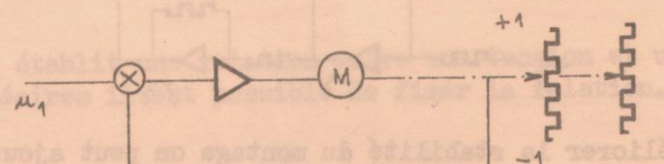


A partir du module sinus on obtient fonction arc sinus en effectuant le montage ci-contre.

2.3,16. INVERSION D'UNE FONCTION PAR SERVOMECANISME

Un servomécanisme constituant un système bouclé, il est comme avec un a ficateur bouclé possible d'inverser une fonction.

L'arbre du moteur d'un servomécanisme étant positionné à un angle θ, obtient sur le curseur d'un potentiomètre alimenté aux extrémités par les tensio -1 et $+1$ une tension $u_2 = k\,\theta$, ou encore si l'excursion du potentiomètre var entre $-\text{Ⓤ}_M$ et $+\text{Ⓤ}_M$ = $u_2 = \dfrac{\theta}{\text{Ⓤ}_M}$



Sur un potentiomètre de fonction entraîné par le moteur on peut produi une tension

$$u_3 = f\left(\frac{\theta}{\text{Ⓤ}_M}\right)$$

Si on applique à l'entrée de l'amplificateur la tension $u_1 - u_2$, le moteur tourne jusqu'à ce que $u_1 - u_2 = 0$

on a alors $u_1 = \dfrac{\theta}{\text{Ⓤ}_M}$ et $u_3 = f\left(\dfrac{\theta}{\text{Ⓤ}_M}\right)$

c'est-à-dire $u_3 = f\,(u_1)$

Si on applique à l'entrée de l'amplificateur la tension $u_1 - u_3$ le moteur tourne jusqu'à ce que $u_1 - u_3 = 0$

on a alors $u_1 = f\left(\dfrac{\theta}{\text{Ⓤ}_M}\right)$ $u_2 = \dfrac{\theta}{\text{Ⓤ}_M}$

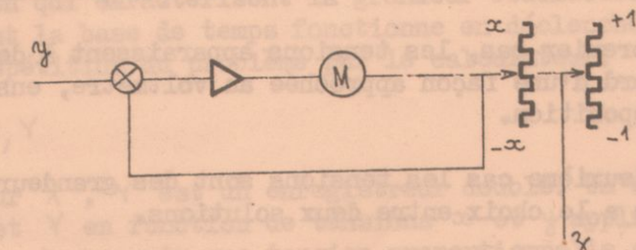
d'où $u_1 = f\,(u_2)$ c'est-à-dire encore

$$u_2 = \overset{-1}{f}\,(u_1)$$

En appliquant ces principes on peut avoir les opérations suivantes :

2.3,16.1. Division

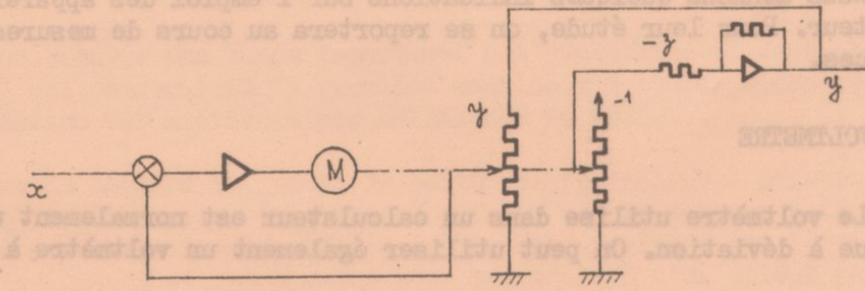
On réalise le montage représenté ci-dessous



On a $y = \dfrac{\theta}{\text{Ⓤ}_M}\,x$ et $z = \dfrac{\theta}{\text{Ⓤ}_M}$

d'où $y = z\,x$ c'est-à-dire finalement $z = \dfrac{y}{x}$

2.3,16.2. Extraction de racine carrée



On a $x = \dfrac{\theta}{\text{Ⓤ}_M}\,y$ et $\dfrac{\theta}{\text{Ⓤ}_M} = -z$ $y = -z$

d'où $x = y^2$ ou encore $y = \sqrt{x}$

2.4. LES ORGANES DE MESURE

2.4.1. INTRODUCTION DES DONNEES

Un problème est caractérisé d'une part par les équations qui le régi (équations algébriques ou équations différentielles) d'autre part par les cond initiales. Celles-ci sont données sous formes de tensions qu'il faut afficher potentiomètres après réalisation du câble d'abord d'une façon approchée à l'ai voltmètre, ensuite d'une façon précise par la méthode d'opposition.

2.4.2. SORTIE DES RESULTATS

Les résultats d'un problème peuvent se présenter différemment selon s'agit d'un problème statique : résolution d'un système d'équations algébrique d'un problème dynamique : résolution d'un système d'équations différentielles.

Dans le premier cas, les tensions apparaissent à des sorties d'ampli On les mesure d'abord d'une façon approchée au voltmètre, ensuite d'une façon par la méthode d'opposition.

Dans le deuxième cas les tensions sont des grandeurs fonction du tem Pour le mesurer, on a le choix entre deux solutions.

1° - Arrêter le problème (mise en mémoire) on est alors ramené au pr précédent. La seule difficulté est la mesure de l'instant auquel le problème a arrêté. Il peut être résolu par l'utilisation d'un intégrateur qui délivre une proportionnelle au temps (intégration d'une constante).

2° - Enregistrement du phénomène. On a le choix entre deux moyens :

a) Enregistrement photographique sur un oscilloscope
b) Utilisation d'un tracteur X, Y.

2.4.3. APPAREILS DE MESURE

Nous donnons quelques indications sur l'emploi des appareils de mesu le calculateur. Pour leur étude, on se reportera au cours de mesures électriqu électroniques.

2.4.3.1. VOLTMETRE

Le voltmètre utilise dans un calculateur est normalement un voltmètr électronique à déviation. On peut utiliser également un voltmètre à affichage rique.

2.4.3.2. METHODE D'OPPOSITION

Cette méthode consiste à opposer à la tension à mesurer la tension a bornes d'un potentiomètre parcouru par un courant constant. L'appareil de zéro constitué par un voltmètre à échelle logarithmique donnent une grande précisio voisinage du zéro.

Il existe deux types de potentiomètres

a) le potentiomètre bobiné constitue par un fil enroulé sur une hélice à 10 tours et sur lequel frotte un curseur.

b) le potentiomètre à décades commandé par un clavier.

Avant de faire une mesure précise il est intéressant d'avoir une idée d'ensemble de l'évolution du phénomène étudié. Pour cela on accélère son évolution en faisant un changement d'échelle des temps par modification de toutes les capacités des intégrateurs et on répète cette évolution par la mise en oeuvre automatique, à un rythme fixé par le calculateur, de la séquense : conditions initiales - mémoire - résolution.

La tension qui caractérisent la grandeur étudiée est alors appliquées à un oscilloscope dont la base de temps fonctionne en déclenché à la fréquence fixée par le rythme de répétition du problème par le calculateur.

2.4.3.4. TRACEUR X, Y

Le traceur X , Y est un enregistreur double. Un stylet se déplace selon deux directions X et Y en fonction de tensions x et y appliquées à deux entrées. L'une des entrées peut recevoir une tension proportionnelle au temps.

La description du traceur (X,Y) a été vue avec le générateur de fonction à suiveur.

Chapitre III

LES CALCULATEURS NUMERIQUES

3.1. PRINCIPES DE FONCTIONNEMENT

3.1.1. ROLE DES MACHINES NUMERIQUES

Les machines numériques sont des systèmes aptes à traiter des ensembles de nombres entiers, c'est-à-dire à faire correspondre à plusieurs ensembles, un autre ensemble qui s'en déduit par des règles logiques.

Dans les calculateurs scientifiques, on traite des nombres qui satisfont aux règles du calcul mathématique mais cela n'est pas une nécessité et les calculateurs peuvent être utilisés pour des problèmes de gestion.

3.1.2. HISTORIQUE

On peut faire remonter l'origine des calculateurs numériques au boulier chinois qui existait 3 000 ans avant Jésus-Christ et qui utilisait le code biquinaire.

Pour trouver une étape importante à l'évolution, il faut attendre Blaise PASCAL (1642) qui construisit la première machine à calculer mécanique. Cette remarquable réalisation fut améliorée par LEIBNITZEN en 1673.

Dans la machine de PASCAL la suite des opérations à effectuer qui constitue le "programme" n'était pas automatiquement faite par la machine mais nécessitait la présence d'un opérateur humain. L'idée de la carte perforée sur laquelle on peut enregistrer les ordres et les données pour l'exécution d'un calcul remonte à JACQUARD qui l'employa pour la commande des métiers à tisser. L'utilisation par une machine "lisant" ceux-ci au fur et à mesure de la réalisation des opérations revient au mathématicien anglais BARRAGE 1802 qui malheureusement ne put faute d'argent jamais aboutir.

Il fallut attendre 1930 pour que le professeur américain Haward AIKE[N] reprenne le problème et réalise en 1944 sa première machine MARK I dans laquel[le] utilisait des relais électromagnétiques et des rubans perforés.

Un nouveau pas fut franchi en 1949 par John VON NEUMANN lorsque celu[i] établit dans la machine même l'utilisation du code binaire au lieu du code déc[imal] et l'introduction globale de l'ensemble des données et des instructions destin[ées à] diriger l'exécution du calcul.

Les progrès importants qui ont été faits depuis portent principaleme[nt sur] la technologie du matériel, en particulier l'emploi de composants à semi-conduc[teurs] et de mémoires magnétiques. Ils sont permis une plus grande rapidité mais surto[ut une] plus grande fiabilité.

3.1.3. BASES DE FONCTIONNEMENT D'UNE MACHINE NUMERIQUE

Un calculateur numérique peut essentiellement faire deux choses :

1° - des opérations logiques :

Il peut reconnaître l'identité de deux termes d'un ensemble, et par c[onsé]quence reconnaître si une proposition est juste ou fausse ; il peut aussi class[er] les éléments d'un ensemble par application de cette règle.

2° - des opérations mathématiques :

Il fait essentiellement des opérations arithmétiques.

A partir d'opérations élémentaires qui sont faites très rapidement, [on] élabore le traitement des grandeurs mathématiques. Il faut pour cela pouvoir di[sposer] à tout moment des éléments de l'ensemble qui sont stockés dans des mémoires. [Ces] éléments doivent pouvoir être transférés dans des registres où ils sont traités.

3.1.4. CODE BINAIRE - SYSTEME BINAIRE

Pour comprendre le fonctionnement d'une machine, nous allons examiner [un] cas simple : l'addition de deux nombres :

Le premier point est de faire correspondre à l'ensemble des nombres e[ntiers] un ensemble de signaux susceptibles d'être construits et traités par la machine.

Ces signaux peuvent être soit une certaine répartition de tensions e[n] sur un ensemble de conducteurs, soit une suite de valeurs de tension échelonnée[s dans] le temps et existant sur un conducteur. Il est de plus nécessaire qu'on puisse [passer] facilement de la forme répartition statique de tensions à la forme répartition [dans] le temps des tensions puisqu'il faut pouvoir d'une part stocker une informatio[n et] d'autre part transférer une information d'un organe dans un autre. Il faut même [que] la correspondance entre ces deux formes soit simple, c'est-à-dire que ces deux [types] de signaux aient la même structure.

Enfin, il faut établir une correspondance entre les signaux compatibles avec la technologie de la machine et les signaux perceptibles par un opérateur humain.

Pratiquement, les signaux compatibles avec la technologie des machines électroniques, les seules que nous étudions ici, sont des tensions électriques qui peuvent prendre plusieurs valeurs. Dans les machines actuellement construites, les valeurs sont au nombre de deux ; la machine fonctionne en binaire ; cela n'est pas une nécessité théorique, mais une facilité pour la réalisation des machines (il existe également machines mécaniques, pneumatiques... dans lesquelles les signaux ne sont pas des tensions).

Le signal élémentaire qui peut prendre deux valeurs est un bit, contraction de binary digit. Une expression binaire constituée par une suite de bits est un mot.

Les signaux accessibles aux opérateurs humains sont des chiffres et des lettres.

La correspondance entre ces deux types de signaux constitue le code. Si celui-ci ne comporte que des nombres, il est dit numérique. S'il comporte également des lettres, il est dit alphanumérique.

Pour définir un code numérique binaire, il faut se donner la loi de correspondance entre la suite des nombres entiers et une suite de signaux constitués par un ensemble de tensions susceptibles de prendre deux valeurs que l'on note 0 et 1.

Le choix d'un code dépend essentiellement du but recherché.

S'il s'agit d'une transmission de données, comme il peut en exister entre un satellite artificiel et un récepteur terrestre, le but recherché est le minimum d'erreur dans l'interprétation de signaux altérés par des informations parasites, le bruit. Ce n'est pas le cas dans un calculateur. Ce que l'on recherche, c'est que le code soit susceptible d'un traitement mathématique, c'est-à-dire essentiellement qu'on puisse définir l'addition. Le code binaire naturel répond à ces conditions, mais il en existe d'autres (par exemple l'Excess 3).

3.1.5. COMPTAGE EN SYSTEME BINAIRE

Il est essentiel de bien distinguer deux concepts différents :

- un système de numération
- un code

Nous supposerons évidemment que tous les deux sont binaires.

On peut donner une image simple du comptage en système binaire de la façon suivante.

Un opérateur produit à des instants aléatoires un signal sonore.

Plusieurs opérateurs sont susceptibles d'appuyer sur un bouton qui a chaque pression modifie l'état d'une lampe : allumée, éteinte, allumée ....

Le premier opérateur appuie à chaque signal sonore. Le second appuie lorsque la lampe du premier opérateur s'éteint, le troisième appuie lorsque la lampe du deuxième opérateur s'éteint et ainsi de suite.

En repérant par le chiffre 0, les lampes éteintes et par 1 les lampes allumées, on a une suite des nombres binaires qui constitue un comptage en numération binaire naturelle

0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Il existe une règle pour passer d'un nombre au suivant lorsqu'on ajou une unité. En conséquence, on peut définir l'addition par apport successif d'un et définir une règle pour obtenir la somme de deux nombres écrits dans ce systè numération.

Le code, c'est-à-dire la correspondance entre la répartition des sig 0et 1 et le nombre, reflète la numération. Il en serait tout autrement si on a affecté à chaque nombre une répartition de 0 et 1 par voie de tirage au sort p exemple. On aurait obtenu là encore un code binaire mais qui aurait été improp faire des opérations arithmétiques.

3.1.6. ADDITION EN SYSTEME BINAIRE

La règle d'addition en système binaire se déduit de la règle élémentaire.

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 avec retenue 1.

Faire une retenue, c'est ajouter un 1 décalé d'une case à gauche à la somme obtenue sans retenue. Cela peut d'ailleurs conduire à une nouvelle retenue et ainsi de suite.

L'addition de deux nombres se présente donc de la façon suivante :

00101011	
+ 00010010	
00111001	Somme sans retenue
00000100	Retenue (décalée)
00111101	Somme

La machine pourrait donc opérer de la façon suivante. On inscrit chacun des nombres dans un registre, c'est-à-dire un ensemble de conducteurs définis par leur rang et dont le potentiel peut prendre deux valeurs représentées par 0 ou 1 (soient A et B ces registres). On inscrit dans un troisième registre C 0 ou 1 selon que les conducteurs de même rang des registres A et B portent soit 0,0 ou 1,1 soit 0,1 ou 1,0.

On inscrit dans un quatrième registre D qui au départ ne porte que des 0 au rang n , la valeur 1 lorsque à ce rang les registres A et B ont la combinaison 1.1.

On décale d'un rang vers la gauche les nombres binaires inscrits dans le registre D.

On transfère le contenu du registre C dans le registre A et le contenu du registre D dans le registre B.

On fait la somme des registres A et B selon le processus qui vient d'être défini et ainsi de suite. On s'arrête lorsque le registre B ne contient plus que des 0. La somme est alors le contenu du registre A.

Cette suite d'opérations semble fastidieuse mais chacune des opérati
est très simple et peut être effectuée rapidement; l'ensemble constitue un cyc
répétitif. Par ailleurs, nous n'avons fait intervenir qu'une suite d'états sta
Or, le temps intervient dans le transfert d'un registre à l'autre. On peut pro
de ce facteur supplémentaire pour diminuer le nombre des registres.

3.1,7. ADDITION ALGEBRIQUE EN SYSTEME BINAIRE

Définir la soustraction revient à définir les nombres négatifs.

Un entier sera maintenant caractérisé par sa valeur absolue comme pr
demment, et par son signe matérialisé par une tension 0 pour le signe +, 1 pou
signe -, dans une case du registre.

Les nombres positifs 00101011 et 00010010 seront maintenant représen
par 0;00101011 et 0;00010010 (le chiffre avant le point virgule représente le

Si dans un nombre A tel que 0;00010010, on change tous les 0 en 1 et
les 1 en 0, on a le nombre négatif B 1; 11101101 appelé complément restreint.

Si on ajoute une unité à ce nombre, on a B' = 1;11101110 appelé compl
vrai. En ajoutant A et B' selon la règle énoncée pour l'addition on a:

```
  0;00010010
+ 1;11101110
------------
  0;00000000
```

On trouve le nombre zéro puisque le dernier chiffre à gauche, n'ayan
de case prévue tombe.

Si on ne considère que les valeurs arithmétiques, on a :

$$A + |B'| = 1\ 00000000$$

Autrement dit, l'opposé de A est $|B'|$ - 1000 00000, Compte tenu qu
dernier chiffre à gauche du nombre binaire 100000000 ne peut être inscrit, ce
revient à retrancher 1 000 000 00 au nombre inscrit dans le registre, on peut,
les opérations algébriques, remplacer l'opposé d'un nombre par son complément

La règle d'addition algébrique est la suivante. On remplace les nomb
négatifs par leur complément vrai et on effectue la somme selon la règle de l'
dition arithmétique y compris le signe. On ne tient pas compte des débordement
gauche du signe.

Donnons des exemples

$A_1 = 0;00101011 \qquad A_2 = 0;000\ 1001$

Le calcul de $A_1 + A_2$ donne

```
 0;00101011
.0;00010010
-----------
 0;00111101
```

Le calcul de $A_1 - A_2$ donne

```
0;00101011
1;11101110
----------
0;00011001
```

Le calcul de $-A_1 + A_2$ donne

```
1;11010101
0;00010010
----------
1;11100111
```

Le calcul de $-A_1 - A_2$ donne

```
1;11010101
1;11101110
----------
1;11000011
```

On peut aussi opérer avec les compléments restreints.

En ajoutant au nombre A = 0;00010010 son complément restreint = 1;11101101

on a :

```
0;00010010
1;11101101
----------
1;11111111
```

On convient que le nombre 1;11111111 représente également zéro.

On a la règle suivante que l'on peut facilement démontrer. Les nombres négatifs étant écrits sous forme de complément restreint, l'addition s'effectue normalement sur tous les chiffres (c'est-à-dire y compris le bit de signe) et s'il y a une retenue du dernier rang à gauche, on la reporte au dernier rang à droite.

3.1,8. COMPARAISON

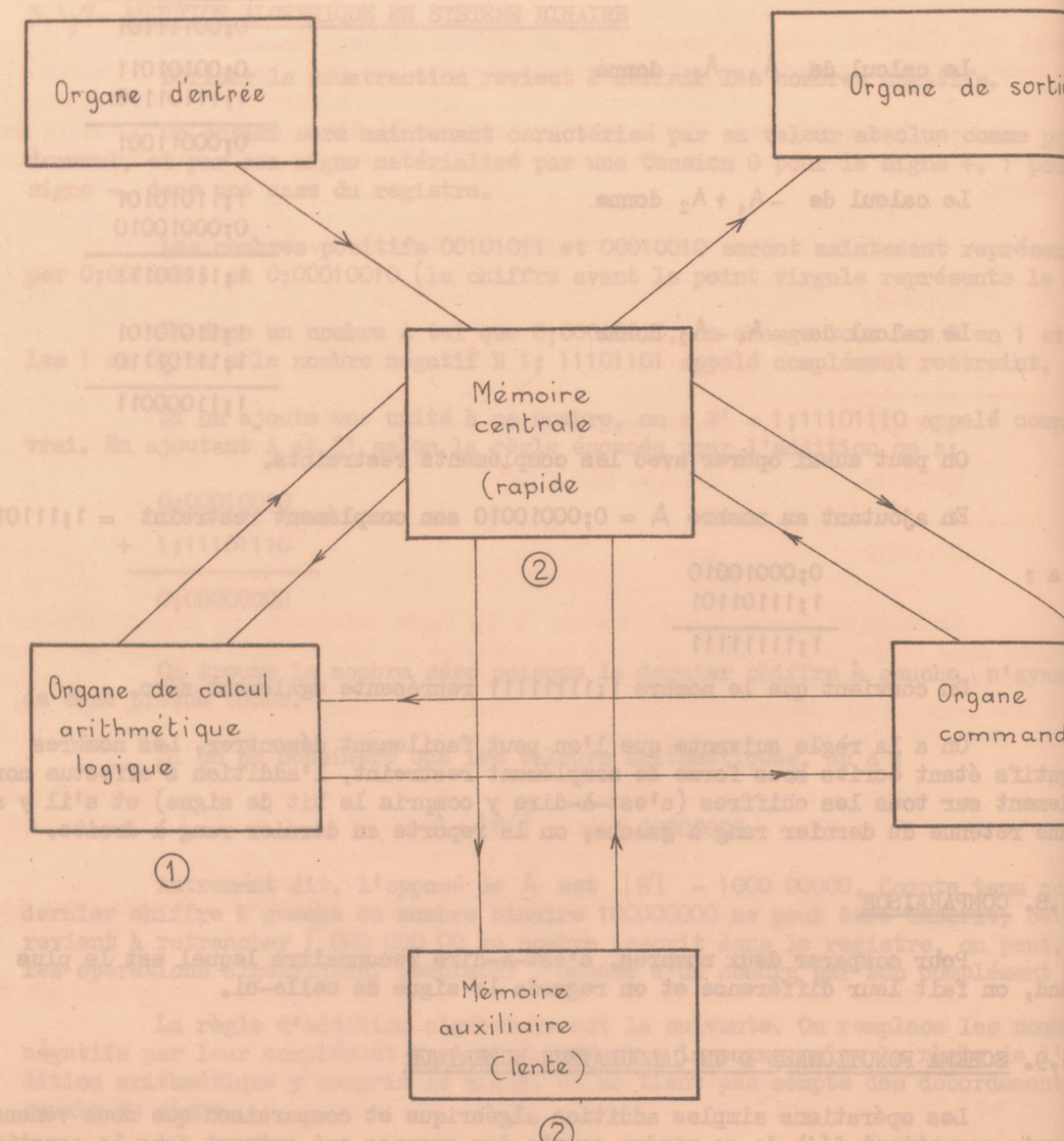
Pour comparer deux nombres, c'est-à-dire reconnaître lequel est le plus grand, on fait leur différence et on regarde le signe de celle-ci.

3.1,9. SCHEMA FONCTIONNEL D'UN CALCULATEUR NUMERIQUE

Les opérations simples addition algébrique et comparaison que nous venons de voir permettent déjà de se rendre compte des organes qui entrent dans la constitution d'un calculateur.

La suite des opérations élémentaires que la machine a à effectuer constitue le programme.

On a le schéma fonctionnel ci-dessous :



ORGANE DE CALCUL

Nous venons sur l'exemple de l'addition algébrique, de donner une idée de la façon dont s'opérait un calcul.

L'organe de calcul permet d'effectuer les opérations arithmétiques et les opérations logiques d'une part en fonction des données et des résultats intermédiaires qui lui sont fournis par la mémoire, d'autre part en fonction des ordres successifs qui lui sont fournis par l'organe de commande.

2. MEMOIRES

L'organe fondamental qui conditionne les possibilités d'un calculateur et par suite son prix est la mémoire. Dans celle-ci on loge, d'une part la suite des ordres, c'est-à-dire le programme, d'autre part les donnnées initiales et les résultats intermédiaires.

Une mémoire est caractérisée par sa capacité et sa rapidité, le prix étant fonction de ces deux éléments. Souvent, on adjoint à une mémoire rapide de capacité limitée, une mémoire lente (mémoire auxiliaire) de grande capacité.

3. ORGANE DE COMMANDE

L'organe de commande orchestre la circulation des nombres et des instructions et l'exécution des opérations conformément aux ordres qui ont été introduits dans le programme et qui sont logés dans la mémoire.

Les différentes opérations se font un rythme défini par une horloge.

4. ORGANES D'ENTREE

Il est nécessaire d'introduire sous forme de code d'une part le programme, d'autre part les données.

Ces éléments sont codés à l'aide d'une machine à écrire spéciale sous forme de rubans perforés ou de cartes perforées. Ils sont ensuite introduits dans la machine à l'aide d'un lecteur (ils peuvent être introduits directement à partir de la machine à écrire).

5. ORGANES DE SORTIE

Le calculateur fournit les résultats sous forme codée. La traduction en écriture courante se fait par une imprimante : machine à écrire ou imprimante rapide.

On peut également sortir les résultats sous forme de courbes établies par une table traçante.

3.2. REPRESENTATION DES NOMBRES

3.2.1. LES NOMBRES

Nous avons vu qu'on peut représenter un nombre entier positif ou négat en système binaire, mais cette représentation est insuffisante. En effet :

a) l'utilisateur du calculateur fournit à la machine des données en sy décimal et veut avoir ses résultats en système décimal. Il est par suite nécess que les organes d'entrées codent les chiffres d'un nombre décimal en un code bi pour leur introduction dans le calculateur. Ce code n'est normalement pas celui système de numération binaire. A l'inverse, les organes de sortie doivent recev les chiffres dans ce même code pour être transformés en chiffres imprimés.

b) un calculateur est amené à traiter des nombres très grands ou très La numération binaire pure ne se prête pas sans modification à l'utilisation de nombres.

3.2.2. NUMERATION BINAIRE

Nous avons vu qu'un nombre binaire se représente par suite de 0 et de Une écriture telle que $a_n . a_{n-1} . a_{n-2} \cdots a_3 . a_2 . a_1$ dans laquelle un terme a vaut 0 ou 1 représente le nombre $a_n 2^n + a_{n+1} 2^{n-1} + \cdots + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0$

Par exemple, l'écriture 10010 représente le nombre $2^4 + 0 + 0 + 2 + 0$.

Cette écriture est dite pondérée. Le terme le plus à gauche à le poids plus fort, le terme le plus à droite le poids le plus faible.

Il résulte de l'écriture précédente que la multiplication par 2^n d'un binaire revient à décaler de n rangs vers la gauche tous les termes et à compl à droite par des zéros.

Partant de la table de multiplication élémentaire

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

ou encore

on déduit la règle de multiplication de deux nombres binaires.

Pour chaque terme du multiplicateur, si celui-ci est 1, on décale le m tiplicande du poids n de ce terme et on ajoute les résultats partiels obtenus

Par exemple pour multiplier 10010 par 101 on a l'opération :

```
   10010
 ×   101
 -------
   10010
  00000
 10010
 -------
 1011010
```

Le produit est :

$$1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

soit en système décimal : $64 + 0 + 16 + 8 + 0 + 2 + 0 = 90$

Pratiquement, un calculateur ne pouvant ajouter que deux nombre à la fois, on fait des additions partielles après chaque décalage.

3.2.3. ECRITURE SEMI-LOGARITHMIQUE DES NOMBRES BINAIRES

L'écriture précédente ne se prête qu'au traitement des nombres entiers positifs ou négatifs si l'on adjoints aux bits d'écriture du nombre, le bit de signe.

Il est possible de l'étendre à la représentation d'un nombre fractionnaire en numération binaire en considérant celui-ci comme une somme de fractions de forme

$$\frac{1}{2}, \quad \frac{1}{2^2}, \quad \frac{1}{2^3}, \quad \frac{1}{2^4} \quad \ldots\ldots$$

Un nombre binaire entier ou fractionnaire peut s'écrire

$$N = a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_1 2 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} \cdots$$

$$N = 2^{n+1} \left[a_n 2^{-1} + a_{n-1} 2^{-2} + a_{n-2} 2^{-3} \ldots \right]$$

Par exemple le nombre décimal 11 s'écrit

$$2^4 \left[\frac{1}{2} + 0 + \frac{1}{8} + \frac{1}{16} \right]$$

Tout nombre binaire se présente donc comme le produit d'une puissance de 2, par un nombre inférieur à 1.

On écrira d'une part la mantisse qui dans l'exemple vaut 1011 en numé
binaire, (écriture du nombre fractionnaire 0,1011) et l'exposant $n+1$ qui dan
vaut 100 (en binaire).

3.2,4. VIRGULE FIXE - VIRGULE FLOTTANTE

Il résulte de ce qui précède que l'on rencontre dans un calculateur d
modes d'écritures des nombres en système binaire.

a) l'écriture en virgule fixe

C'est l'écriture normale des nombres entiers. Elle sert en particulie
l'écriture d'un indice.

On peut l'utiliser pour les nombres fractionnaires mais la position d
virgule est immuable.

b) l'écriture en virgule flottante c'est l'écriture normale des nombr
algébriques.

Le mot qui représente le nombre comporte par suite les éléments suiva

- le signe de la mantisse
- la mantisse écrite en binaire
- le signe de l'exposant
- l'exposant écrit en binaire

Cette écriture permet de cadrer un nombre, c'est-à-dire de l'écrire a
une précision finie en ne gardant qu'un nombre limité de chiffres significatif

La seule limitation dans l'écriture d'un nombre ou très petit ou très
est dans la valeur maximale de l'exposant (positif ou négatif).

3.2,5. DOUBLE PRECISION

Pour disposer d'un nombre suffisant de bits dans l'écriture d'un nomb
on utilise deux mots. On dit alors que l'on écrit en double précision.

3.2,6. OPERATIONS SUR LES NOMBRES ECRITS EN VIRGULE FLOTTANTE

L'addition de deux nombres en virgule flottante nécessite avant toute
un cadrage. Il faut que dans l'écriture les exposants soient égaux. On obtient
résultat en faisant des décalages pour le plus petit. (Si les exposants sont n
et n' il faut décaler la mantisse du plus petit de $n-n'$ rangs vers la droite).
Compte tenu de l'existence possible de retenues, l'exposant de la somme peut ê
différent de n . Il y aura donc lieu de faire un cadrage de la somme.

La multiplication en virgule flottante ne présente pas de difficultés
puisqu'on ajoute algébriquement les exposants et qu'on multiplie les mantisses.

Si on utilise la double précision chacun des nombres est décomposé en deux
éléments, l'un de poids fort (a_1) l'autre de poids faible (a_2). On a :

$$A = a_1 + a_2$$

$$B = b_1 + b_2$$

On calcule $AB = a_1 b_1 + a_1 b_2 + a_2 b_1$ et on néglige le terme $a_2 b_2$

.2,7. NUMERATION DECIMALE ECRITE EN CODE BINAIRE

Pour l'introduction des données dans un calculateur ainsi que pour la sortie
des résultats on utilise un code binaire pour une numération décimale.

Le code est normalement choisi en fonction de la technologie de la machine.
Il peut d'ailleurs être utilisé pour le traitement de l'information si les données
sont nombreuses et les calculs peu conséquents comme c'est le cas en gestion.

Nous donnons quelques exemples de codes.

.2,7.1. CODE DECIMAL DE POSITION

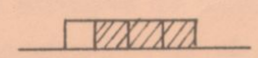
A chaque chiffre d'un nombre décimal, on fait correspondre une case.

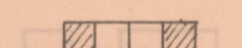
On utilise ce code avec les cartes perforées. La carte comprend un certain
nombre de colonnes où sont inscrits les chiffres de 0 à 9 ; on perfore un chiffre dans
chaque colonne.

On peut utiliser ces cartes pour représenter des lettres en ménageant dans
chaque colonne deux cases supplémentaires dont l'une au moins est perforée. Chaque
colonne comporte alors deux ou trois perforations. On a un code alphanumérique.

.2,7.1. CODE 1, 2, 4, 8

A chaque chiffre, on affecte quatre cases correspondant à des poids 2^0
2^1, 2^2, 2^3

Exemple : chiffre 7 

chiffre 9 

Ce code est surabondant car il permet de représenter les chiffres jusqu'à 15.

Une variante est le code 1, 2, 4 ou code octal qui permet de travaill
en numération à base 8.

3.2.7.3. CODES A 7 BITS

Ces codes utilisent 7 positions binaires. Tout chiffre est représenté
deux bits, ce qui permet à la machine de détecter certaines erreurs.

3.3. TECHNOLOGIE DES CALCULATEURS

3.3.1. <u>SIGNAUX UTILISES DANS LES CALCULATEURS</u>

Un ensemble de signaux élémentaires permet de représenter une express
binaire. Cet ensemble peut avoir deux formes.

a) forme statique.

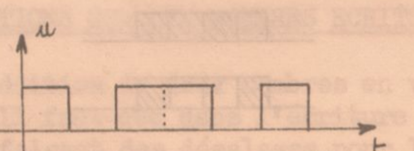
On convient que deux tensions U_0 et U_1 (par exemple $U_0 = 0^{V}$ et $U_1 =$
représentent le 0 logique et le 1 logique. Une suite de conducteurs portés à l'
ou l'autre de ces tensions représente une suite de 0 et de 1.

Si $U_0 < U_1$ la logique est dite positive ce que nous supposerons. Da
cas contraire elle est dite négative.

b) forme dynamique

On produit une impulsion sur un conducteur lorsqu'on fait brusquement
sa tension d'une valeur à une autre, ces deux valeurs représentant l'une le 0 l
l'autre le 1 logique. Selon qu'on passe ou non de la tension la plus faible à l
élevée on a une impulsion positive ou une impulsion négative.

En découpant le temps en intervalle de temps égaux τ (dans un calcul
est de l'ordre de la microseconde), et en appliquant durant ces intervalles de
soit la tension correspondant au 0 logique soit celle qui correspond au 1 log
obtient un train d'impulsions qui représente une expression binaire.



La figure ci-contre représente le
nombre 101101

3.3.2. <u>TRANSFERT DES INFORMATIONS</u>

L'opération fondamentale dans un calculateur est l'addition algébriq
puisque les autres opérations s'en déduisent.

Une opération logique comme celle qui consiste à reconnaître l'identité de
x grandeurs s'y ramène. En effet, les grandeurs peuvent être représentées par des
bres binaires dont la différence doit être nulle.

Pour effectuer une somme de deux nombres binaires rangés dans des mémoires,
les transfère dans un compteur appelé encore registre ou accumulateur. La somme
si effectuée est ensuite transférée dans la mémoire pour libérer l'organe.

Pour effectuer ces transferts, on est amené généralement à modifier la forme
l'information.

Un nombre binaire étant matérialisé par une répartition de tension sur un
semble de conducteurs C, pour transférer celui-ci dans un registre, une première
ution consiste à réunir fil à fil les conducteurs C aux conducteurs homologues du
istre. On fait ainsi un transfert parallèle. Une autre solution consiste à faire
respondre à la répartition statique de tension (forme statique) une succession
impulsions échelonnées dans le temps (forme dynamique). Ces impulsions sont transmises
une seule ligne au registre qui les range de façon à retrouver une répartition
tique de tensions. On fait ainsi un transfert série.

Les opérations de transfert font intervenir le temps. Certaines doivent être
ultanées. Compte tenu de ces impératifs, il est nécessaire d'imposer à la machine
rythme réglé par une horloge. On fixe ainsi le rythme de base, intervalle de temps
tre deux impulsions représentent deux chiffres binaires successifs.

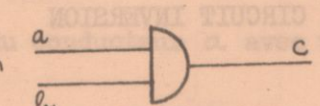
3.3. <u>CIRCUITS DE LOGIQUE</u>

Les organes de traitement sont constitués par des circuits de logiques. Ces
éments font correspondre à une répartition de tensions sur les conducteurs d'entrée
répartition de tension sur les conducteurs de sortie selon une loi qui caractérise
circuit.

Les tensions peuvent prendre 2 valeurs correspondant au zéro logique et au 1
gique.

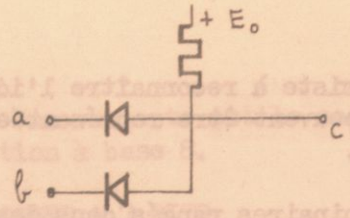
3.3.1 CIRCUIT ET

Notation



La loi de correspondance est la suivante :

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1



On peut réaliser ce circuit avec de
selon le schéma ci-contre.
En dehors du cas où les deux bornes
b sont portées au potentiel E u
moins des deux diodes est passante
potentiel de C est au zéro logique

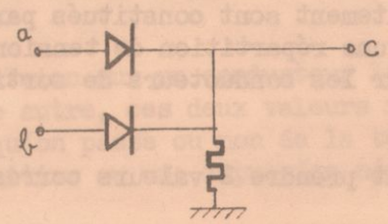
3.3,3.2. CIRCUIT OU INCLUSIF

Notation 

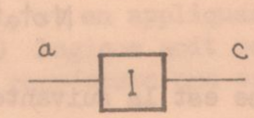
La loi de correspondance est la suivante :

a	b	c
0	0	0
1	0	1
0	1	1
1	1	1

Il est facile de voir avec le schéma ci-dessous que si a ou b (ou deux) est porté au potentiel E une au moins des diodes est passante et le pot de c est égal au un logique (valeur E).



3.3,3.3. CIRCUIT INVERSION

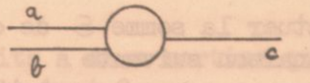
notation 

La loi de correspondance est la suivante

a	c
0	1
1	0

c est encore noté $\overline{a}$

3.3,3.4. Circuit NI

Notation 

La loi de correspondance est la suivante

a	b	c
0	0	1
1	0	0
0	1	0
1	1	0

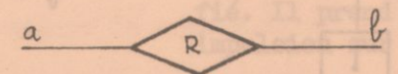
Certains circuits peuvent être réalisés avec des composants passifs non linéaires les diodes, d'autres nécessitent l'utilisation de composants actifs les transistors (les circuits à tubes sont abandonnés).

Les composants passifs ont l'inconvénient de déformer les signaux ; ceux-ci doivent être régénérés au moyen de circuits actifs.

Il existe d'autres circuits que ceux que nous venons d'indiquer mais toutes les fonctions logiques peuvent être réalisées par des combinaisons soit du circuit inversion et de circuits ET ou bien OU soit uniquement à partir de circuits NI .

Ces circuits sont appelés circuits de base.

3.3,3.5. CIRCUIT RETARD

Notation 

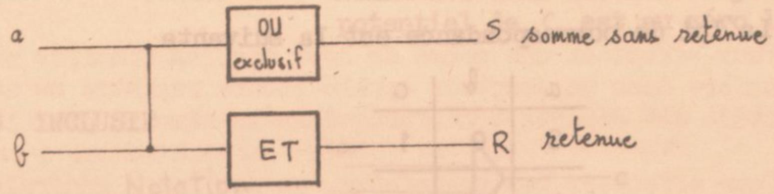
Le conducteur b prend la vzleur logique du conducteur a avec un retard τ

3.3,3.6. CIRCUIT OU EXCLUSIF

Ce circuit est constitué à partir des circuits de base. Il donne la correspondance suivante

a	b	c
0	0	0
1	0	1
0	1	1
1	1	0

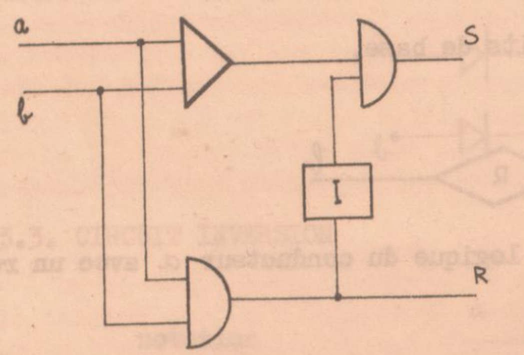
Pour effectuer la somme S de deux nombres binaires a et b avec re
on effectue la combinaison suivante

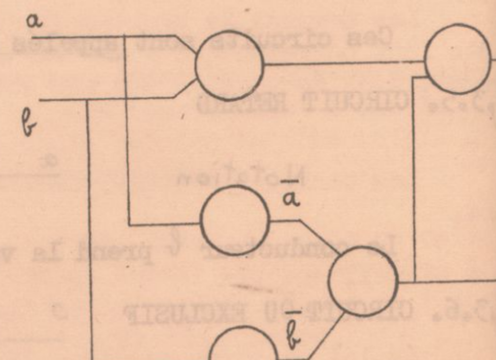


On a en effet les combinaisons suivantes

a	b	S	R
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

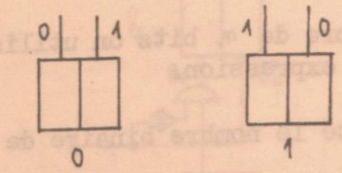
On peut réaliser ces circuits à partir des circuits de base de plusi
façons. Nous donnons deux exemples.





3.3.4. BASCULEURS

Les basculeurs sont des circuits à deux sorties qui peuvent prendre deux états complémentaires, soit 0 et 1, soit 1 et 0

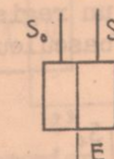


Dans le premier état représenté, on convient que le basculeur représente le 0 logique, dans le deuxième cas, on convient qu'il représente le 1 logique

Pour modifier l'état d'un basculeur on envoie une impulsion sur une entrée. Si par exemple la tension normalement appliquée sur cette entrée est celle correspondant au 0 logique, on applique pendant une durée très brève celle correspondant au 1 logique.

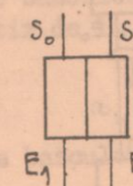
Selon le point où cette tension est appliquée on obtient différents types de basculeurs

3.3.4.1. BASCULEUR SYMETRIQUE A UNE ENTREE



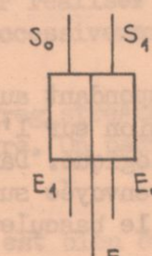
L'état du basculeur change à chaque impulsion envoyée sur l'entrée E

3.3.4.2. BASCULEUR SYMETRIQUE A DEUX ENTREES



Le basculeur prend l'état 0 si on envoie une impulsion sur l'entrée E_0 , s'il était déjà dans l'état 0 rien n'est modifié. Il prend l'état 1 si on envoie une impulsion sur l'entrée E_1 .

3.3.4.3. BASCULEUR SYMETRIQUE A TROIS ENTREES



L'envoi d'une impulsion sur l'entrée E modifie l'état.

l'envoie d'une impulsion sur l'entrée E_0 produit la remise à 0

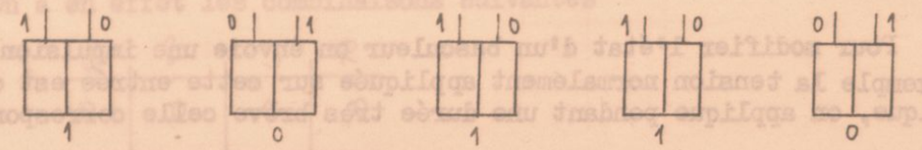
l'envoie d'une impulsion sur l'entrée E_1 produit la remise à 1.

3.3,5. REGISTRE

Un registre est un ensemble de basculeurs permettant de matérialiser
expression binaire.

Si par exemple on veut inscrire un nombre de n bits on utilise n bas
que l'on positionne de façon à représenter cette expression.

Sur la figure ci-dessous on a représenté le nombre binaire de 5 bits



Sur les registres on est amené à faire un certain nombre d'opérations

3.3,5.1. TRANSFERT D'UN NOMBRE D'UN REGISTRE DANS UN AUTRE

Pour transférer un nombre d'un registre A dans un registre B, on ét
à l'aide de portes ET des liaisons entre les sorties des basculeurs du regist
et les entrées des basculeurs du registre B

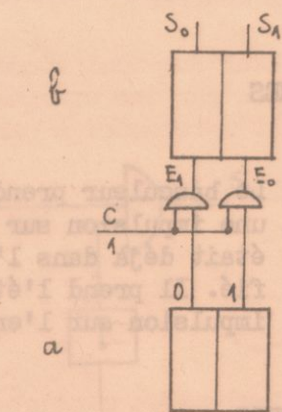


Fig. 1

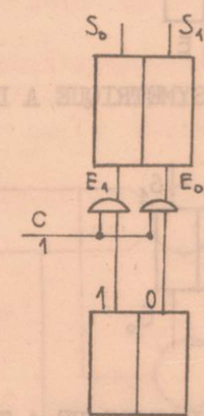
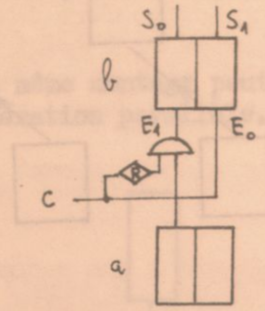


Fig. 2

Dans le cas de la figure 1 une impulsion correspondant au 1 logique ét
envoyée sur le conducteur C produit l'envoie d'une impulsion sur l'entrée E_0 c
positionne le basculeur b, s'il ne l'était déjà, au 0 logique. Dans le cas de
figure 2 une impulsion correspondant au 1 logique étant envoyée sur le conduct
produit une impulsion sur l'entrée E_1 ce qui positionne le basculeur b, s'il
l'était déjà, au 1 logique

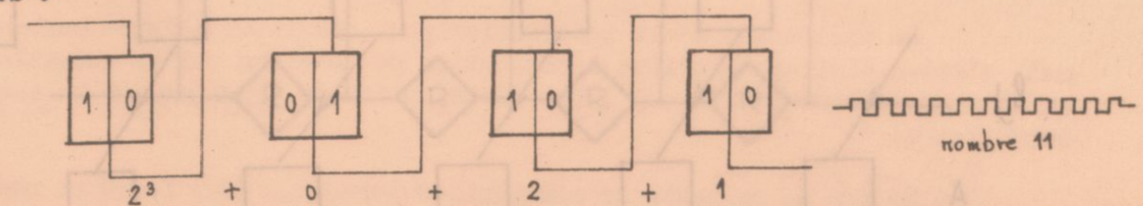
On peut n'utiliser qu'un seul fil entre chaque basculeur en utilisant un circuit de retard.



L'impulsion envoyée sur C remet à zéro le basculeur b, puis retardée elle le met à 1 si le basculeur a est à 1

3.3,5.2. REGISTRE COMPTEUR

Pour compter une suite d'impulsions positives on construit une chaîne de basculeurs :



Le premier (le plus à droite sur la figure) reçoit toutes les impulsions mais transmet une impulsion au suivant seulement lorsqu'il passe de l'état 1 à l'état 0. Autrement dit il ne transmet qu'une impulsion sur deux (c'est un dédoubleur d'impulsions). De même chaque basculeur ne transmet qu'une impulsion sur deux au suivant. Il est facile de voir à partir de la décomposition d'un nombre binaire sous la forme

$$a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0$$

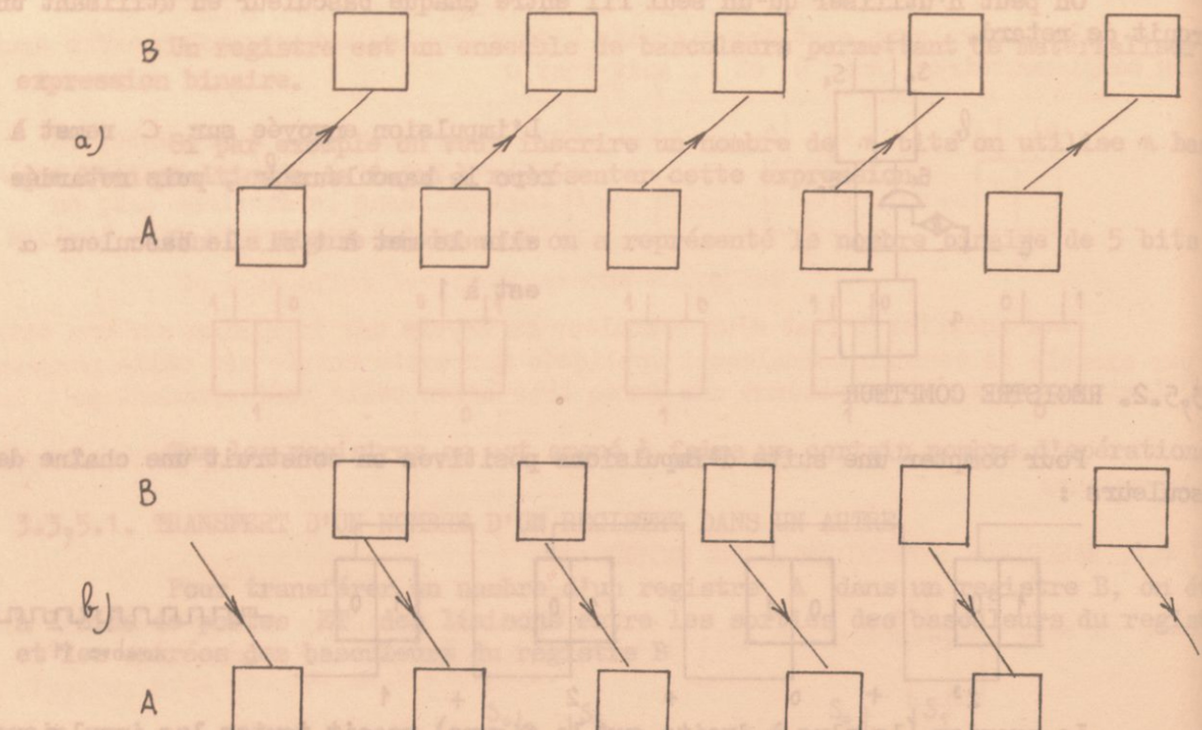
que l'état de l'ensemble des basculeurs représente le nombre d'impulsions écrit en numération binaire.

3.3,5.3. REGISTRE A DECALAGE

Pour réaliser un registre à décalage on utilise deux ensemble de basculeurs et on fait successivement les opérations de transfert représentées en a) et b).

Le registre A contient le nombre qui doit être décalé, le registre B sert d'intermédiaire. On peut évidemment par ce procédé faire des décalages à droite ou à gauche.

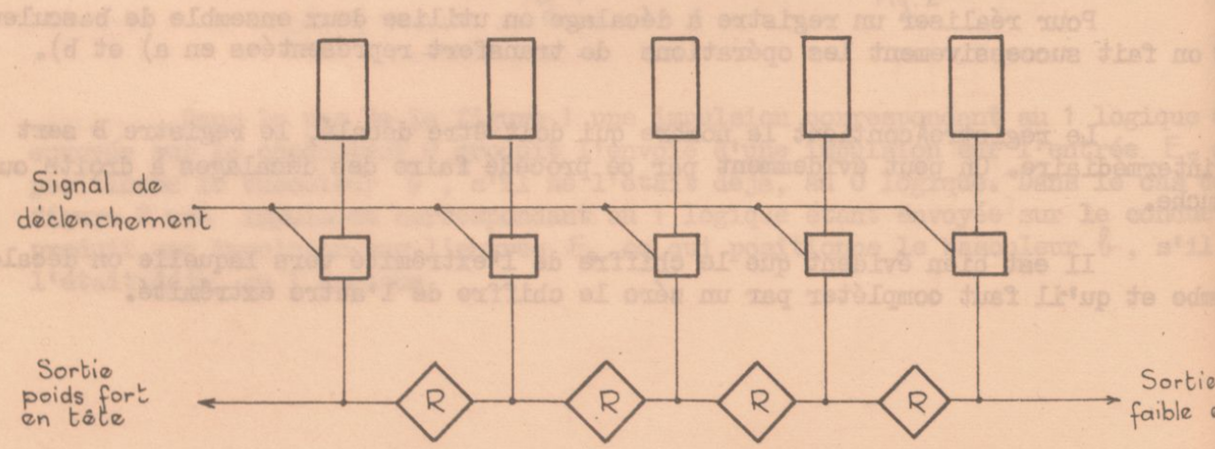
Il est bien évident que le chiffre de l'extrêmité vers laquelle on décale tombe et qu'il faut compléter par un zéro le chiffre de l'autre extrêmité.



3.3.6. CONVERSION DES NUMERATIONS SERIE ET PARALLELE

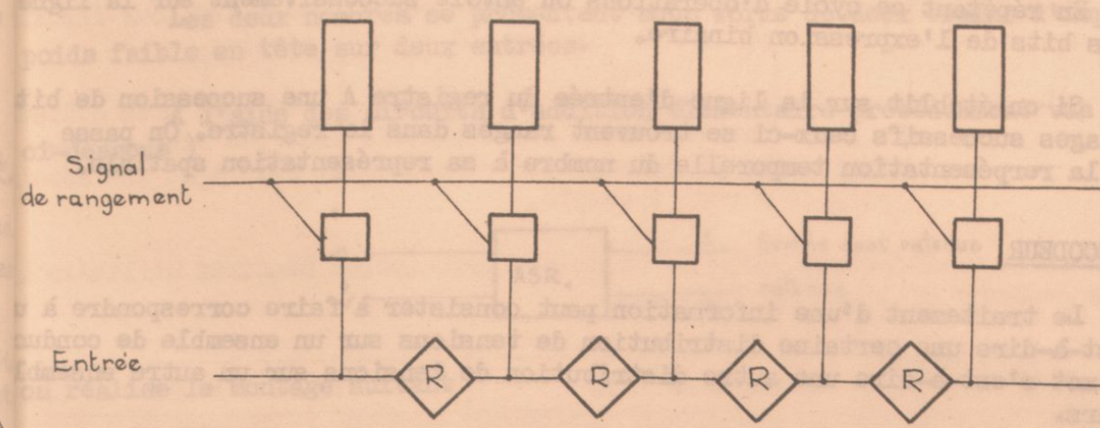
Lorsqu'on inscrit un nombre binaire, on dispose sur les conducteurs cor pondant à une des sorties de chacun des basculeurs qui constituent le registre, d répartition des tensions (0 et 1 logique) correspondant au nombre.

Pour transférer et traiter cette information, il faut faire correspondr à la répartition spatiale des tensions une suite d'impulsions échelonnées dans le (répartition temporelle). On utilise le montage dont le schéma de principe est re ci-dessous. Ce montage est constitué par des basculeurs et des circuits de logiqu



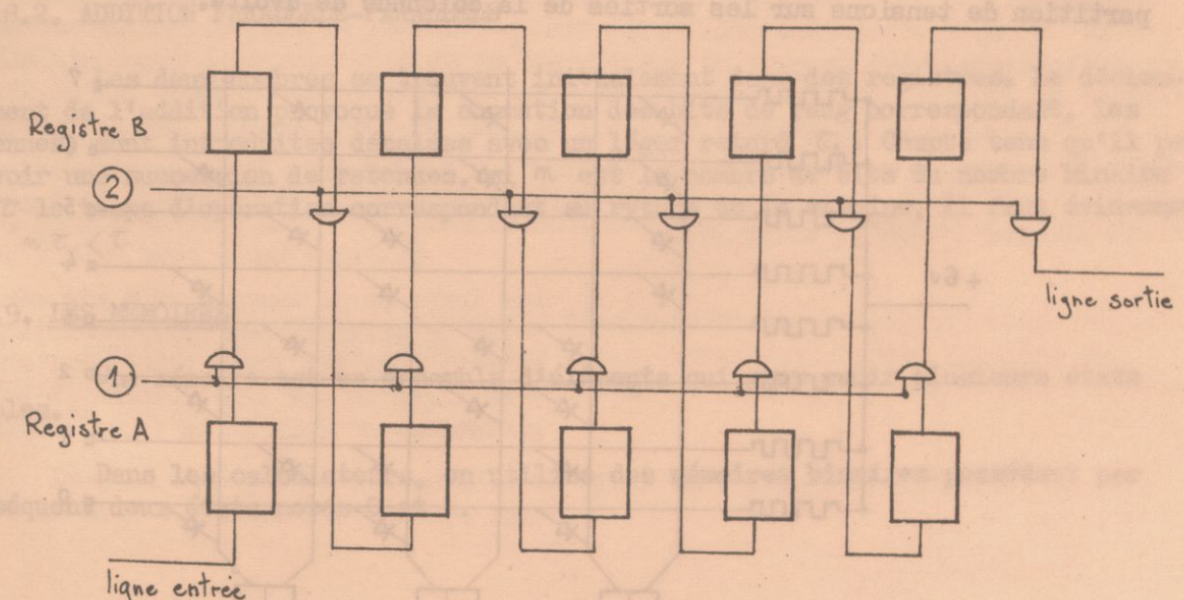
Le signal de déclenchement libère simultanément tous les chiffres inscrits dans le registre, les circuits de retard marqués ⓡ assurent l'échelonnement des impulsions.

Le même montage peut être évidemment utilisé pour convertir une numération série en numération parallèle.



Le signal entrant dans l'exemple représenté ci-dessus, poids faibles en tête se trouve exister sur chacun des conducteurs de l'élément où il doit se ranger au bout du temps $(n-1)\tau$, n étant le nombre de bits et τ le retard d'un circuit de retard. Il suffit alors d'appliquer pendant un temps court (inférieur à la durée d'une impulsion) le signal de rangement.

Le registre à décalage est également utilisé pour faire la conversion des numérations série et parallèle.



Le registre A contient le nombre dont on veut faire la conversion.

Le registre B étant remis à zéro (circuits non représentés), l'envoi impulsion sur les portes commandées par le conducteur 1 transfère dans le regi le contenu du registre A. Après remise à zéro du registre A, l'envoi d'une imp sur les portes commandées par le conducteur 2 transfère avec décalage le conter registre B dans le registre A le bit le plus à droite étant envoyé sur la ligne

En répétant ce cycle d'opérations on envoie successivement sur la li différents bits de l'expression binaire.

Si on établit sur la ligne d'entrée du registre A une succession de par décalages successifs ceux-ci se trouvent rangés dans le registre. On passe ainsi de la rerpésentation temporelle du nombre à sa représentation spatiale.

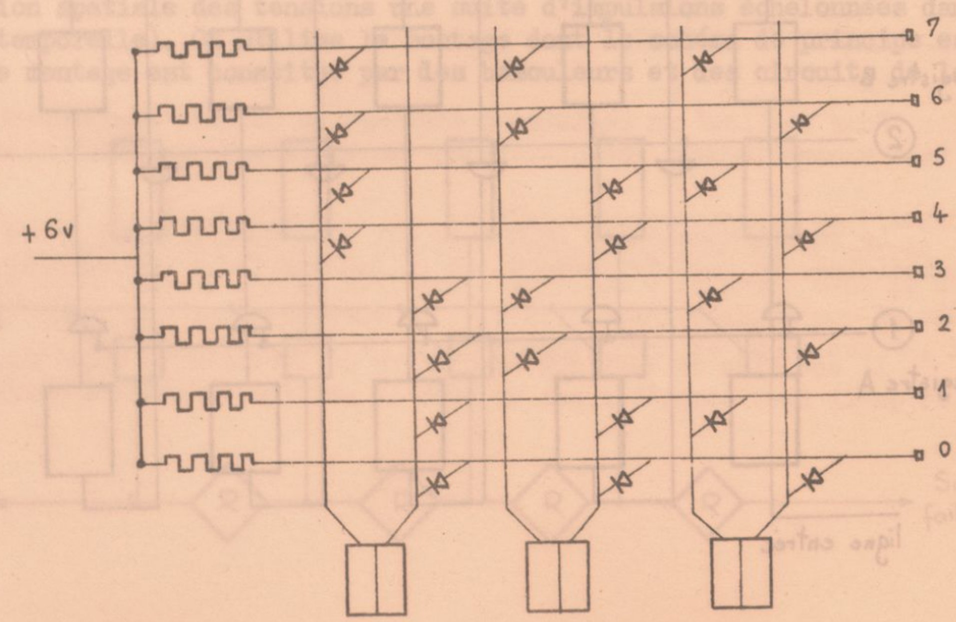
3.3,7. DECODEUR

Le traitement d'une information peut consister à faire correspondre mot, c'est-à-dire une certaine distribution de tensions sur un ensemble de con un autre mot c'est-à-dire une autre distribution de tensions sur un autre ense conducteurs.

Ce cas se présente en particulier lorsqu'il faut faire un changement ou lorsqu'il faut faire correspondre à un ordre l'application de tension sur portes ou encore faire correspondre à une adresse l'accès sur un certain éléme mémoire.

Il existe plusieurs types de décodeurs nous nous limiterons au décod diodes. Cet organe est constitué par un ensemble de portes formant une matrice

La figure ci-dessous représente un décodeur faisant correspondre à binaire matérialisé par les états des basculeurs un code octal matérialisé pa partition de tensions sur les sorties de la colonnde de droite.



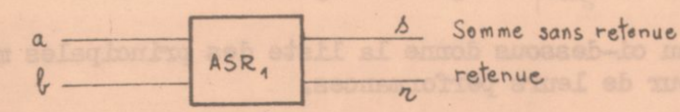
3.3,8. L'ORGANE DE CALCUL

L'élément essentiel de l'organe de calcul est le totalisateur. Il permet l'addition de deux nombres binaires ou l'addition d'un nombre binaire et du complément (vrai ou restreint) d'un autre nombre binaire dans le cas d'une soustraction.

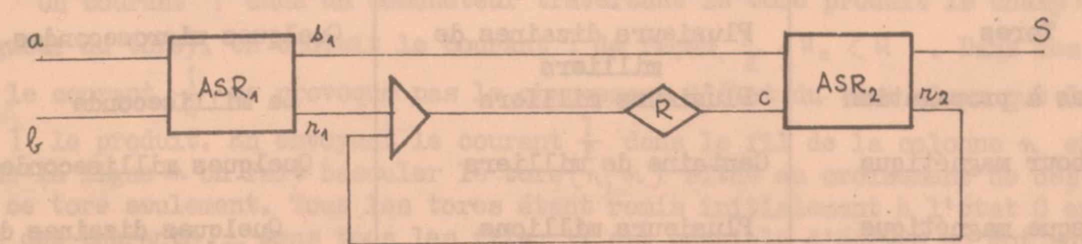
3.3,8.1. ADDITION SERIE-SERIE

Les deux nombres se présentent sous forme de deux trains d'impulsions, poids faible en tête sur deux entrées.

A l'aide des circuits d'addition élémentaire précédemment vus (schéma ci-dessous)



on réalise le montage suivant



On peut vérifier que le cas $r_1 = 1$ $r_2 = 1$ simultanément ne peut pas se présenter.

3.3,8.2. ADDITION PARALLELE-PARALLELE

Les deux nombres se trouvent initialement dans des registres. Le déclenchement de l'addition provoque la sommation des bits de rang correspondant, les retenues, sont introduites décalées avec un léger retard τ_1. Compte tenu qu'il peut y avoir une succession de retenues, si n est le nombre de bits du nombre binaire et τ le temps d'opération correspondant au rythme de la machine, il faut évidemment que $n\tau_1 < \tau$

3.3,9. LES MEMOIRES

Une mémoire est un ensemble d'éléments qui peut avoir plusieurs états stables.

Dans les calculateurs, on utilise des mémoires binaires possédant par conséquent deux états notés 0 et 1.

Une mémoire est caractérisée par les éléments suivants.

1° - __La capacité__ : c'est-à-dire le nombre de bits qui peuvent être stock
Généralement les bits sont groupés en mots. On peut d
également définir une mémoire par le nombre de bits p
mots et le nombre de mots.

2° - __Le temps d'accès__ : c'est-à-dire le temps moyen nécessaire pour insc
un mot ou lire un mot dans la mémoire.

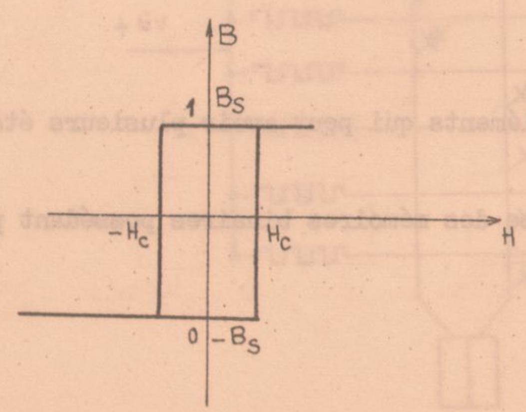
3° - __Le prix__ : Les mémoires rapides et de grande capacité sont très coût
On peut être par suite amené à avoir dans une machine une
mémoire rapide de capacité relativement faible et une mé
lente de grande capacité.

Le tableau ci-dessous donne la liste des principales mémoires utilisées
les ordres de grandeur de leurs performances.

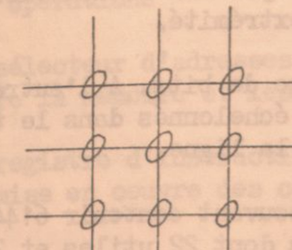
NATURE	CAPACITE EN BITS	TEMPS D'ACCES
Tores	Plusieurs dizaines de milliers	Quelques microsecondes
Lignes à propagation	Plusieurs milliers	La milliseconde
Tambour magnétique	Centaine de milliers	Quelques millisecondes
Disque magnétique	Plusieurs millions	Quelques dizaines de millisecondes
Bande magnétique	Plusieurs millions	La seconde

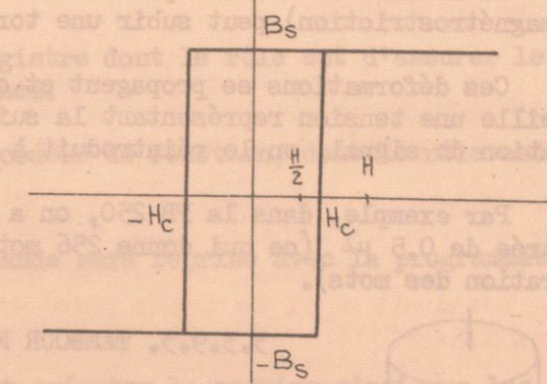
3.3,9.1. LES MEMOIRES A TORES

L'élément de base est le tore de ferrite de très petite taille dont le
cycle d'hystérésis est rectangulaire. Le tore a l'un des deux états de saturation.
ou $+B_s$ (0 ou 1). En faisant passer un courant dans un fil qui traverse le tore
on fait changer l'état de saturation.



Les tores de la mémoire sont mis sous
forme de matrice, des fils conducteurs
passant dans les tores placés sur une
même ligne ou une même colonne. L'enre
gistrement se fait de la façon suivant





Un courant I dans un conducteur traversant le tore produit le champ $H = \frac{I}{\ell}$
ℓ longueur du tore). On choisit le courant I de façon $\frac{H}{2} < H_c < H$. Dans ces con-
itions le courant $\frac{I}{2}$ ne provoque pas le changement d'état du tore alors que le
ourant I le produit. En envoyant le courant $\frac{I}{2}$ dans le fil de la colonne n et dans
fil de la ligne m on fait basculer le tore (n, m) situé au croisement de ces deux
ls et ce tore seulement. Tous les tores étant remis initialement à l'état 0 en
nvoyant des courants $-\frac{I}{2}$ dans tous les fils, il est possible d'écrire un mot sur la
gne m dont le fil est parcouru par le courant $\frac{I}{2}$ en envoyant les impulsions conve-
bles sur les fils de colonne.

Pour effectuer la lecture on utilise un fil supplémentaire qui traverse
us les tores. En envoyant le courant $-\frac{I}{2}$ dans les fils, colonne n ligne m on
cueille une force électromotrice dû au basculement du tore si celui-ci est dans
état 1. Il est à noter que la lecture est destructrice de l'information. Si on veut
conserver il est nécessaire de réinscrire après la lecture.

Pour avoir des mémoires plus importantes, on utilise un empilage de matrices
tore, autant qu'il y a de bits dans un mot. Celui-ci est défini par les coordonnées
m, m) d'un tore d'une matrice.

Du fait que le cycle d'hystérésis d'un tore n'est pas parfaitement rectan-
laire, il apparaît lors de la lecture des f.e.m. parasites. On les minimise en
aborant davantage le câblage mais cela ne change rien aux principes exposés. Donnons
s ordres de grandeur.

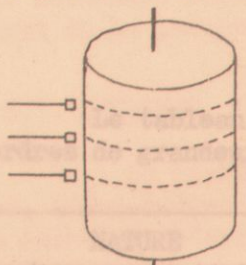
Dans la Γ 60, on a des mots de 24 bits. On utilise un empilement de 24
trices de 64 × 64 ce qui fait 98 304 tores.

3.3,9.2. LIGNE A PROPAGATION (ou mémoire circulante)

La mémoire est formée par un fil de nickel qui par une action électr tique (magnétrostriction) peut subir une torsion à une extrémité.

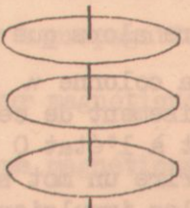
Ces déformations se propagent et on a une suite de bits. A l'autre e on recueille une tension représentant la suite des bits échelonnés dans le tem régénération du signal, on le réintroduit à l'entrée de la ligne.

Par exemple, dans la PB 250, on a des lignes pouvant contenir 6144 i d'une durée de 0,5 μs (ce qui donne 256 mots de 24 bits dont 22 utiles et 2 s la séparation des mots).



3.3.9.3. TAMBOUR MAGNETIQUE

Sur la paroi latérale d'un cylindre tournant autour vertical à quelques milliers de tours par minute, un pellicule d'un matériau magnétique est déposée. Cett cule peut être aimantée localement. Des têtes d'enre trement et de lecture sont réparties sur des vertica mettant l'accès aux pistes circulaires placées sur l



3.3.9.4. DISQUE MAGNETIQUE

Un empilage de disques magnétiques tourne autour d'u vertical. Des têtes d'enregistrement et de lecture q se déplacer radialement se glissent entre les disque permettent l'accès aux pistes.

3.3,9.5. BANDES MAGNETIQUES

Une bande magnétique montée sur un dérouleur peut être aimantée local On a en général 6 à 7 pistes. On peut atteindre une densité de caractères de p centaines au centimètre. On peut lire de 40 000 à 90 000 caractères à la secon

3.3,9.6. CARTES PERFOREES ET BANDES PERFOREES

Ces éléments qui interviennent dans les organes d'entrée et sortie s mémoires qui stockent les informations en dehors de la machine.

3.3,10. <u>ORGANE DE COMMANDE</u>

La suite des opérations que doit effectuer un calculateur constitue programme.

L'organe de commande assure le déroulement de ces opérations. Il com d'une part l'horloge qui fixe le rythme de la machine, d'autre part les organe assurent la progression du programme, l'interprétation et la mise en oeuvre de férents éléments qui le constituent.

On trouve les éléments suivants :

- le compteur ordinal et son registre de réserve dont le rôle est de régler la suite des opérations
- le sélecteur d'adresses avec son registre dont le rôle est d'assurer le transfert entre la mémoire et les autres organes.
- le registre d'instruction et le décodeur de fonction, dont le rôle est d'assurer la mise en oeuvre des opérations.

L'étude de l'organe de commande sera reprise avec la programmation.

3.3,11. <u>L'ORGANE D'ENTREE</u>

La suite des ordres que doit exécuter la machine ainsi que les données numériques sont introduites sous forme de mots binaires à partir de lettres et de chiffres frappées sur un clavier de machine à écrire.

Le support de l'écriture binaire peut avoir plusieurs formes.

1) <u>Cartes perforées</u>

C'est le support le plus ancien. Il a l'avantage de permettre le changement d'une instruction ou d'une donnée en échangeant la carte correspondante.

La carte est généralement formée de 80 colonnes qui sont perforées d'un certain nombre de trous selon un code spécial à la machine. La lecture se fait par des balais frotteurs qui établissent un circuit électrique chaque fois qu'un trou est rencontré.

2) <u>Ruban perforé</u>

Le ruban perforé est un ruban de papier qui porte une suite de séries de perforations selon un code.

La lecture se fait soit à partir d'un lecteur mécanique établissant des contacts électriques soit à partir d'un lecteur optique (éclairement de cellules photoélectriques lorsqu'il y a des trous).

3.3,12. <u>L'ORGANE DE SORTIE</u>

L'organe de sortie fait correspondre à des mots binaires des lettres ou des chiffres, soit sur une feuille de papier, soit sur un écran d'oscilloscope cathodique. Il peut également faire correspondre à la suite des mots binaires un tracé de courbe sur une feuille de papier.

La sortie sous forme de caractère imprimés (lettres et chiffres) se fait soit sur une machine à écrire spéciale, soit sur une imprimante rapide qui frappe simultanément tous les caractères d'une ligne (on atteint 600 lignes à la minute).

3.4. LA PROGRAMMATION

3.4,1. LES INSTRUCTIONS

Un calcul ou plus généralement un traitement d'informations est eff
un calculateur par une suite d'opérations élémentaires. Cette suite d'opérati
réaliser est communiquée à la machine sous la forme d'une suite d'instruction

Deux points interviennent alors.

1° – l'instruction correspond au traitement d'un nombre rangé dans
mémoire. L'endroit où il se trouve placé est défini par une adresse, c'est-à-
un nombre binaire.

L'instruction doit donc comporter une adresse. C'est le nombre plac
cette adresse qui par exemple sera transféré dans un totalisateur.

2° – certaines instructions ont un caractère conditionnel. Selon le
d'une comparaison entre deux nombres, l'instruction qui doit ensuite être exé
est différente. Il est donc nécessaire que l'on puisse appeler les instructio
cessives autrement que dans l'ordre dans lequel elles sont inscrites dans le
gramme.

3.4,2. STRUCTURE D'UNE INSTRUCTION

La structure d'une instruction dépend de l'organisation du calculat
mais toutes les instructions comportent nécessairement un ordre à exécuter qu
appelle encore fonction ou code opératoire. Ce code se représente symboliquem
un groupe de lettres mais il lui correspond un nombre binaire qui sera inscri
la mémoire.

3.4,2.1. INSTRUCTION A SIMPLE ADRESSE

La plupart des machines utilisent des instructions à simple adresse
est par suite constituée de la lettre de fonction à laquelle est adjointe l'a
à laquelle le nombre doit être prélevé dans la mémoire.

Pour effectuer une addition, on opère en deux instructions. La prem
transfère le premier nombre dans le totalisateur préalablement vidé, la secon
transfère le deuxième nombre dans le totalisateur l'addition se réalisant au
et à mesure de l'arrivée du second nombre.

3.4,2.2. INSTRUCTION A PLUSIEURS ADRESSES

Dans une machine à simple adresse l'organe de calcul, c'est alors
un totalisateur, doit effectuer successivement toutes les opérations élément

Pour accroître les possibilités d'une machine on peut avoir un orga
calcul à plusieurs registres ; l'instruction porte alors l'adresse du registr
le nombre doit être transféré.

Une autre organisation du calculateur consiste à transférer simultanément deux nombres. L'instruction comporte alors deux adresses.

3.4,3. RUPTURE DE SEQUENCE

Une possibilité essentielle d'un calculateur électronique est qu'il puisse faire un choix.

Dans le déroulement d'un calcul, on est amené à comparer deux grandeurs et à continuer le calcul d'une façon différente selon le résultat de cette comparaison. Le programme est de ce fait évolutif.

Nous allons prendre un exemple simple qui sera développé dans la troisième partie, celui du calcul des racines d'une équation du second degré $Ax^2+Bx+C=0$. Nous allons ici restreindre le problème au cas où $A\neq 0$, $B\neq 0$, $C\neq 0$ et où on ne recherche que les racines réelles.

L'organisation du calcul est la suivante (voir page ci-contre).

Selon le cas $\Delta \geqslant 0$ où $\Delta < 0$ la suite des opérations effectuées par le calculateur est différente et par suite les instructions. On dit que l'on a affaire à un ordre conditionnel.

Il se peut que, au cours du déroulement du programme, on ait à faire un calcul auxiliaire faisant l'objet d'un sous-programme. Il faut alors sauter des instructions quitte à revenir ensuite à l'instruction qui suit en séquence. On a alors affaire à un ordre inconditionnel.

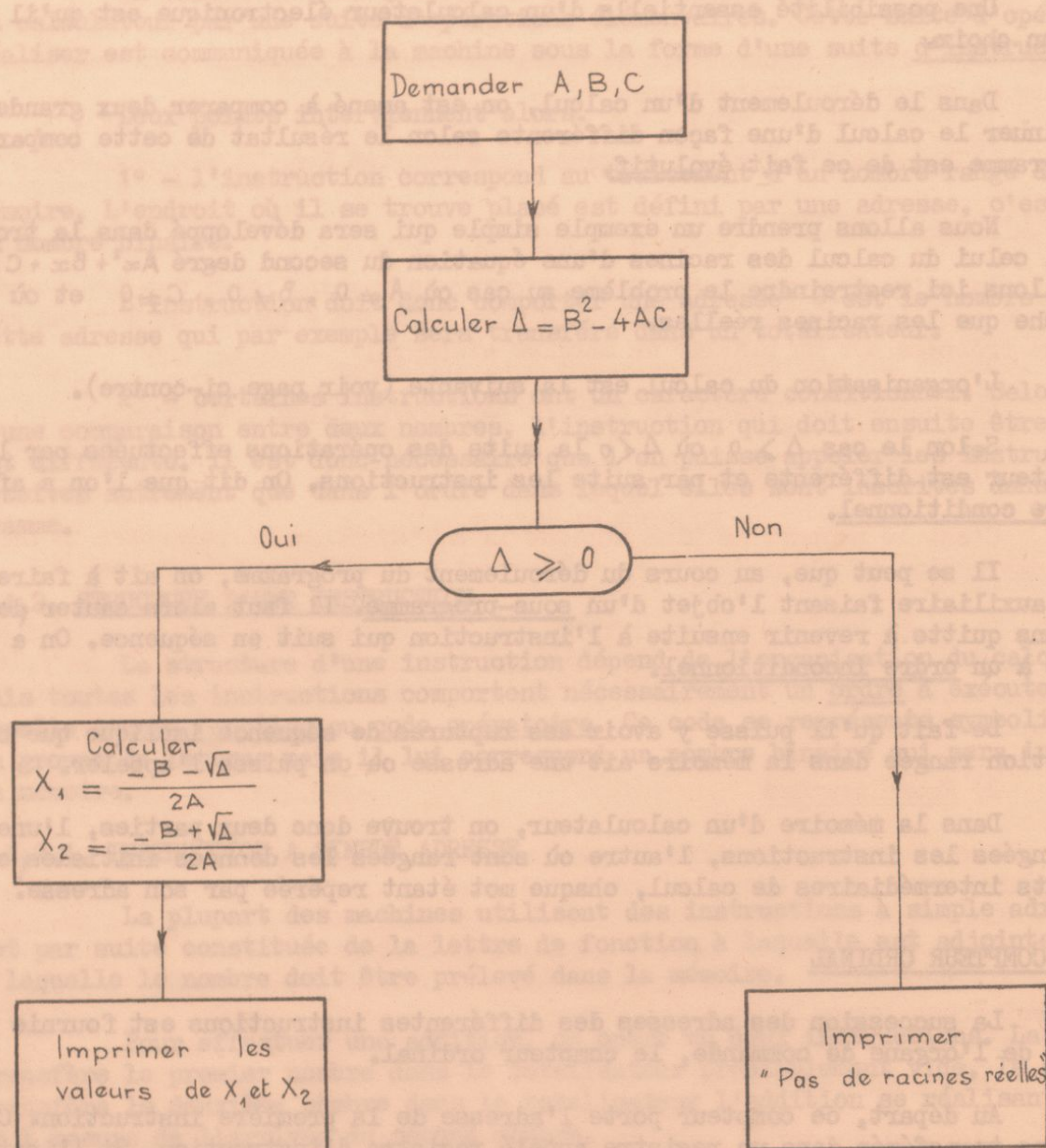
Le fait qu'il puisse y avoir des ruptures de séquence implique que chaque instruction rangée dans la mémoire ait une adresse où on puisse l'appeler.

Dans la mémoire d'un calculateur, on trouve donc deux parties, l'une où sont rangées les instructions, l'autre où sont rangées les données initiales et les résultats intermédiaires de calcul, chaque mot étant repérée par son adresse.

3.4,4. COMPTEUR ORDINAL

La succession des adresses des différentes instructions est fournie par un élément de l'organe de commande, le compteur ordinal.

Au départ, ce compteur porte l'adresse de la première instruction. Celle-ci est alors transférée dans un registre appelé registre d'instruction. Si l'instruction ne comporte pas de rupture de séquence, le compteur ordinal progresse d'une unité et lorsque l'instruction en cours est exécutée, l'instruction suivante est transférée dans le registre d'instruction.



Si l'instruction en cours comporte une rupture de séquence, l'adresse de l'instruction suivante, qui par conséquent ne se trouve pas à la suite, est transférée dans le compteur ordinal. Lorsque l'instruction en cours est exécutée, cette instruction suivante est transférée dans le registre d'instruction.

Dans le cas d'un ordre conditionnel, on réalise un aiguillage. Si le choix ne correspond pas aux instructions qui suivent l'instruction de test, celles-ci sont sautées et le déroulement du programme se continue normalement sans qu'il y ait à revenir à l'instruction qui suit l'aiguillage. Il n'y a donc pas lieu de conserver l'adresse de cette instruction. Au contraire dans le cas d'un ordre inconditionnel, correspondant par exemple à l'adresse N, dans lequel on appelle un sous-programme, il faut se reporter à l'adresse N′ de celui-ci et revenir à l'adresse N+1 lorsque le sous-programme est exécuté. Ce résultat est obtenu en mettant dans le registre de réserve l'adresse N+1 .

3.4.5. MISE EN OEUVRE D'UNE INSTRUCTION

Nous venons de voir que le rôle du compteur ordinal est de contenir les adresses successives des instructions, soit que les adresses se suivent lorsque le programme se déroule en séquence (le compteur progresse alors d'une unité), soit que ces adresses ne se suivent pas lorsqu'il y a une rupture de séquence.

Le transfert de l'instruction de la mémoire dans le registre d'instruction se fait par l'intermédiaire d'un sélecteur d'adresses. Cet organe comporte un registre où est transféré le contenu du compteur ordinal et un décodeur qui, au nombre binaire de l'adresse, fait correspondre des tensions appliquées à des circuits porte de façon à assurer la liaison entre la mémoire correspondante et le registre d'instruction.

L'instruction qui se trouve maintenant dans le registre d'instruction est décodée par le décodeur de fonction.

Si l'instruction comporte un transfert entre une mémoire et le registre de l'organe de calcul, cette opération est faite par l'intermédiaire du sélecteur d'adresses. Cet organe sert par conséquent en général deux fois par instruction.

Lorsqu'une instruction est exécutée, le cycle recommence jusqu'à ce qu'on trouve une instruction d'arrêt qui marque toujours la fin d'un programme.

3.4.6. LE LANGAGE MACHINE

Pour résoudre un problème, il faut écrire une suite d'instructions. C'est le rôle du programmeur. Celui-ci a à sa disposition les ordres machine qui normalement se présentent sous forme de mots binaires et qui constituent le langage interne de la machine. Cette forme étant pénible à manier extérieurement on fait correspondre aux mots binaires des lettres ou groupes de lettres et de chiffres qui constituent un langage extérieur appelé langage machine.

L'introduction des instructions dans le calculateur, se fait donc par l'intermédiaire de lettres et de chiffres codées par l'organe d'entrée. Il faut leur faire correspondre les mots binaires du langage interne. Cette opération est faite par la machine elle-même. On utilise pour cela un programme dit "programme des ordres initiaux" que l'on introduit à la mise en route du calculateur.

3.4.7. LA PROGRAMMATION EN LANGAGE MACHINE

La programmation en langage machine est une affaire de spécialiste.

Le programmateur est obligé de détailler toutes les opérations élémen et de savoir le langage extérieur codé spécifique de la machine. Il utilise un d'instructions différentes allant de la vingtaine pour les calculateurs les plu simples à la centaine pour les calculateurs importants. Il est évident que l'a tation du nombre d'instructions à la disposition du programmeur accroît ses pos bilités mais augmente son temps de formation ainsi que la complexité de la mach

Pour résoudre certaines problèmes, la succession des ordres peut parf être très volumineuse. Certains programmes ont des milliers d'instructions. Leu écriture est longue et fastidieuse, il faut suivre pas à pas le travail de la m organiser les mémoires. Très vite, on a cherché à faire effectuer par la machine partie fastidieuse du travail du programmeur. Dans cette optique, on a établi d langages symboliques.

3.4.8. LE LANGAGE SYMBOLIQUE

Le langage symbolique a pour but de remédier aux deux inconvénients d langage machine, à savoir :

- l'effort pour l'apprendre, effort qui rebute beaucoup d'utilisateurs
- l'organisation des mémoires

On s'affranchit de la sujétion de l'organisation des mémoires en lais soin au calculateur lui-même. On se contente de définir les dimensions des gran c'est-à-dire la place qui doit leur être réservée dans les mémoires.

On remédie à l'emploi d'un grand nombre d'ordres élémentaires en conde ceux-ci dans des ordres beaucoup plus généraux que la machine décomposera ensuit

La machine devient ainsi capable de comprendre des formules mathémati très compliquées moyennant un certain nombre de conventions. Par ailleurs, les rations logiques sont représentées par des expressions dans lesquelles on emplo langue de l'utilisateur : calculer, demander, imprimer, ... (langage Mage) - al si, faire... (langage algol).

Ces langages sont évidemment beaucoup plus simples. Ils permettent de de programmer des problèmes beaucoup plus vastes dans un temps donné. Ils prése néanmoins quelques inconvénients qu'il faut signaler.

D'abord, quand le langage devient très élaboré, il devient à son tour compliqué et nécessite un apprentissage assez long.

Ensuite ce langage a besoin d'être traduit en langage machine. Le cal lateur est alors doté d'un compilateur, c'est-à-dire d'un programme qui effectu traduction.

Enfin, ce langage est moins performant que le langage machine en ce sens u'il occupe davantage de place dans les mémoires ce qui resteint les possibilités calculateur. En effet, les règles de traduction d'un compilateur ont un caractère andard et sont destinées à servir à des problèmes très différents. Elles ne peuvent s par suite, s'adapter parfaitement aux caractéristiques spécifiques d'une question nnée.

Malgré ces inconvénients les langage symboliques apportent une aide considé- ble dans l'emploi des calculateurs. En dehors du cas où le calculateur a toujours effectuer le même programme comme c'est le cas lorsqu'il fait partie d'un ensemble tomatisé, les langages symboliques sont universellement adoptés.

.4.9. LES DIFFERENTS LANGAGES SYMBOLIQUES

Dans le choix d'un langage symbolique, on a à faire un compromis entre deux endances.

1° - Trouver un langage qui soit adapté au mode pensée de l'utilisateur.

2° - Trouver un langage qui soit adapté à la structure du calculateur.

On entend par là que la compilation soit possible, ce qui n'est pas oujours le cas pour les petits calculateurs, qu'elle ne nécessite pas un temps trop ong si elle est possible et que le programme machine que l'on obtient après compilation 'ait pas une longueur exagérée par rapport à celui que l'on aurait en programmant irectement en langage machine.

Les langages les plus utilisés sont les suivante :

.4.9.1. ALGOL (Algorithmic Oriented Language)

Ce langage d'un caractère universel est principalement destiné au traitement es problèmes scientifiques. Il est en effet bien adapté aux processus des raisonnements athématiques.

Conçu en dehors de toute considération relative à une machine spéciale, il ermet des échanges entre utilisateurs. Par contre, sa compilation est très compliquée t n'est pas possible avec certains petits calculateurs.

.4.9.2. FORTRAN (Formuler translation)

Ce langage créé par I.B.M. est très répandu.

Assez commode d'emploi, sa compilation est toutefois laborieuse avec un petit alculateur (on peut être obligé de passer par un langage intermédiaire).

.4.9.3. COBOL (Common Business Oriented Language)

Ce langage est principalement destinée aux problèmes de gestion. De ce fait, es procédures d'entrée de données et de sortie des résultats sont plus simples u'avec ALGOL. Par contre, la formulation d'équations compliquées est peu commode.

3.4,9.4. MAGE (Méthode d'Assemblage de Grande Efficacité)

Ce langage est utilisé avec les calculateurs de la S.E.T.I. (PB 250 Pallas) auquel il est bien adapté.

3.4,9.5. PAF (Programmation Automatique des Formules)

Ce langage est utilisé avec le calculateur CAB 500 construit initia par la SEA puis par la Société BULL - GENRAL ELECTRIC.

3.4,10. <u>MISE EN OEUVRE D'UN PROGRAMME</u>

Dans le traitement d'un problème sur calculateur, il faut distingue plusieurs phases.

3.4,10.1. LE TRAITEMENT MATHEMATIQUE DU PROBLEME

Pour qu'un problème puisse être traité, il faut posséder une méthod matique de résolution.

Les méthodes utilisées dérivent du calcul numérique tel qu'il était avant l'existence des calculateurs électroniques. Leur étude constitue une br des mathématiques appelée l'analyse numérique. Elles conduisent à l'élaborati <u>algorithme</u>.

Un algorithme est une ensemble de règles précises qui définit un pr de calcul destiné à obtenir un résultat déterminé à partir de certaines donné

3.4,10.2. L'ORGANIGRAMME

L'algorithme définit les opérations élémentaires qui constituent le d'un problème. Pour décrire cet algorithme la méthode la plus employée consis tracer un organigramme, c'est-à-dire un schéma de l'enchaînement des opératio

L'organigramme permet d'analyser tous les cas possibles qui peuvent présenter dans l'évolution d'un calcul. Il comporte par conséquent un certai de ramifications à partir d'opérations de comparaison (on dit encore tests).

Si la même suite d'opérations doit se répéter comme c'est le cas lo utilise une méthode itérative de résolution, l'organigramme comporte des bouc

La construction de l'organigramme sera développée dans la troisième

3.4,10.3. LA PROGRAMMATION EN LANGAGE SYMBOLIQUE

Cette opération doit être faite avec une extrême précision, la mach n'admettant aucune erreur dans le choix et l'utilisation de symboles.

4,10.4. LA COMPILATION

Un calculateur est normalement capable de détecter certaines fautes faites ns la programmation. Elle les signale au moyen d'un code qui s'inscrit sur la machine écrire organe sortie. Il y a lieu en conséquence de revoir le programme à cet endroit de reprendre ensuite la compilation. Pour faciliter cette opération de passage en chine, on décompose souvent le programme en éléments qui sont ensuite assemblés.

4,10.5. LA VERIFICATION

Certaines fautes ne sont pas détectées par la machine. Il est donc recommandé effectuer la vérification d'un résultat obtenu à la main ou à l'aide d'un calculateur bureau.

4,10.6. L'EXPLOITATION DU PROGRAMME

C'est la dernière phase du traitement. Il faut veiller soit dans la rédaction problème, soit dans le choix de certaines données qu'elle ne dure pas un temps agéré.

DEUXIEME PARTIE

LE CALCULATEUR ANALOGIQUE NADAC 20

Chapitre I

ORGANISATION DU CALCULATEUR

I.1. GENERALITES

Le NADAC 20 est un petit calculateur analogique à courant continu fabriqué par la S.E.A. (SOCIETE D'ELECTRONIQUE ET D'AUTOMATISME).

La différence essentielle entre les petis (NADAC 20) et les grands calculateurs (NADAC 200) est la présence dans ces derniers d'une enceinte thermostatée contenant les résistances et les condensateurs. On assure ainsi à ces éléments de mesure une précision d'une puissance de 10 supplémentaire, précision que l'on retrouve dans le calcul, au prix d'amplificateurs plus performants. L'amélioration du gain et une meilleure correction de la dérive des amplificateurs permettent d'autre part une augmentation des temps de résolution utilisables (20 minutes sur NADAC 20, quelques heures sur NADAC 200).

Enfin, le nombre plus élevé des organes autorise des problèmes plus importants.

I.2. PRESENTATION DU NADAC 20

Le NADAC 20 se compose :

. d'un bâti contenant les alimentations, le panneau de commande et pouvant recevoir dans deux rangées superposées de douze alvéoles différents modules.

. d'un certain nombre de modules dont le nombre est fonction du problème à résoudre. Un quelconque des 18 types de modules peut occuper n'importe quelle place dans la machine.

. ces modules se connectent par leurs fiches situées à l'arrière sur réseau de câbles transportant les tensions d'alimentation, les barres de masse, lignes nécessaires au fonctionnement du calculateur. On distingue principaleme parmi ces lignes la ligne résolution commandant l'évolution ou l'arrêt, la lig commande de résolution du problème, la ligne de commande de résolution du prob répétitive, la ligne saturation indiquant par le panneau de commande (lampe r une limite de tension de sortie d'un quelconque amplificateur (donc une grande logique trop élevée).

- la ligne mesure renvoyant une grandeur de sortie d'un module (par du bouton mesure de ce dernier), sur le panneau de commande.

- les barres de masse comprennent : deux masses fonctionnelles - poi zéro du problème-points zéro des alimentations.

- la masse de service - retour de courant des relais - la masse de qualité - point zéro des chappers.

. en conséquence on ne trouve sur le panneau avant des modules que l connexions nécessaires au problème à résoudre.

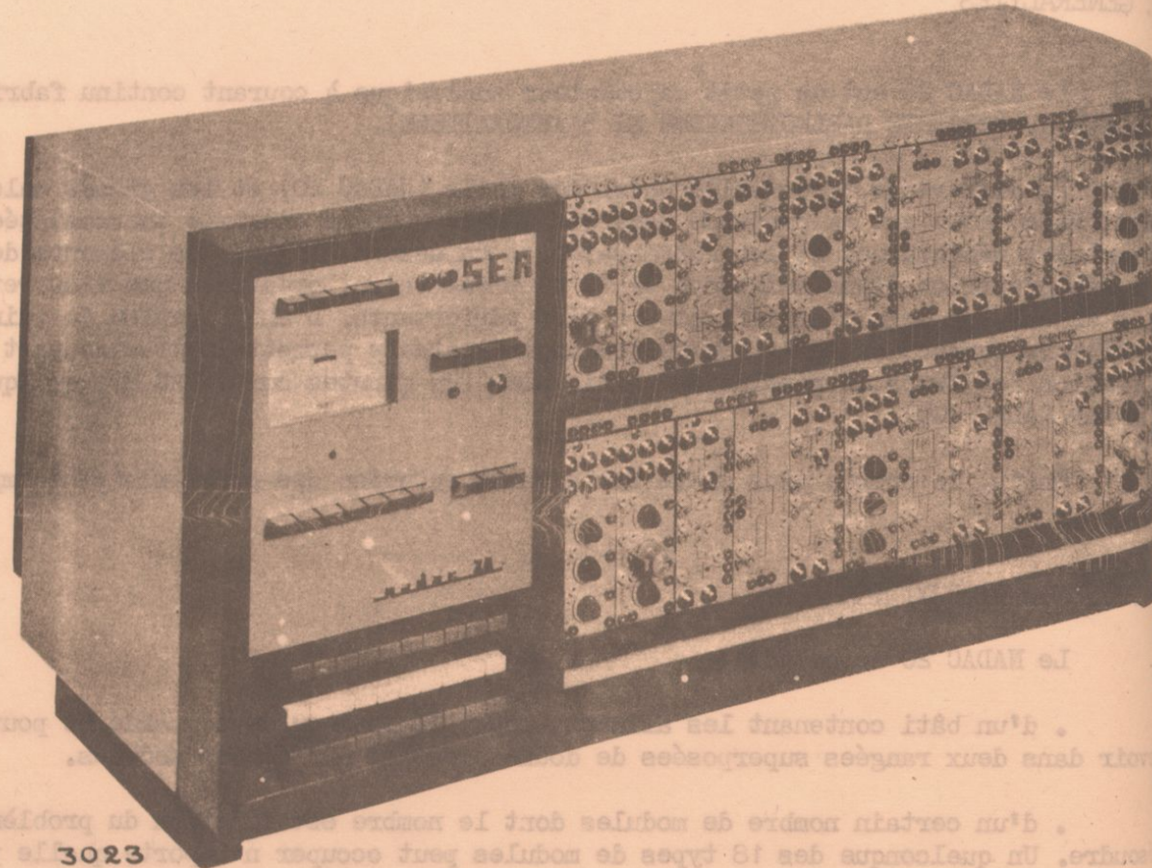


Fig. 1 : NADAC 20 à clavier à décades

Chapitre II

ORGANES DE MISE EN OEUVRE

1. LES ALIMENTATIONS

Derrière la porte constituant le panneau de commande, à gauche du bâti, n découvre 4 tiroirs d'alimentation.

.1,1. <u>TIROIR D'ALIMENTATION GENERALE</u>

Ce tiroir génère les tensions continues et alternatives nécessaires à la achine. La face avant de ce tiroir porte des fusibles et des douilles de mesure. Ces ouilles de mesure (chute de tension des ballasts d'une pré-régulation pour la plupart 'entre elles) sont inutilisables par l'opérateur, elles ne sont utiles qu'au dépan- eur.

.1,2. <u>TIROIR 6,3 V - 400 Hz</u>

Ce tiroir est alimenté par le - 27 V fourni par le tiroir d'alimentation énérale. Il délivre une tension à 400 Hz réglable à 6,3 V (non critique) par un otentiomètre à axe fendu. Cette tension alimente les "choppers" des amplificateurs.

.1,3. <u>TIROIR $\pm$ 25 V et + 50 V</u>

Ce tiroir porte des douilles colorées où apparaissent les tensions réglables éparément mais non mesurables au voltmètre de la machine (calibre 20 V).

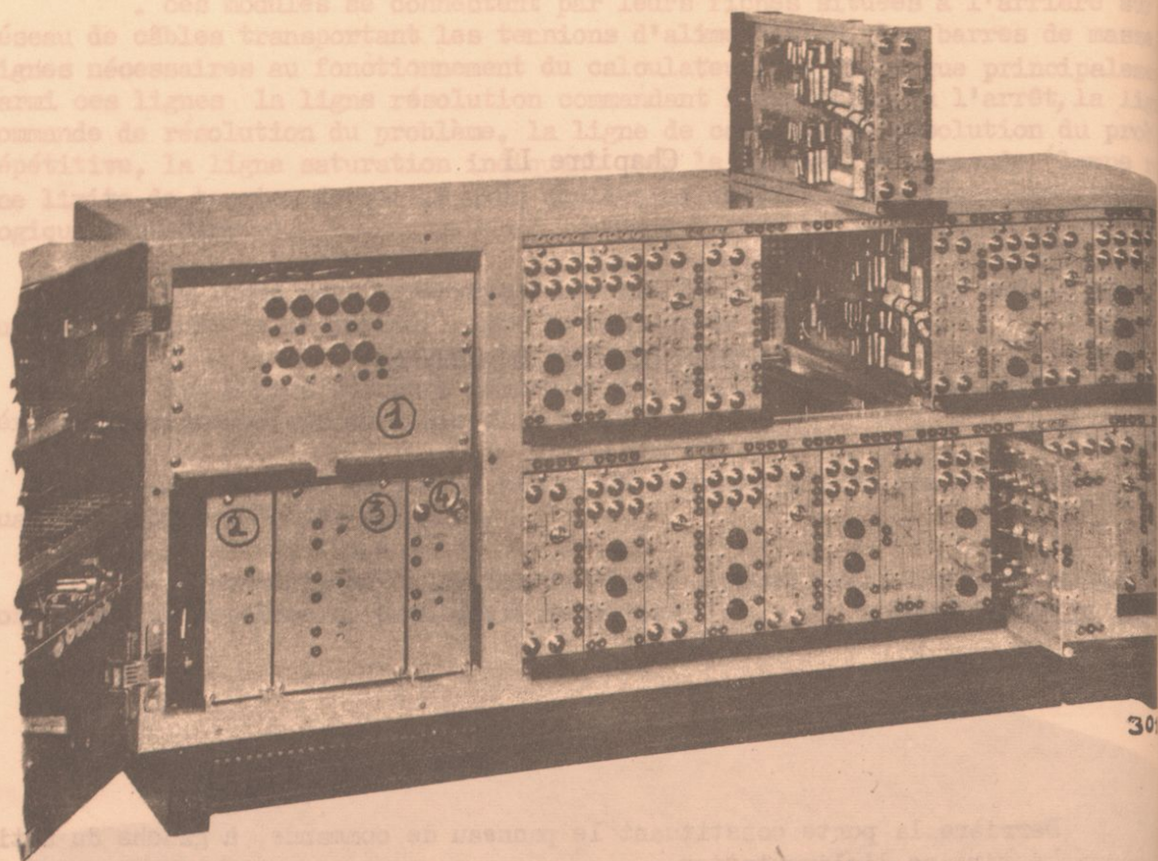


Fig. 2 : Vue de NADAC 20, panneau de commande ouvert

1. Tiroir d'alimentation générale
2. Tiroir 6,3 V - 400 Hz
3. Tiroir ± 25 V et + 50 V
4. Tiroir ± 20 V

2.1,4. TIROIR ± 20 V

Ce tiroir fournit les tensions de référence de calcul ± 20 V avec u maximum de 300 mA et une tension d'ondulation crète-crète inférieure à 1 mV.

Pour réduire les erreurs apportées par les tiroirs utilisants des g teurs de fonctions à diodes (Modules : X2, Multiplieurs, Log. X etc...) les t ± 20 V doivent être réglées à 0,1 V près.

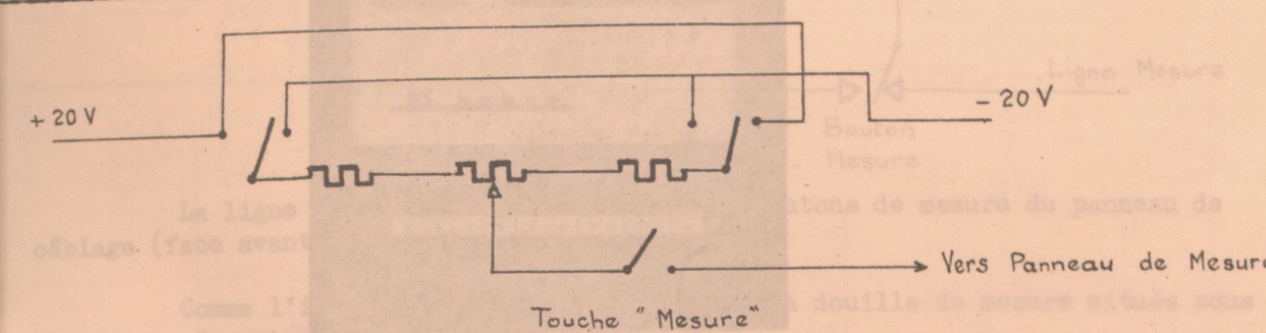
Pour tous les calculs, il est très important d'avoir le + et le - 2 égaux en valeur absolue à mieux que 10^{-3}.

Ce réglage appelé "symétrie référence" s'obtient en deux temps :

a) Réglage du symétriseur : la touche 2 V du clavier "mesure" étant enfoncée, on règle le potentiomètre de symétrie (situé à l'intérieur du tiroir le plus près de la face avant) de manière à obtenir une même déviation pour la position + 20 V et la position - 20 V du commutateur du tiroir.

b) Réglage d'égalité des tensions : les touches 20 V étant enfoncées simultanément, (le calibre devient alors + à - 100 mV), on annule la déviation en agissant sur le potentiomètre à axe fendu du -20 V.

SCHEMA DU REGLAGE "SYMETRIE"



Ces réglages sont à faire tous les mois au minimum.

REMARQUE

A l'issue de ce réglage, il est important de ramener en position normale les touches 20 V et 2 V du clavier de mesure.

2.2. PANNEAU DE COMMANDE

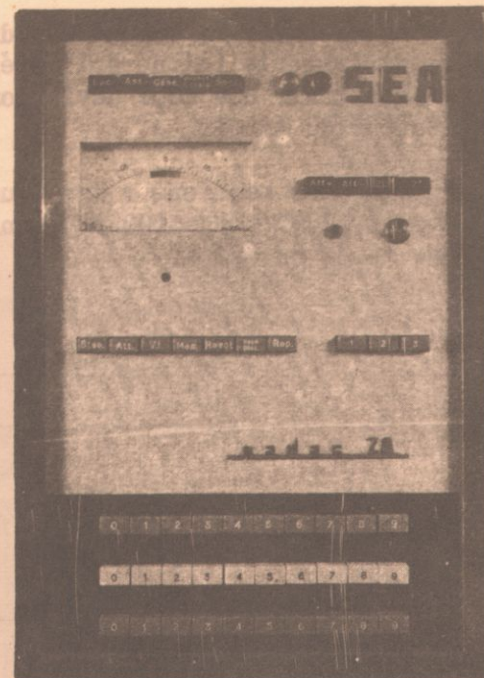
Le panneau de commande permet la mise en route de la machine, le départ, l'immobilisation, l'arrêt du problème ou sa résolution cyclique.

Il comporte deux voyants lumineux, quatre claviers, un voltmètre et un potentiomètre.

2.2,1. VOYANTS LUMINEUX

L'allumage de la lampe blanche indique la mise sous tension de la machine.

La lampe rouge indique soit un défaut d'une quelconque alimentation, soit une saturation d'un des amplificateurs. Dans ce dernier cas, l'allumage de la lampe rouge s'accompagne du passage de la machine en position "mémoire", il appartient alors à l'opérateur, de rechercher l'amplificateur ayant saturé ; une signalisation individuelle de la saturation d'un amplificateur serait préférable.



PUPITRE DE COMMANDE
AVEC CLAVIER A DÉCADES

2.2,2. CLAVIER DE MISE EN ROUTE

Ce clavier se trouve en haut du panneau de commande, Il comporte cin touches repérées :

"L O C " - commande locale, enfoncée si un seul NADAC 20 est en serv

"A S S" - commande asservie, enfoncée sur la machine asservie à une machine mère.

"G E N E " - commande générale, enfoncée si le NADAC 20 est la machine

"OUV. LOC." - ouverture locale, enfoncée pour libérer le NADAC 20 de la machine mère et pour pouvoir le commander de son propre c

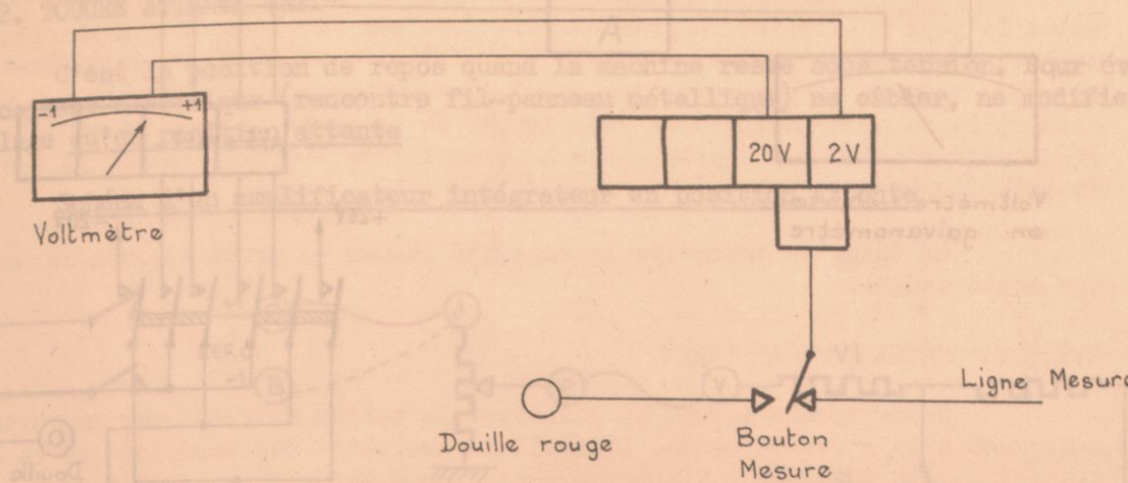
"S E C T" - secteur, enfoncée pour la mise sous tension.

2.2,3. CLAVIER DE MESURE

Ce clavier situé à droite du voltmère, comporte quatre touches ainsi repérées : AFF +, AFF -, + 20 V, 2 V.

Les touches 20 V et 2 V indiquent le calibre du voltmètre pour les p d'aiguille indiquées + et - 1. Ce sont les seuls repères qui rappellent à l'op que la grandeur + 1 correspond à 20 V.

SCHEMA DE CONNEXION DU VOLTMETRE A L'AIDE DES TOUCHES 20 V et 2 V



La ligne mesure est reliée à tous les boutons de mesure du panneau de câblage (face avant des modules).

Comme l'indique le schéma, le bouton et la douille de mesure situés sous le panneau de commande sont prioritaires.

Les touches "AFF +, AFF -" permettent l'affichage des paramètres du calcul et la mesure des résultats, par une méthode d'opposition.

L'enclenchement d'une touche AFF + (AFF -) connecte le + 20 V (- 20 V) au sommet d'un potentiomètre de comparaison et fait fonctionner le voltmètre en galvanomètre.

SCHEMA DES CONNEXIONS DU VOLTMETRE A L'AIDE DES TOUCHES AFF + et AFF - (v. p. II.8)

Grâce à un amplificateur semi-logarithmique, la sensibilité du voltmètre utilisé en galvanomètre est maximum au centre du cadran, elle est nulle aux extrémités.

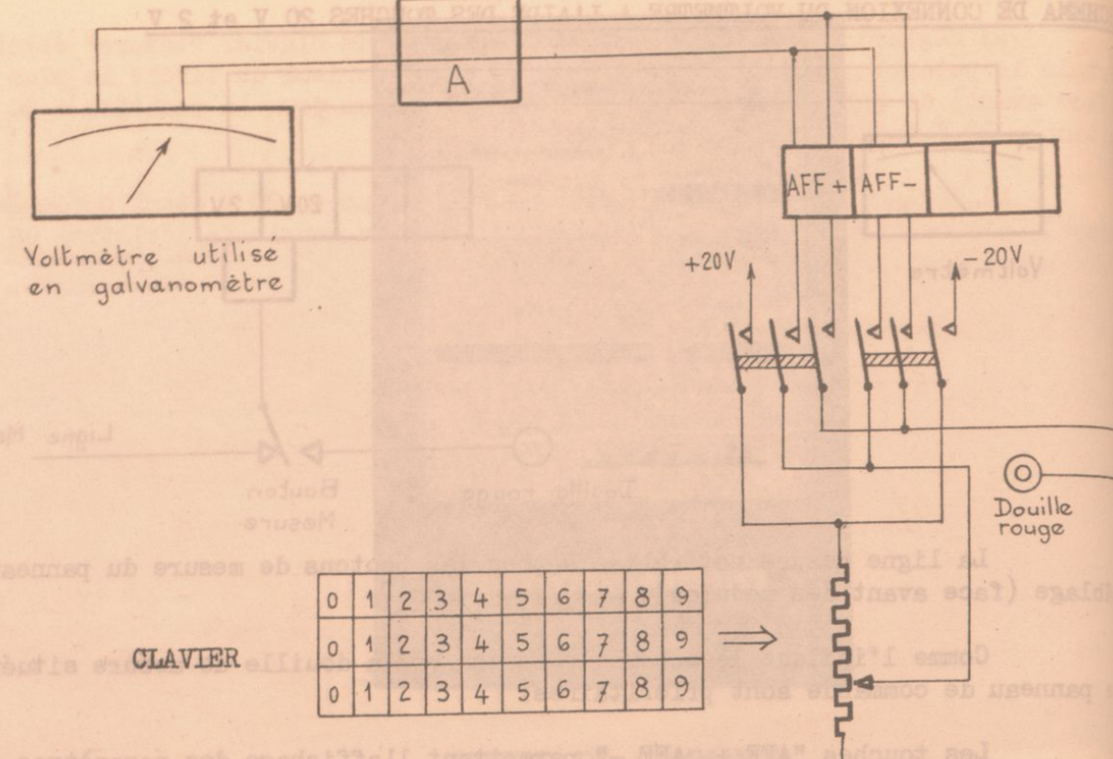
Le positionnement du curseur du potentiomètre de comparaison est réalisé par l'enfoncement de trois touches du clavier d'affichage y compris les touches zéro si besoin est (la valeur 1 est obtenue lorsqu'aucune touche n'est enfoncée).

2.2,4. CLAVIER FONCTIONNEL

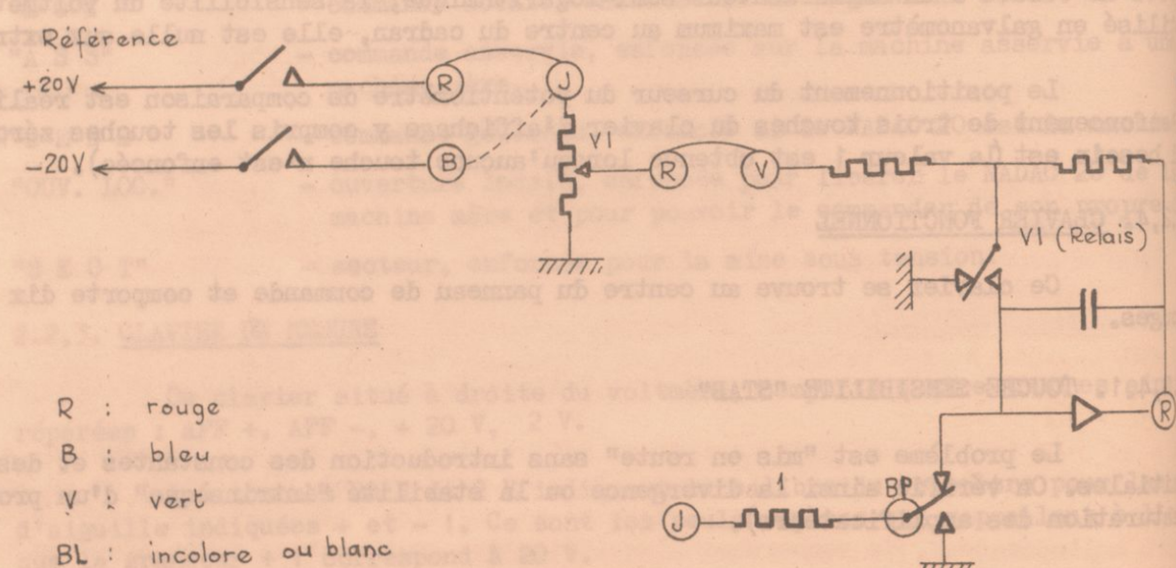
Ce clavier se trouve au centre du panneau de commande et comporte dix touches rouges.

2.2,4.1. TOUCHE SENSIBILITE "STAB"

Le problème est "mis en route" sans introduction des constantes et des valeurs initiales. On vérifie ainsi la divergence ou la stabilité "intrinsèque" d'un problème (saturation des amplificateurs).



A : Amplificateur à loi semi-logarithmique



R : rouge

B : bleu

V : vert

BL : incolore ou blanc

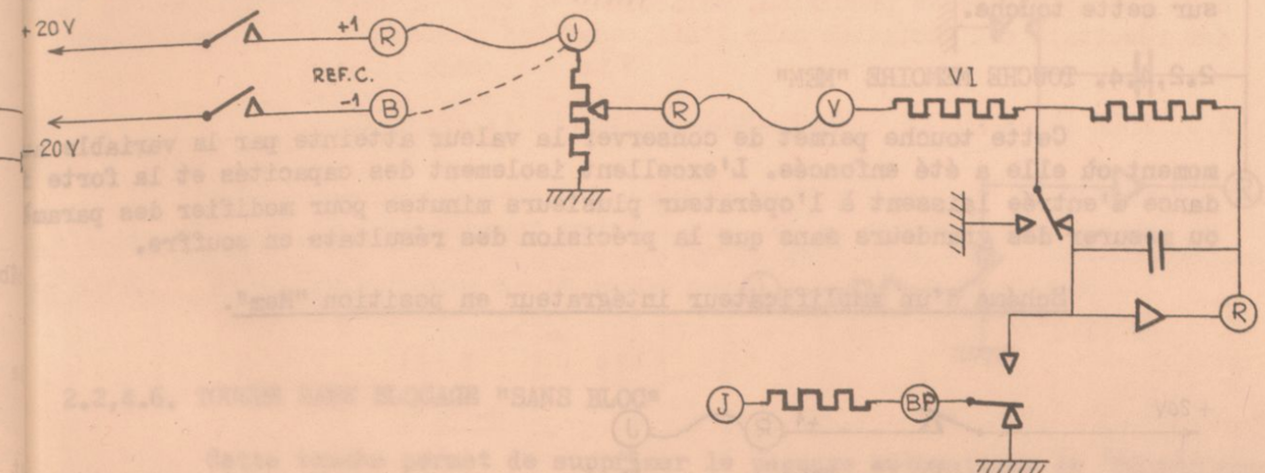
J : jaune

SCHEMA D'UN AMPLIFICATEUR EN POSITION STABILISEE

2.2,4.2. TOUCHE ATTENTE "ATT"

C'est la position de repos quand la machine reste sous tension. Pour éviter tout accident électrique (rencontre fil-panneau métallique) ne câbler, ne modifier un câblage qu'en position attente

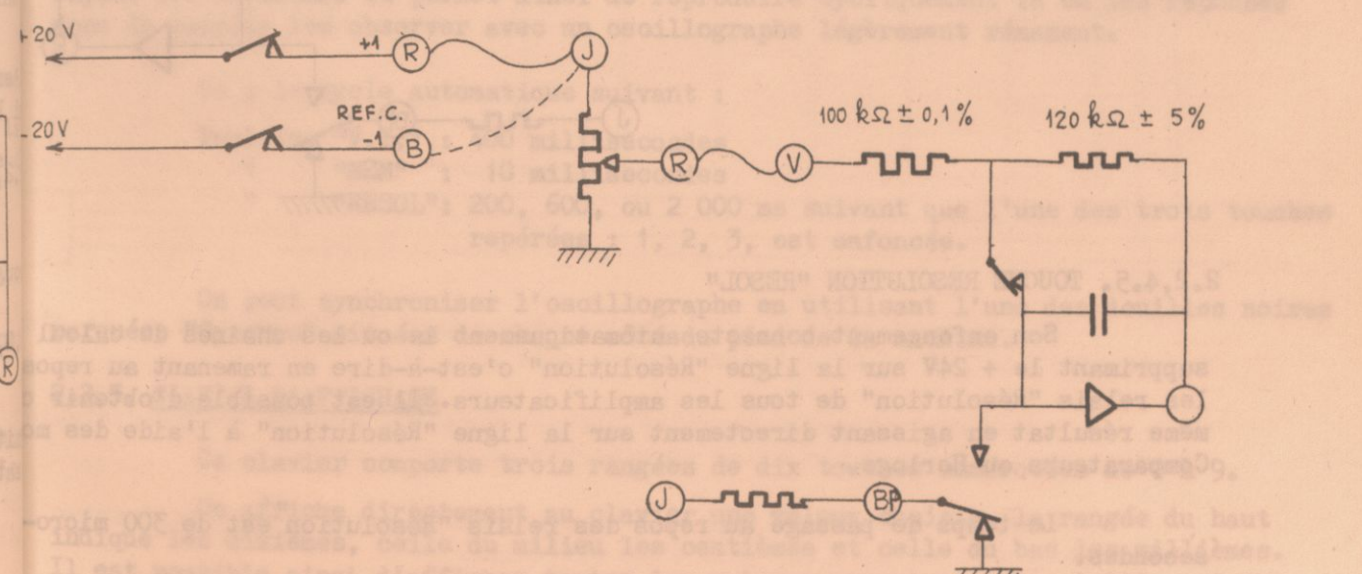
Schéma d'un amplificateur intégrateur en position attente



2.2,4.3. TOUCHE VALEUR INITIALE "V.I."

Cette touche permet l'affichage des conditions initiales à la sortie des amplificateurs intégrateurs.

Schéma d'un amplificateur intégrateur en position "V.I."



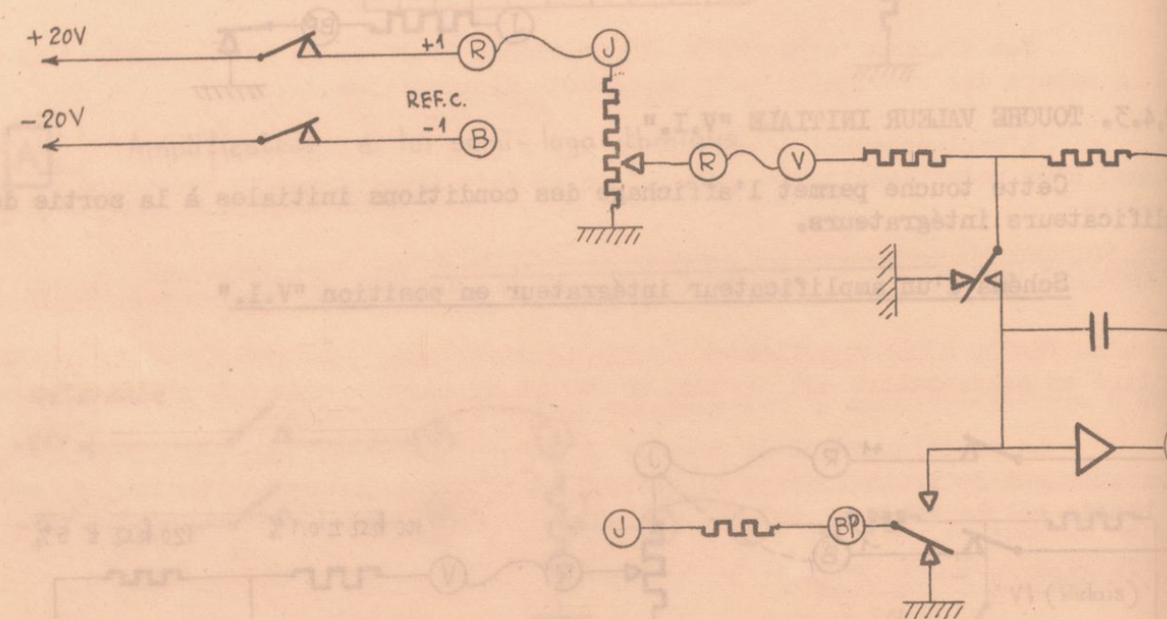
Les valeurs différentes des résistances d'attaque (100 kΩ, précis
qu'utile pour multiplier la grandeur d'entrée par 10, lorsque l'amplificateur
utilisé en sommateur) et de bouclage (120 KΩ peu précise) se justifient à c
la chute interne du potentiomètre ; 100 KΩ est une résistance faible qui se tr
ainsi plus ou moins compensée. Cela impose en tout cas la méthode normale de
d'une valeur initiale, c'est-à-dire à la sortie de l'amplificateur et non au
du potentiomètre.

Le temps de charge de la capacité impose un arrêt minimum de une se
sur cette touche.

2.2,4.4. TOUCHE MEMOIRE "MEM"

Cette touche permet de conserver la valeur atteinte par la variable
moment où elle a été enfoncée. L'excellent isolement des capacités et la fort
dance d'entrée laissent à l'opérateur plusieurs minutes pour modifier des par
ou mesurer des grandeurs sans que la précision des résultats en souffre.

<u>Schéma d'un amplificateur intégrateur en position "Mem".</u>

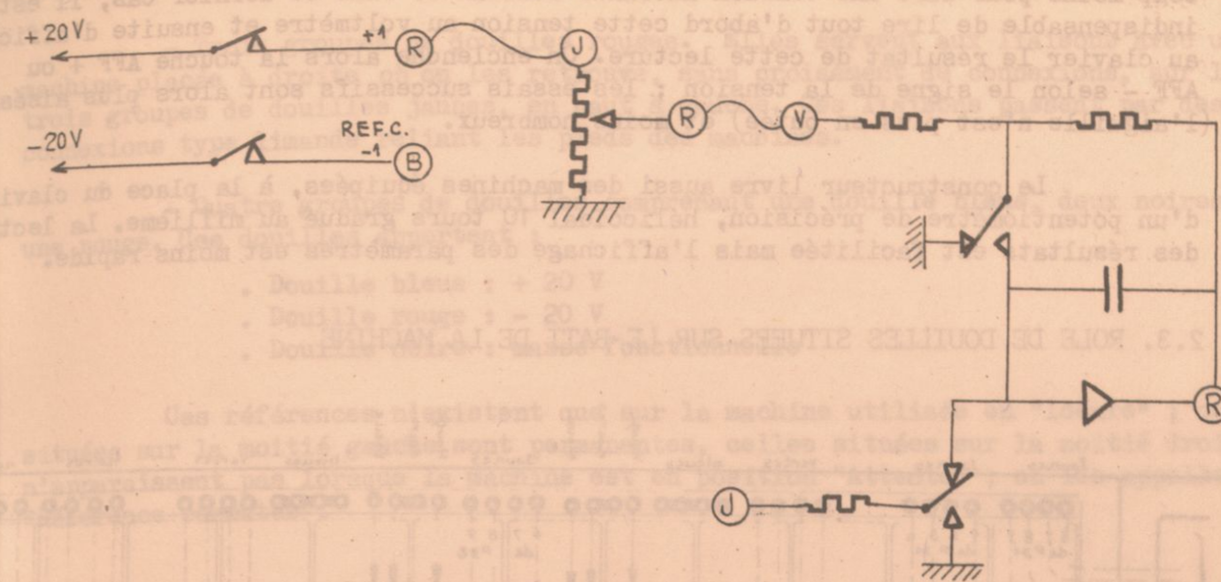


2.2,4.5. TOUCHE RESOLUTION "RESOL"

Son enfoncement connecte automatiquement la ou les chaines de calc
supprimant le + 24V sur la ligne "Résolution" c'est-à-dire en ramenant au re
les relais "Résolution" de tous les amplificateurs. Il est possible d'obtenir
même résultat en agissant directement sur la ligne "Résolution" à l'aide des
Comparateurs ou Horloge.

Le temps de passage au repos des relais "Résolution" est de 300 mic
secondes.

<u>Schéma d'un amplificateur intégrateur en position "Résolution"</u>



2.2,4.6. TOUCHE SANS BLOCAGE "SANS BLOC"

Cette touche permet de supprimer le passage automatique de "Résolution" en "Mémoire" lorsqu'un amplificateur sature ; seule, la lampe rouge s'allume. On utilise cette propriété quand la grandeur de sortie d'un amplificateur dépasse de peu 20 V et pour peu de temps, ou lorsqu'au moment de la saturation, elle n'a plus d'intérêt pour la suite du problème.

2.2,4.7. TOUCHE REPETITIF "REP"

L'enfoncement de cette touche change toutes les capacités des intégrateurs de 1 microfarad en 0,01 microfarads. Cela se traduit par une évolution 100 fois plus rapide des problèmes et permet ainsi de reproduire cycliquement la ou les réponses donc de pouvoir les observer avec un oscillographe légèrement rémanent.

On a le cycle automatique suivant :

Position "V.I." : 400 millisecondes
" "MEM" : 10 millisecondes
" "RESOL": 200, 600, ou 2 000 ms suivant que l'une des trois touches repérées : 1, 2, 3, est enfoncée.

On peut synchroniser l'oscillographe en utilisant l'une des douilles noires marquées "Synchro" situées de chaque côté du pied de la machine.

2.2,5. <u>CLAVIER D'AFFICHAGE</u>

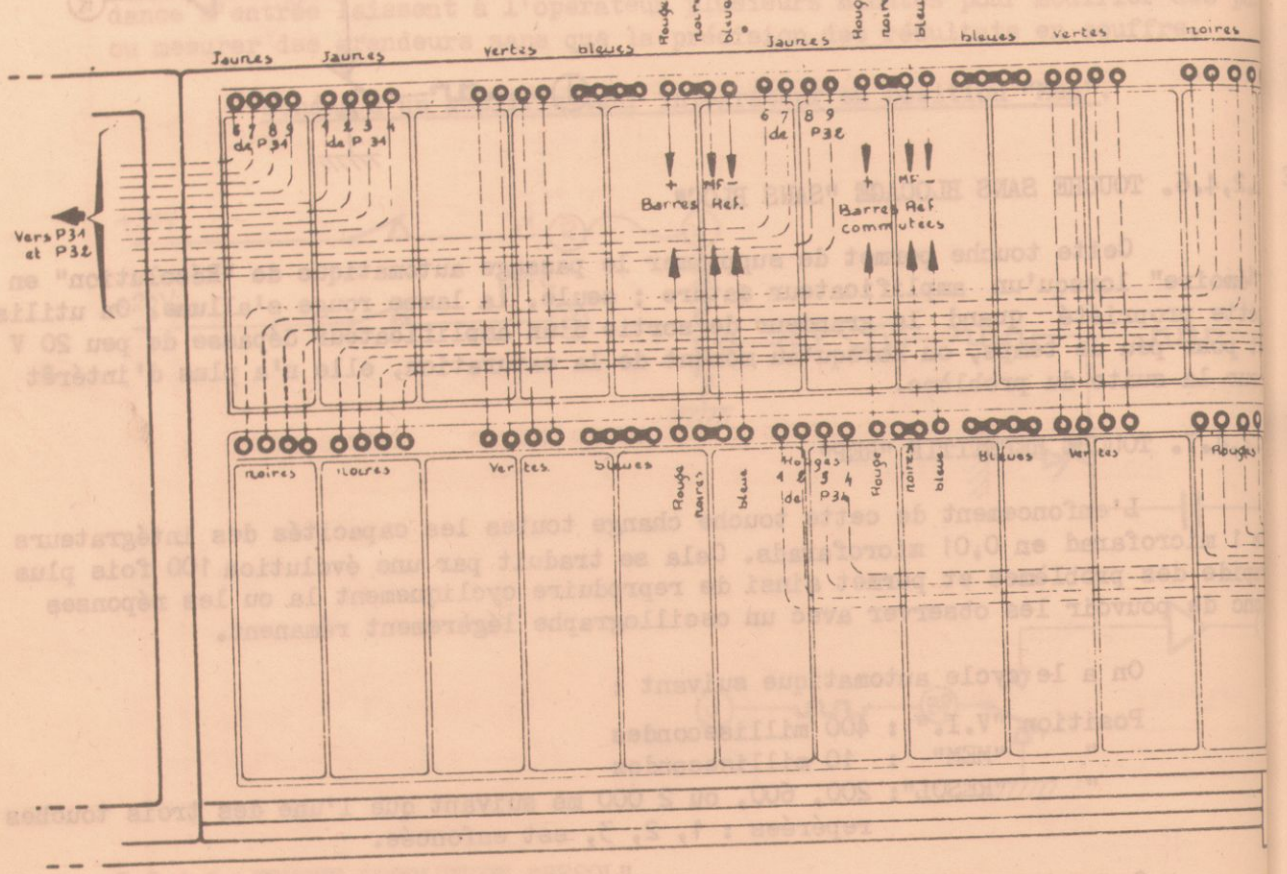
Ce clavier comporte trois rangées de dix touches numérotées de 0 à 9.

On affiche directement au clavier une valeur choisie, la rangée du haut indique les dixièmes, celle du milieu les centièmes et celle du bas les millièmes. Il est possible ainsi d'afficher toutes les valeurs comprises entre 0,000 et 0,999.

Pour afficher la grandeur 1, il suffit de n'enfoncer aucune touche. Le clavier
très commode pour régler une tension (affichage des constantes) mais il l'est
coup moins pour lire une tension inconnue constante. Dans ce dernier cas, il
indispensable de lire tout d'abord cette tension au voltmètre et ensuite d'aff
au clavier le résultat de cette lecture. On enclenche alors la touche AFF + ou
AFF - selon le signe de la tension ; les essais successifs sont alors plus ai
(l'aiguille n'est plus en butée) et moins nombreux.

Le constructeur livre aussi des machines équipées, à la place du cl
d'un potentiomètre de précision, hélicoïdal 10 tours gradué au millième. La le
des résultats est facilitée mais l'affichage des paramètres est moins rapide.

2.3. ROLE DE DOUILLES SITUEES SUR LE BATI DE LA MACHINE



On pourra consulter la figure ci-dessus pour la disposition et le
de ces douilles qui se répartissent comme suit :

- Quatre groupes de douilles bleues. Dans ces groupes indépendants
douilles sont reliées entre elles pour servir de multiplage.

- Quatre groupes de douilles vertes ; elles servent de renvoi entr
tiroirs du bas et du haut par huit fils verticaux.

- Quatre groupes de douilles noires. Les deux groupes situés en haut à droite sont reliés aux deux groupes situés en bas à gauche.

- Trois groupes de douilles rouges. Elles servent aux liaisons avec une machine placée à droite où on les retrouve, sans croisement de connexions, sur les trois groupes de douilles jaunes, en haut à gauche. Ces liaisons passent par des connexions type limande reliant les pieds des machines.

- Quatre groupes de douilles comprenant une douille bleue, deux noires et une rouge. Ces douilles apportent :

  . Douille bleue : + 20 V
  . Douille rouge : - 20 V
  . Douille noire : masse fonctionnelle

Ces références n'existent que sur la machine utilisée en "locale" ; celles situées sur la moitié gauche sont permanentes, celles situées sur la moitié droite n'apparaissent pas lorsque la machine est en position "Attente" ; on les appelle "Référence commutée".

Chapitre III

LES MODULES

3.1. GENERALITES

Sur tous les modules, les douilles jaunes sont les entrées, les douilles rouges sont les sorties, les douilles blanches sont des points "chauds", entrées directes d'un amplificateur ou destinées à être reliées à une entrée directe d'amplificateur. (Leurs connexions doivent être très courtes).

Les douilles rouges, bleues ou noires en bas des tiroirs sont le + 20 V, le - 20 V et la masse fonctionnelle. Ces tensions apparaissent en permanence ou sont supprimées lorsque la machine est en position "attente". Dans ce dernier cas, elles sont alors repérées "Référence Commutée".

Les boutons "Test" ont des rôles divers, mais tous les boutons "Mesure" connectent la sortie qu'ils déservent sur l'unique ligne mesure de la NADAC. La seule fausse manoeuvre possible consiste à appuyer sur deux boutons "Mesure" simultanément. On relierait alors deux générateurs de f.e.m. différentes et de faible résistance interne ; on détruirait alors les potentiomètres ou les transistors de sortie des amplificateurs.

3.2. MODULES POTENTIOMETRES

Un potentiomètre s'utilise pour afficher un paramètre ou pour multiplier une variable par une constante.

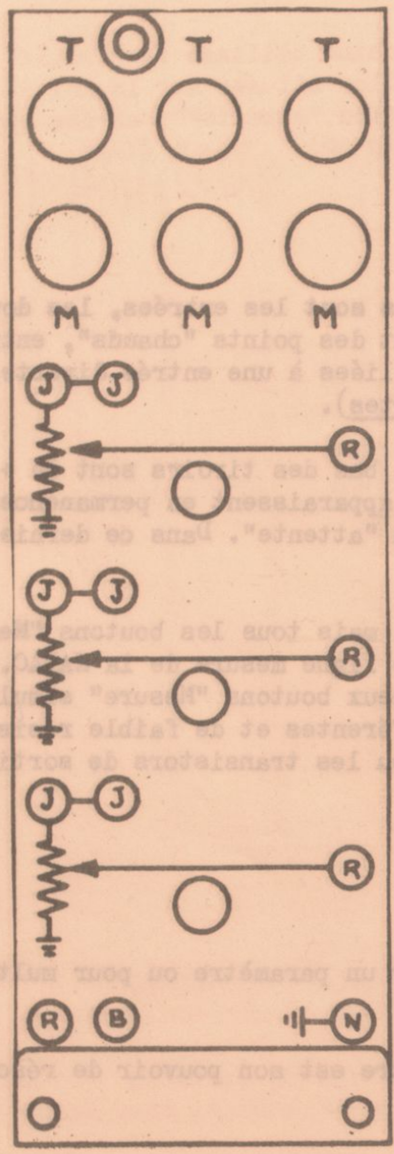
La qualité essentielle d'une potentiomètre est son pouvoir de résolution (ou son nombre de spires par degré).

La précision de sa résistance totale R est sans intérêt. Comme la ré
tance vue du curseur varie entre 0 et R/4, la chute de tension du générateur c
tituée par le curseur variable, rend la linéarité du potentiomètre sans intér

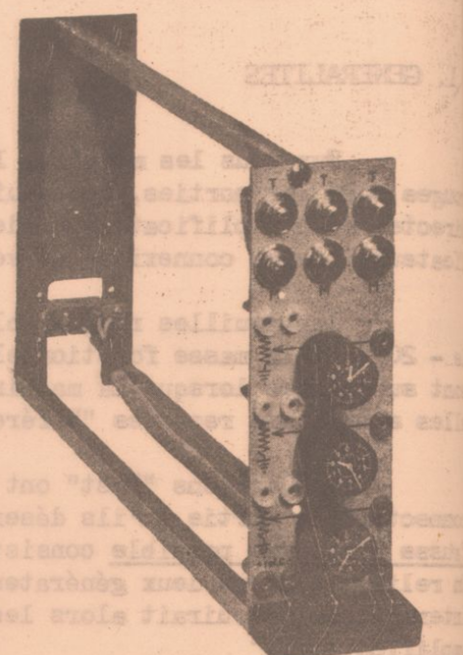
Il résulte de cela que <u>l'on ne devra jamais</u> :

- Utiliser les graduations du cadran sauf comme une indication très approximat
- Mesurer une grandeur sur un curseur sans qu'il débite normalement c'est-à-di que tout le problème soit câblé.

3.2,1. <u>MODULE POTENTIOMETRE M 20 P 30</u>



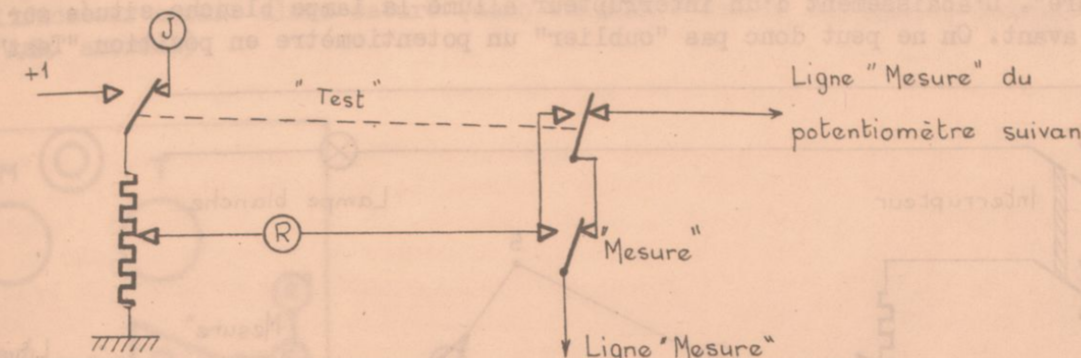
Face avant module M 20 P 30



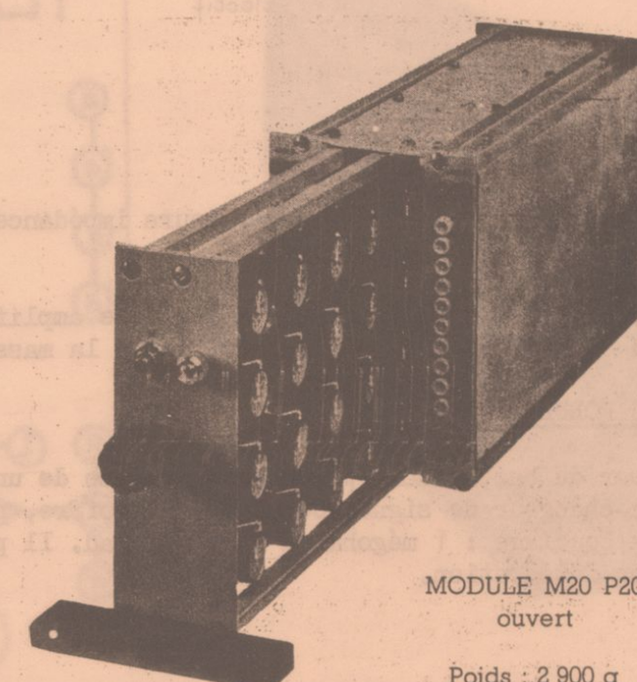
Ce module comprend trois potentiomètres hélicoïdaux "Dixtours" type "BOURNS", Résistance 50 KΩ $\pm$ 5 %, linéarité 0,1 % 10 000 spires pour dix tours.

Les boutons Test et Mesure de gauche à droite, intéressent les potentiomètres de haut en bas. Le bouton Test s'utilise particulièrement pour afficher la valeur de multiplication d'un potentiomètre alimenté par une sortie d'amplicateur. Son action transfère le haut du potentiomètre de la douille jaune à la référence + 1. On utilise donc un "AFF +" pour positionner le potentiomètre.

<u>Schéma de l'action des boutons "Test" et "Mesure"</u>



3.2,2. <u>MODULE M 20 P 200</u>



MODULE M20 P200
ouvert

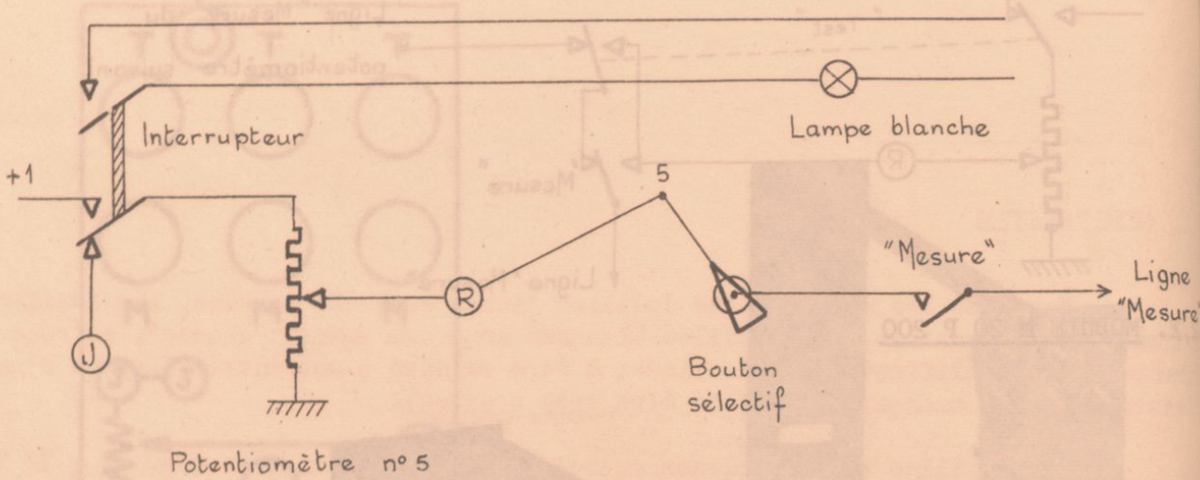
Poids : 2 900 g

Ce module contient vingt potentiomètres de même type que ceux du tir
M20 P30. Son emploi, qui conduit à concentrer les affichages en un endroit du
va en sens contraire de la technique modulaire. Large de deux unités, sa mise
place nécessite l'enlèvement d'un rail de guidage.

On doit ouvrir ce tiroir pour régler les potentiomètres.

<u>Opération Mesure</u> : sélectionner le potentiomètre par le bouton face avant, app
sur "Mesure".

<u>Opération Test</u> : Baisser l'interrupteur associé au potentiomètre, faire l'opéra
"Mesure". L'abaissement d'un interrupteur allume la lampe blanche située sur la
face avant. On ne peut donc pas "oublier" un potentiomètre en position "Test".



3.3. MODULE M 20 A2

Ce module contient deux amplificateurs, leurs impédances de bouclage e
résistances d'entrée.

Ces résistances d'entrée sont reliées à l'entrée amplificateurs en pos
"Résolution" du panneau de commande, elles sont reliées à la masse en "attente"

3.3.1. <u>PANNEAU AVANT DU MODULE</u>

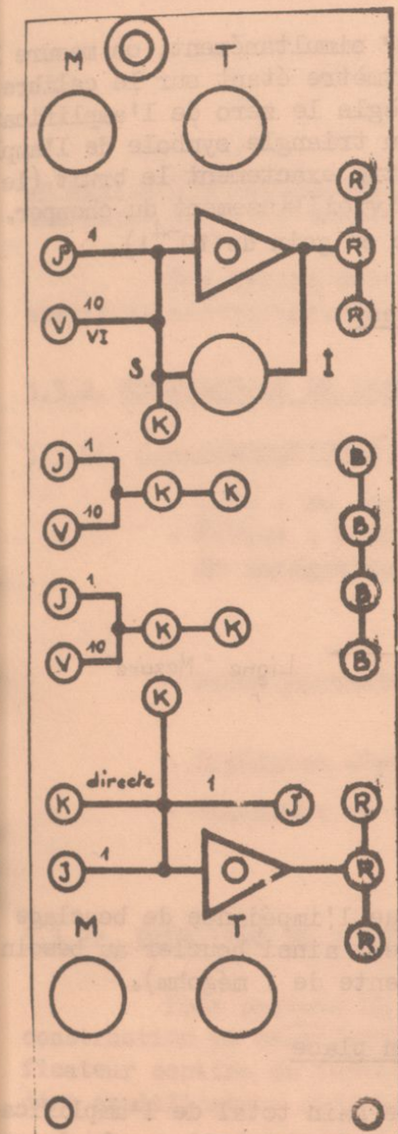
L'amplificateur du bas, bouclé par une résistance de un mégohm, ne pe
utilisé qu'en sommateur-changeur de signe. Celui du haut offre, par le jeu de l'
seur S ou I, le choix du bouclage : 1 mégohm ou 1 microfarad. Il permet donc la f
sommation et la fonction intégration.

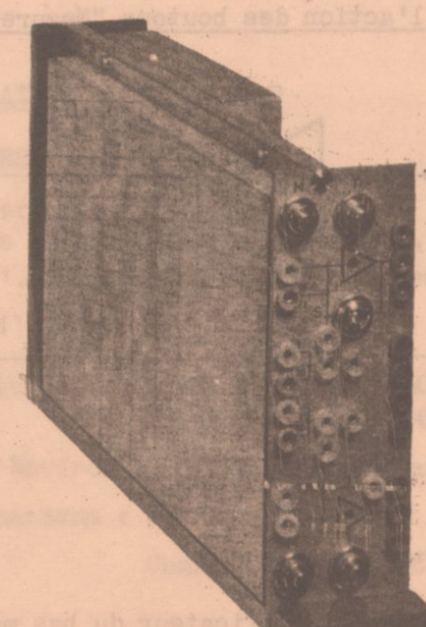
<u>Résistances d'entrée</u> :

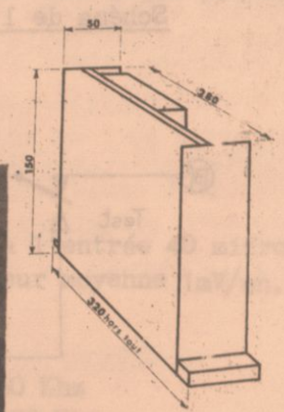
<u>En haut</u> :

Les résistances de 1 mégohm, douille jaune (gain 1) et de 0,1 mégohm, douille verte (gain 10) sont impérativement liées à l'amplificateur du haut. En position I (intégrateur) la résistance indiquée 10 et V.I., de 0,1 mégohm, est utilisée pour l'introduction de la valeur initiale de la sortie de l'amplificateur <u>selon un schéma particulier</u> : elle est toujours reliée à l'entrée directe de l'amplificateur.

En conséquence, il <u>est impossible sur 10 V.I.</u> lorsque l'amplificateur est bouclé par 1 M Ω (clef sur S) <u>de mesurer</u> une grandeur d'entrée supérieure à 0,1 : l'amplificateur serait alors saturé (donc tension d'entrée non nulle, courant dans R.10.V.I. anormal)







Face avant du module

Au centre :

Les deux résistances de 1 mégohm, douille jaune, et les deux résistan
0,1 mégohm, douille verte, sont indépendantes. Elles peuvent être affectées à l'
ficateur du haut, du bas ou même à un amplificateur d'un autre tiroir (par conn
courtes, douilles blanches).

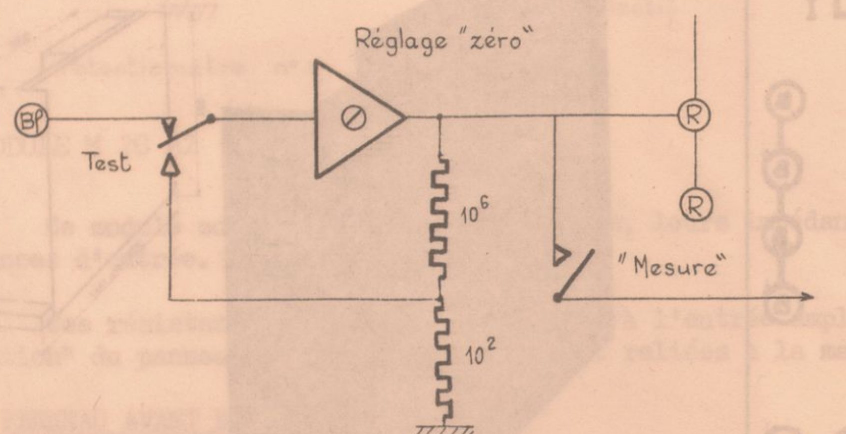
En bas :

La résistance de 1 mégohm, douille jaune est liées à l'amplificateur d

Le bouton "Mesure" connecte la sortie d'amplificateur à la ligne "Mes
Le bouton "Test" déconnecte l'entrée de l'amplificateur pour la relier à un bou
de 1 mégohm, assurant un gain de 10^{-4}. De ce fait, on ne peut pas appuyer sur "
pendant la résolution d'un problème sans fausser les résultats.

En appuyant sur les boutons "mesure" et "Test" simultanément, on mesu
tension de dérive et le bruit de l'amplificateur. Le voltmètre étant sur le calib
2 V, l'aiguille doit rester dans la plage colorée. On règle le zéro de l'amplifi
par le potentiomètre à axe fendu accessible au centre du triangle symbole de l'a
ficateur. Une mesure à l'oscillographe permet de connaître exactement le bruit (
voltmètre "intégré" le bruit) et aussi de surveiller le vieillissement du choppe
bruit doit être inférieur à 6 volts crête à crête (avec ce gain de 10^{-4} !).

Schéma de l'action des boutons "Mesure" et "Test"

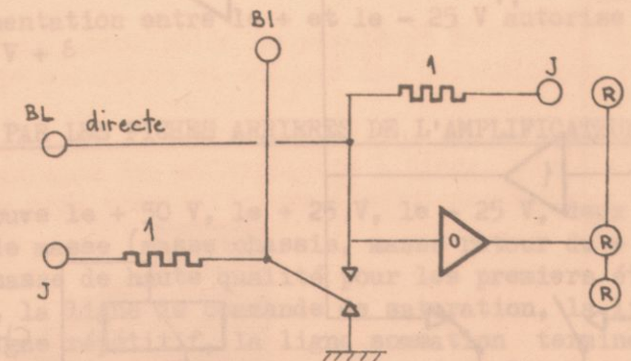


Le dessin de l'amplificateur du bas montre que l'impédance de boucla
n'existe qu'à l'aide d'une connexion extérieure ; on peut ainsi boucler au bes
l'amplificateur sur une impédance particulière (différente de 1 mégohm).

On doit normalement toujours laisser cette connexion en place

En effet, sa suppression ferait apparaître le gain total de l'amplif
teur (10^7) c'est-à-dire une tension de sortie atteignant la saturation - la la
rouge s'allume.

Schéma particulier de l'amplificateur bas



On doit éviter de laisser longtemps un amplificateur en saturation : Le déséquilibre thermique de ses éléments constitutifs apporterait une dérive longue à disparaître.

Les quatre douilles bleues situées à droite sur la face avant peuvent servir de multiplage.

3.3,2. TECHNOLOGIE DE L'AMPLIFICATEUR

3.3,21. CARACTERISTIQUES GENERALES

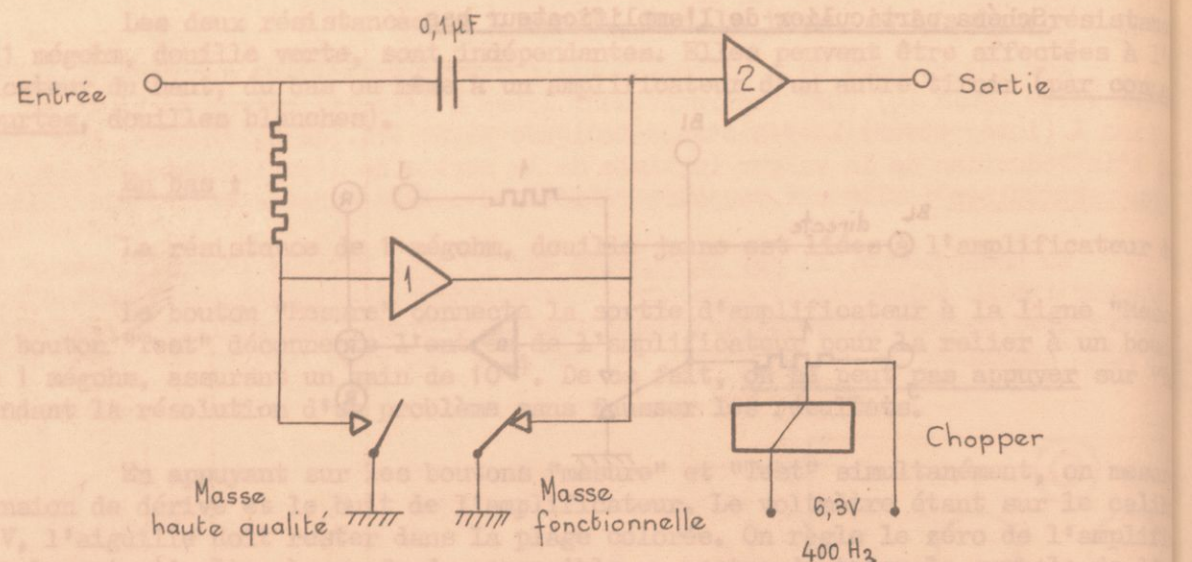
- Gain : en chaine ouverte supérieur à 10^7
- Dérive : à long terme (2 à 3 semaines) ramenée à l'entrée 40 microvolts. En intégrateur RC = 1, inférieure à 2mV/mn, valeur moyenne 1mV/mn.

- Bande passante en gain de I : Minimum - 3 dB à 50 Khz
  Moyenne - 3 dB à 100 Khz
- Impédance d'entrée : Environ 1 mégohm, diminue quand la fréquence augmente.
- Grandeurs de sortie maximum : Tension $\pm$ 20 V + ε
  Courant $\pm$ 10 mA + ε

3.3,22. PRINCIPES DE CONSTRUCTION

Pour obtenir un grand gain avec une faible dérive, la S.E.A. a adopté une construction en deux parties : étages finaux à liaisons directes c'est-à-dire amplificateur continu de faible gain, de faible dérive, Etage d'entrée à chopper c'est-à-dire amplificateur alternatif de grand gain, par principe sans dérive.

Schéma particulier de l'amplificateur

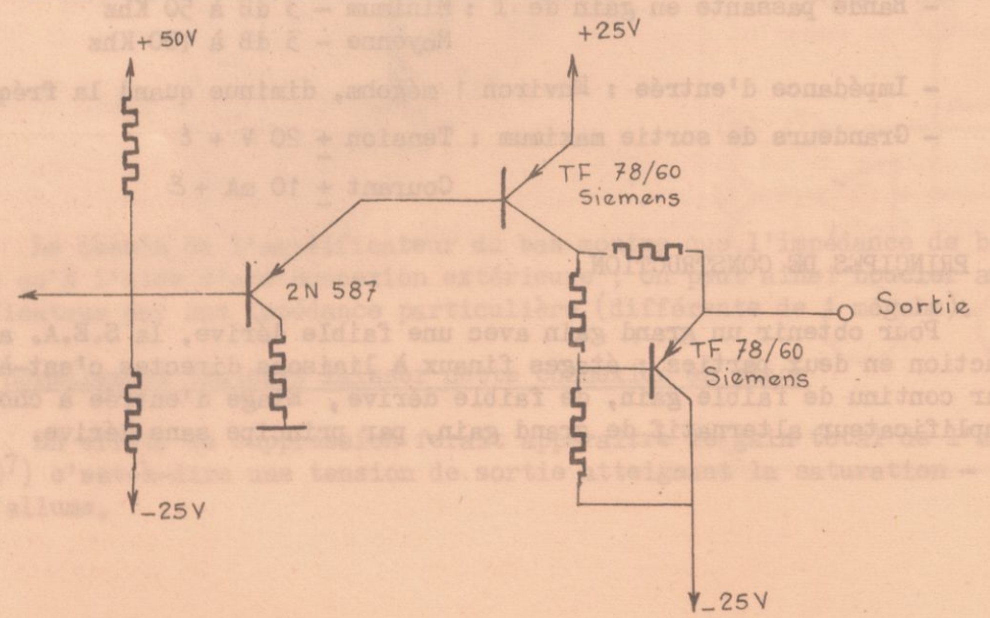


L'amplificateur I reçoit la tension d'entrée découpée ; le deuxième co tact du chopper supprime, par son action synchrone, les alternances d'une polar opposée à celles de l'entrée de I et présente ainsi en entrée de II une tension même forme que celle de l'entrée de I mais amplifiée.

Un filtrage, non représenté, transforme les impulsions dent de peigne la sortie de I en une tension peu ondulée ; le condensateur de 0,1 microfarad r l'entrée à l'amplificateur II améliore la bande passante de l'amplificateur tot

Le chopper reste le point de faible fiabilité de ce montage. L'amplifi teur II possède le potentiomètre de réglage de symétrie.

Schéma de principe de l'étage de sortie



L'alimentation entre le + et le - 25 V autorise un potentiel de sortie entre + et - 20 V + ε

3.3.3. LIAISONS PAR LES FICHES ARRIERES DE L'AMPLIFICATEUR A LA MACHINE

On trouve le + 50 V, le + 25 V, le - 25 V, deux broches pour le 400 Hz, quatre broches de masse (masse chassis, masse retour du + 24 V des relais, masse fonctionnelle, masse de haute qualité pour les premiers étages des amplificateurs. La ligne mesure, la ligne de commande de saturation, la ligne résolution, la ligne du + 24 V, la ligne répétitif, la ligne sommation terminent ce brochage. Ces quatre dernières lignes assurent par l'intermédiaire de 5 relais + diodes, les connexions ordonnées par le clavier fonctionnel.

3.3.4. IMPEDANCES DE CALCUL

Les résistances sont des résistances pelliculaires, type haute qualité professionnelle, vieillies, triées dans des lots à 0,1 % type R.C.M.C.K. 4 Sfernice ou 7020 R.L.C. (U.S.A.).

Le condensateur de 1 microfarad est une batterie de trois capacités 0,33 microfarads ± 1 %, 60 V + 1 appoint, au Styroflex, type P.N.6 Stéafix.

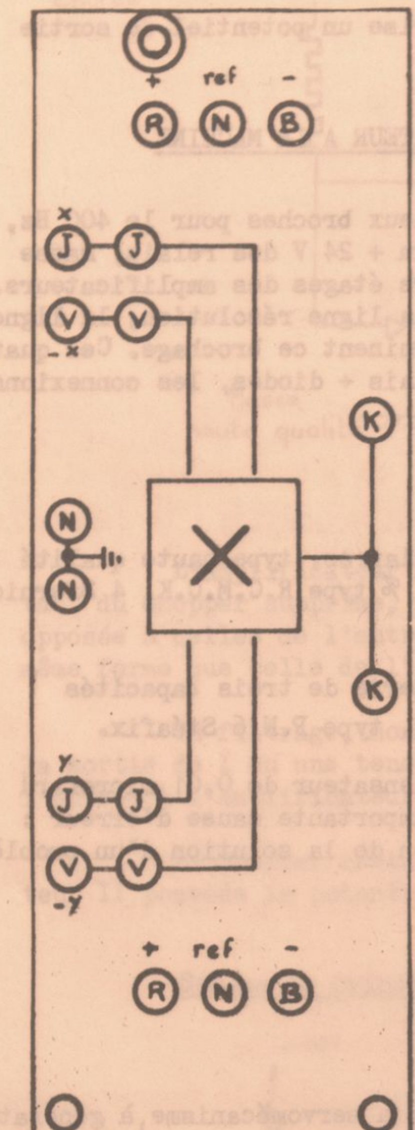
Pour le fonctionnement en répétitif, le condensateur de 0,01 microfard est à ± 2 %, en Styroflex. Cette imprécision est une importante cause d'erreur : le fonctionnement répétitif ne permet que l'observation de la solution d'un problème.

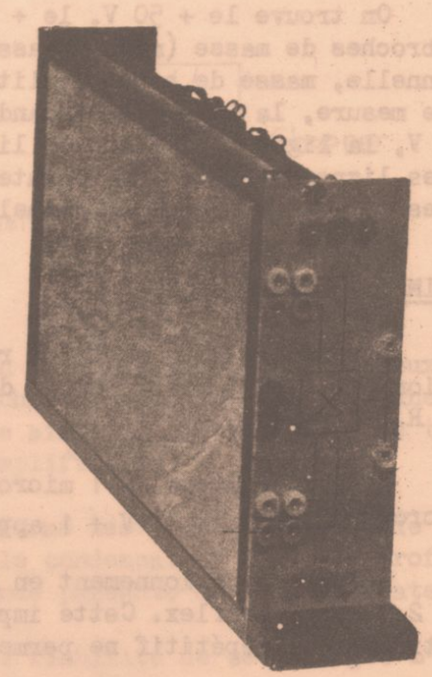
3.4. MULTIPLIEURS

Parmi les trois types de multiplieurs connus : à servomécanisme, à générateur parabolique à diodes, à découpage temporel ; la S.E.A. présente pour le NADAC 20 des tiroirs des deux premiers types.

L'utilisation de tout module "non linéaire", donc de multiplieur, abaisse notablement la précision des calculs. Si l'on peut effectuer une addition, une intégration de durée de l'ordre de la minute, avec une précision de l'ordre de ± 0,1 %, la multiplication s'effectue avec la précision d'environ ± 0,5 % ; cela avec l'un ou l'autre des modèles de multiplieurs de la Nadac 20.

3.4,1. MULTIPLIEUR A DIODES M 20 Md





Face avant du module M 20 Md

Deux générateurs de parabole, 10 diodes, c'est-à-dire à 11 segments pa
branche, permettent de réaliser :

$$\left(\frac{X+Y}{2}\right)^2 - \left(\frac{X-Y}{2}\right)^2 = XY$$

Il est nécessaire de fournir à ce multiplieur les valeurs : X, - X, +
- Y. La sortie de ce tiroir (une douille incolore à haut isolement) doit oblig
rement être renvoyée à l'entrée directe (douille incolore) d'un amplificateur b
par une résistance de 1 mégohm. Il apparait à la sortie de cet amplificateur -

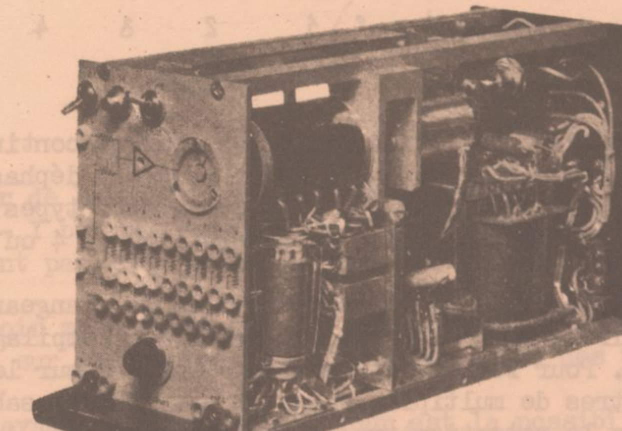
Le principe du fonctionnement des générateurs de paraboles à diodes rend ce multiplieur très sensible à la valeur absolue de la tension de référence. Le réglage des réseaux de diodes du multiplieur s'exécute selon une règle décrite dans la notice.

Lors d'une association de plusieurs machines, il faut veiller à l'égalité de leur tension de référence et engager des multiplieurs étalonnés pour cette valeur de tension.

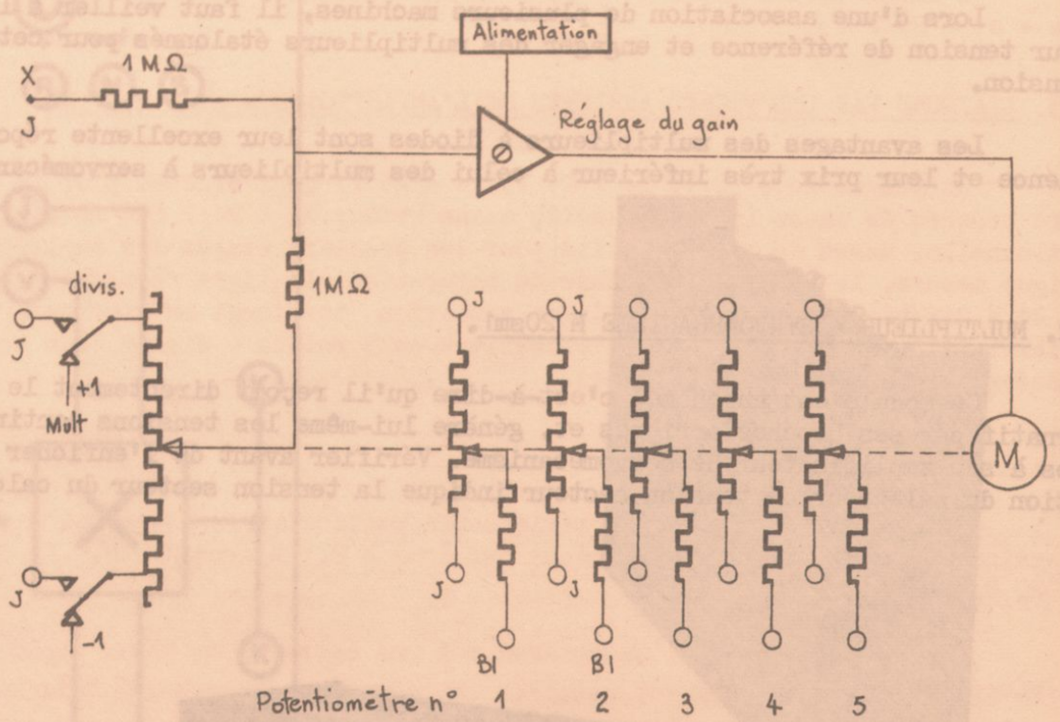
Les avantages des multiplieurs à diodes sont leur excellente réponse en fréquence et leur prix très inférieur à celui des multiplieurs à servomécanisme.

3.4,2. MULTIPLIEUR A SERVOMECANISME M 20sm1.

Ce module est autonome, c'est-à-dire qu'il reçoit directement le secteur alternatif par ses broches arrières et, génère lui-même les tensions continues nécessaires à son amplificateur de servomécanisme. Vérifier avant de l'enficher que la position du sélecteur de tension secteur indique la tension secteur du calculateur.



3.4.2.1. PRINCIPE DE FONCTIONNEMENT



Ce module contient des alimentations à courant continu, une alimentation à 400 Hz, un amplificateur et un moteur d'asservissement déphasé couplé par réducta 1/3 à un potentiomètre multiple de 5 (ou de 10, les deux types existent) fois 20 k un utilisé pour le fonctionnement, donc la possibilité de 4 ou 9 produits).

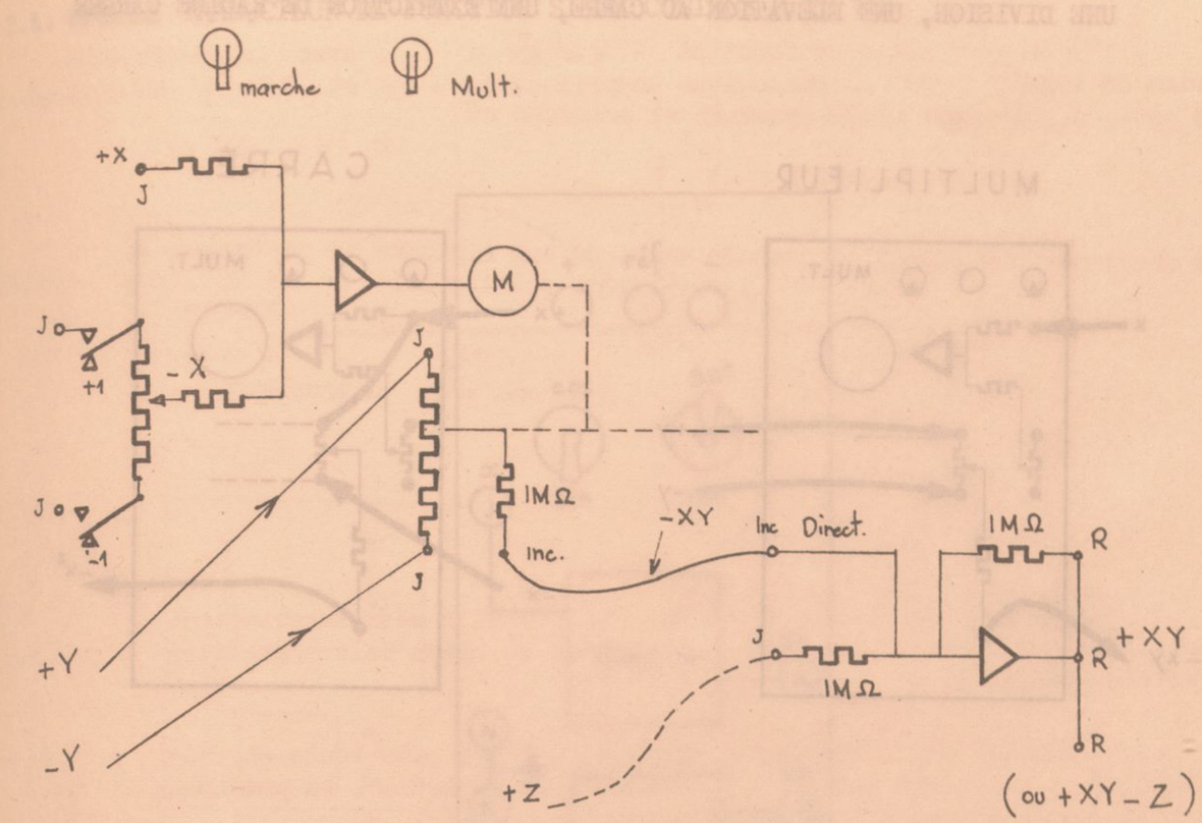
Cet asservissement de position recopie en la changeant de signe la grande d'entrée X. C'est-à-dire que la ligne des curseurs de l'empilage de 5 ou 10 potenti mètres représente - X. Pour retrouver électriquement - X sur le curseur de l'un que conque des potentiomètres de multiplication, il est indispensable qu'il présente la même chute interne que le potentiomètre capteur de position, donc que l'on trouve sur son curseur la même résistance de calcul : 1 M Ω uniquement.

Cette raison, et la sécurité d'emploi (1 curseur à la masse accidentellem détruit 1 potentiomètre) ont amené le constructeur à inclure la résistance calcul d 1 M Ω à l'intérieur du module.

3.4.2.2. UTILISATION

En alimentant par exemple en + Y en haut, - Y en bas un potentiomètre asservi en - X, il apparaîtra en sortie de ce potentiomètre le produit - XY à trave 1 MΩ et l'on ne pourra utiliser ce produit qu'en reliant cette sortie douille incol à un entre directe d'ampli-changeur de signe qui fournira en sortie + XY.

Exemple



Nota : La faible valeur de la résistance du potentiomètre multiplieurs impose la génèse de Y et - Y par des amplificateurs (si Y est un paramètre ne pas le créer directement par un potentiomètre : son curseur débiterait trop).

le bouton mesure du tiroir multiplieur permet, par l'intermédiaire du sélecteur, la mesure de - X sur 0, et sur les autres positions, la mesure des produits ou carrés.

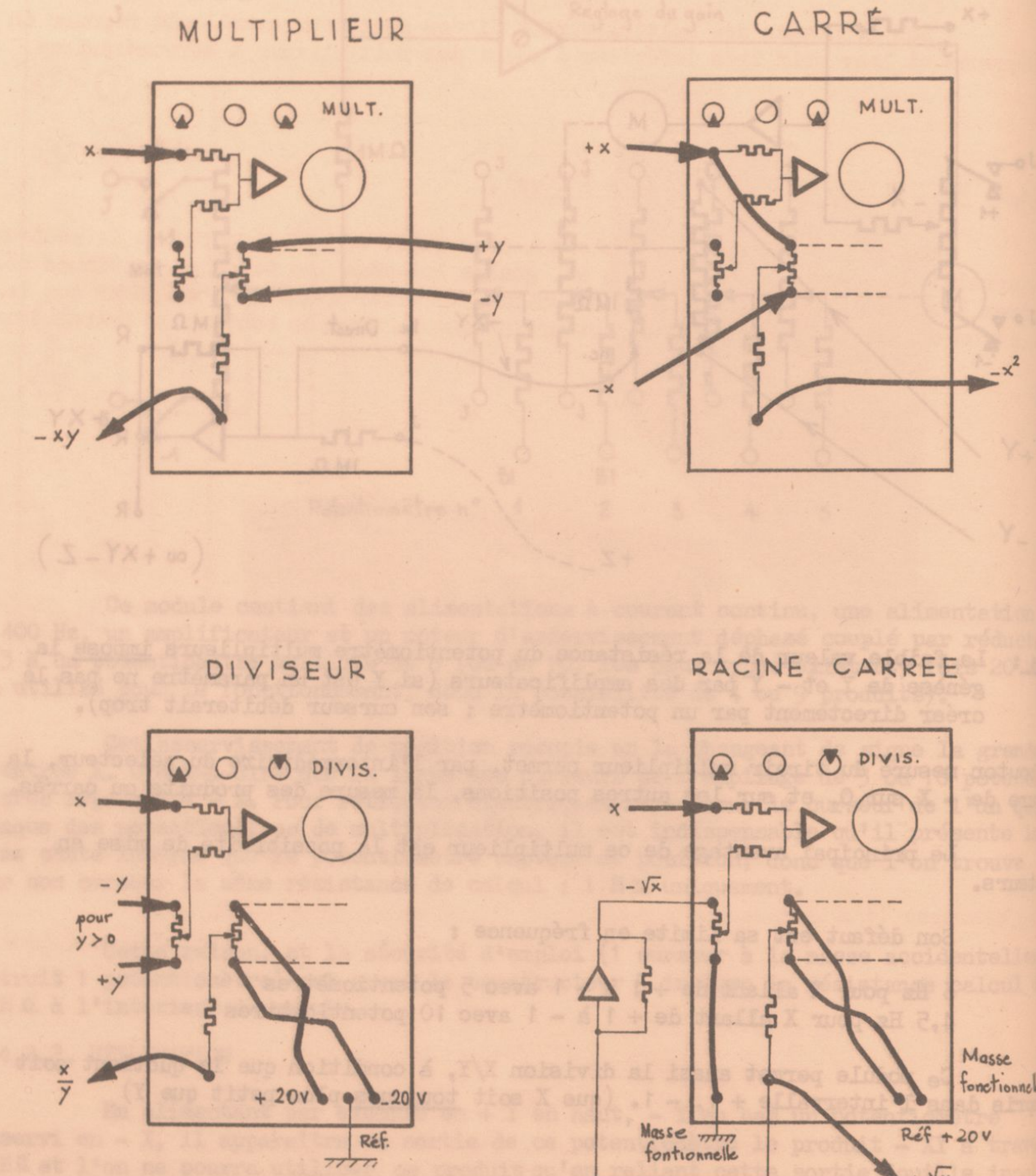
Le principal avantage de ce multiplieur est la possibilité de mise en facteurs.

Son défaut est sa limite en fréquence :

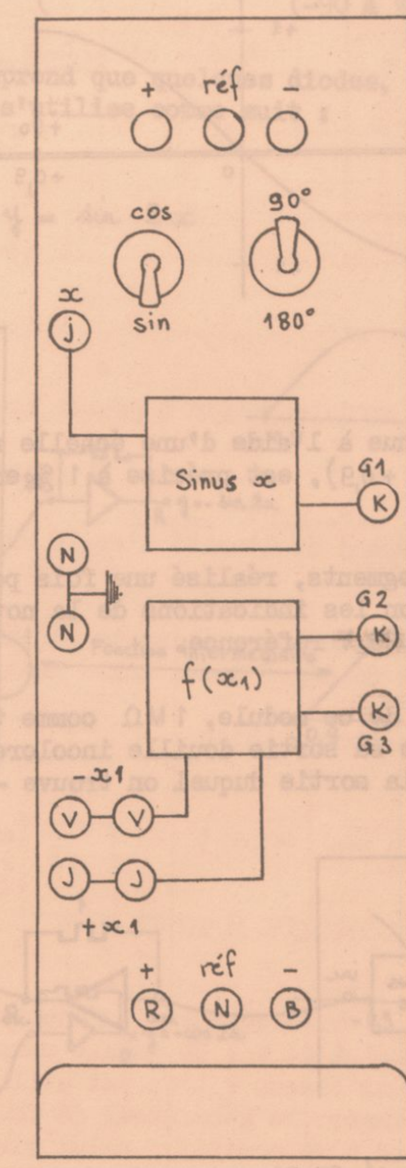
6 Hz pour X allant de + 1 à - 1 avec 5 potentiomètres
4,5 Hz pour X allant de + 1 à - 1 avec 10 potentiomètres

Ce module permet aussi la division X/Y, à condition que le quotient soit compris dans l'intervalle + 1, - 1. (que X soit toujours plus petit que Y)

SCHEMAS DES DIFFERENTS BRANCHEMENTS A EFFECTUER POUR UNE MULTIPLICATION UNE DIVISION, UNE ELEVATION AU CARRE, UNE EXTRACTION DE RACINE CARREE



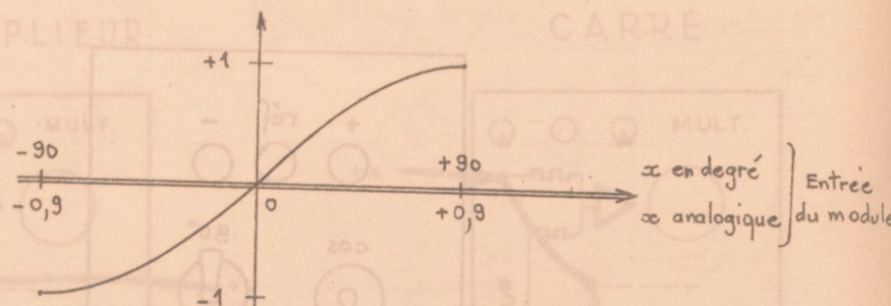
3.5. MODULE TRADUCTEUR DE FONCTION SINUS COSINUS M 20 TS



- FIGURE 24 - Platine avant Module M 20 TSC

3.5.1. CE MODULE COMPREND DEUX PARTIES

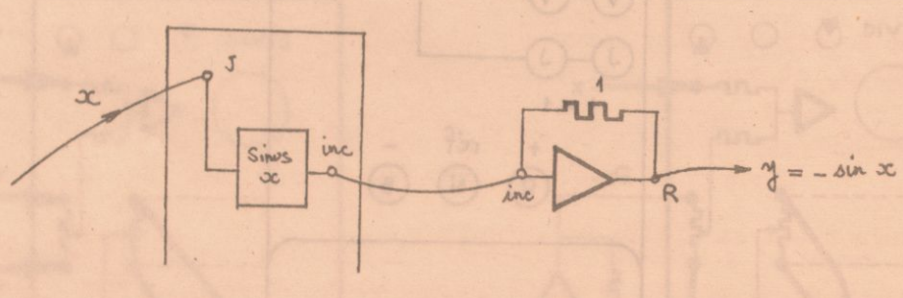
1°) Un générateur de fonction $y = \sin x$, avec x exprimé en centaine de degrés, soit x analogique compris entre $+0,9$ et $-0,9$, y correspondan[t] $+1$ et -1 (1 volt = 5° pour x)



Cette relation obtenue à l'aide d'une échelle résistances-diodes de 24 diodes (23 segments de — à $+0,9$), est précise à 1 ‰ en valeur statique, précision tombant à 4 ‰ à 10 Hz.

Le réglage des 23 segments, réalisé une fois pour toute en usine, peut se vérifier, ou se réajuster selon les indications de la notice. La précision est sensible à la valeur absolue des 20 V référence.

L'impédance interne de ce module, 1 MΩ comme tous les générateurs à diodes Nadac 20, impose la liaison de sa sortie douille incolore à une entrée directe d'amplicateur-douille incolore –, à la sortie duquel on trouve – sin x , selon le schéma :



exemple $x =$ + 0,638 soit 63,8°, $y = -0,8975$

2°) Un traducteur $f(x)$ permet avec 1/2 module $y = \sin x$ d'obtenir

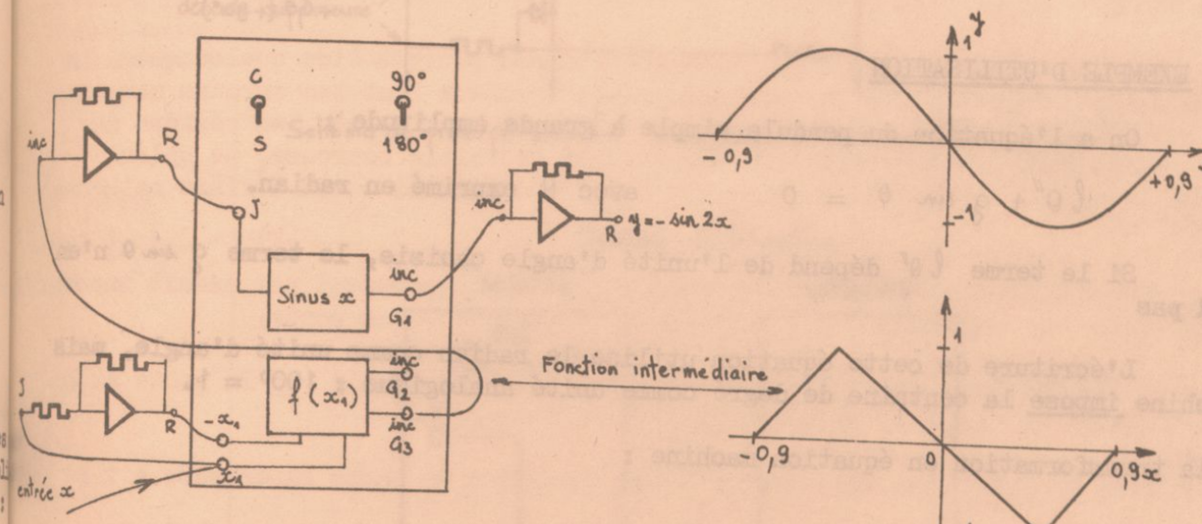
$y = -\sin 2x$ { x en centaine de degrés, compris entre $-0,9$ et $+0,$[9] (–180 et 180 °

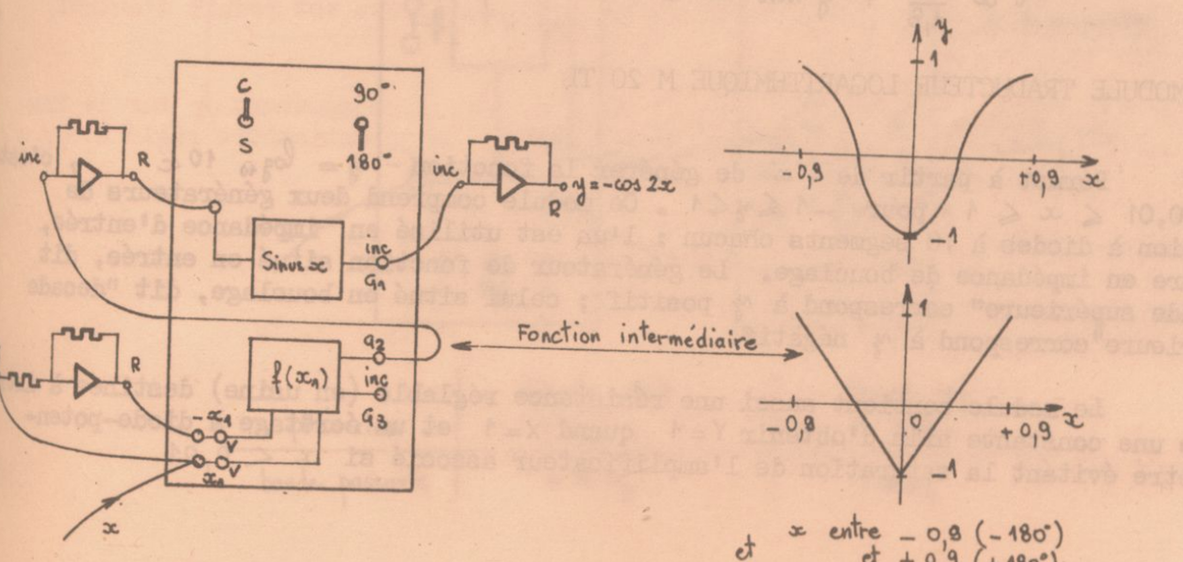
$y = -\cos 2x$ { en centaine de degré, compris entre 0,9 et 10,9 (–180 à + 180°)

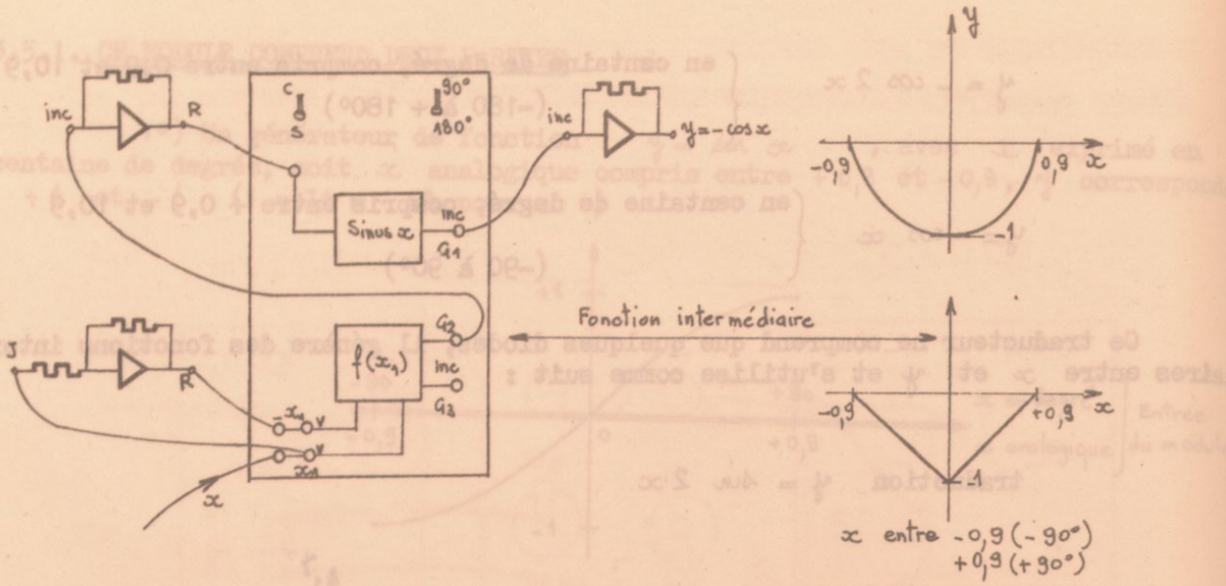
$y = -\cos x$ { en centaine de degré, compris entre + 0,9 et 10,9 (–90 à 90°)

Ce traducteur ne comprend que quelques diodes, il génère des fonctions intermédiaires entre x et y et s'utilise comme suit :

traduction $y = \sin 2x$







3.5.2. EXEMPLE D'UTILISATION

On a l'équation du pendule simple à grande amplitude :

$$l\,\theta'' + g \sin\theta = 0 \qquad \text{avec } \theta \text{ exprimé en radian.}$$

Si le terme $l\,\theta''$ dépend de l'unité d'angle choisie, le terme $g \sin\theta$ n'en dépend pas

L'écriture de cette équation utilise le radian comme unité d'angle, mais la machine <u>impose</u> la centaine de degré comme unité analogique : 100° = 1.

d'où la transformation en équation machine :

$$l\,x''\,\frac{\pi}{1{,}8} + g \sin x = 0 \qquad , \quad x \text{ en centaine de degré.}$$

3.6. MODULE TRADUCTEUR LOGARITHMIQUE M 20 TL

Permet à partir de $-x$ de générer la fonction $y = \log_{10} 10x$, c'es dire $0{,}01 \leqslant x \leqslant 1$ pour $-1 \leqslant y \leqslant 1$. Ce module comprend deux générateurs de fonction à diodes à 10 segments chacun : l'un est utilisé en impédance d'entrée, l'autre en impédance de bouclage. Le générateur de fonction situé en entrée, dit "décade supérieure" correspond à y positif ; celui situé en bouclage, dit "décade inférieure" correspond à y négatif.

Le module contient aussi une résistance réglable (en usine) destinée à int duire une constante afin d'obtenir $Y = 1$ quand $X = 1$ et un écrêtage à diode-potentiomètre évitant la saturation de l'amplificateur associé si $x < 0{,}01$

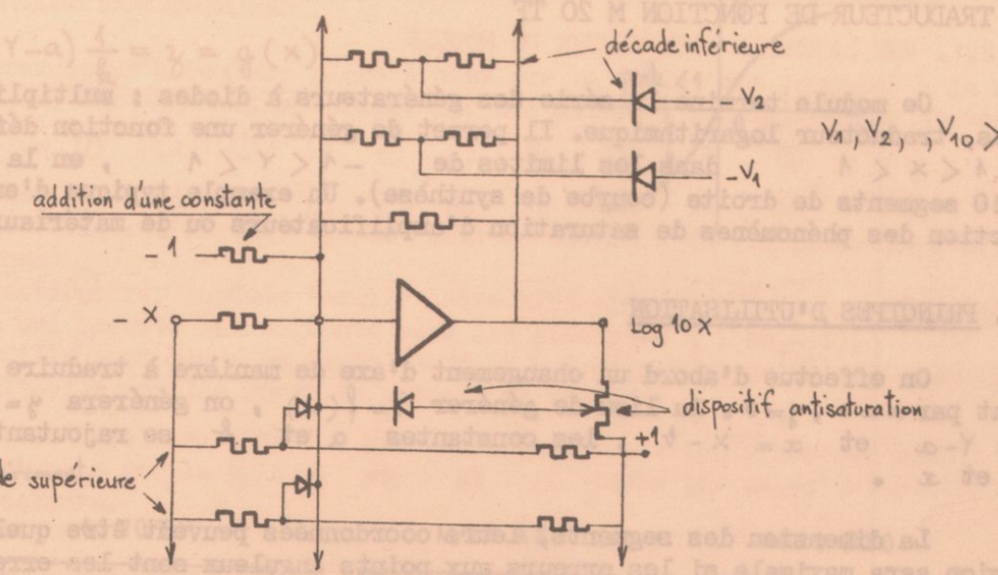
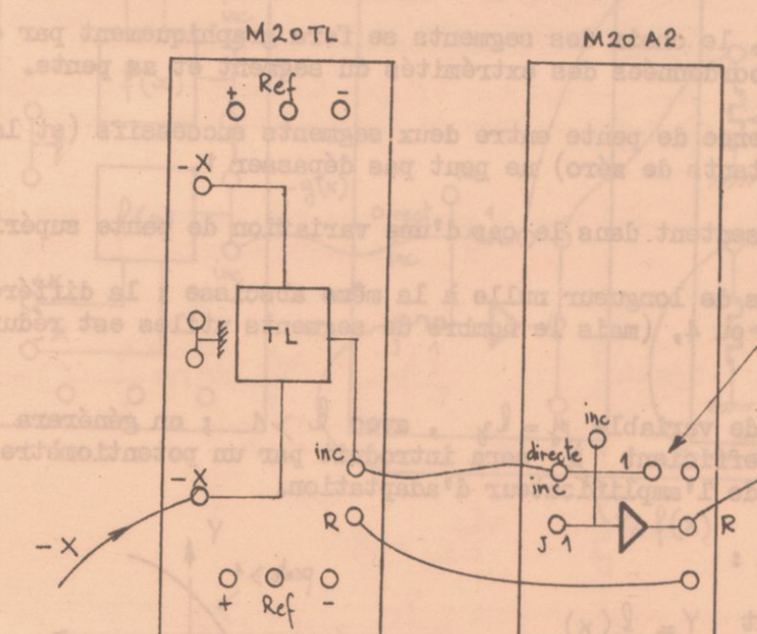


Schéma de principe simplifié (sans compensation température ni réglages)

Schéma d'utilisation



	décade sup.	décade inf.
précision	± 3 ‰	± 6 ‰
bande passante	4 kHz	3 kHz

3.7. TRADUCTEUR DE FONCTION M 20 TF

Ce module termine la série des générateurs à diodes : multiplieur, sinus cosinus, traducteur logarithmique. Il permet de générer une fonction définie pour $-1 < x < 1$ dans les limites de $-1 < Y < 1$, en la découpant en 5 ou 10 segments de droite (courbe de synthèse). Un exemple typique d'emploi est la traduction des phénomènes de saturation d'amplificateurs ou de matériaux magnétiques.

3.7.1. <u>PRINCIPES D'UTILISATION</u>

On effectue d'abord un changement d'axe de manière à traduire une courbe passant par $x = 0$, $y = 0$. Au lieu de générer $Y = f(x)$, on générera $y = g(x)$ avec $y = Y - a$ et $x = X - b$, les constantes a et b se rajoutant facilement à y et x.

La dimension des segments, leurs coordonnées peuvent être quelconques. La précision sera maximale si les erreurs aux points anguleux sont les erreurs maximums et sont toutes égales. Celà amène le choix de segments courts dans les régions de forte courbure, de segments longs dans les régions de faible courbure. Le calcul rigoureux, long, n'est possible que lorsque la courbeà traduire est définie mathématiquement.

En pratique, le choix des segments se fera graphiquement par essais successifs. On notera alors les coordonnées des extrémités du segment et sa pente.

<u>Attention</u> : la différence de pente entre deux segments successifs (et la pente du ou des deux segments partants de zéro) ne peut pas dépasser 1.

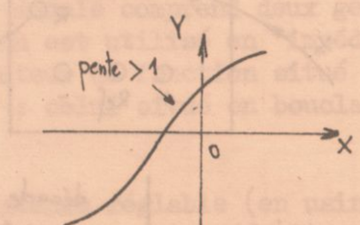
deux solutions se présentent dans le cas d'une variation de pente supérieure à 1 :

placer 2 ou 3 segments de longueur nulle à la même abscisse ; la différence de pente pourra atteindre 2, 3 ou 4, (mais le nombre de segments utiles est réduit d'autant, la précision diminue.

opérer un changement de variable $y = kz$, avec $k > 1$; on générera z sur le traducteur M 20 TF, le coefficient k sera introduit par un potentiomètre réglé à $1/k$ et monté en bouclage de l'amplificateur d'adaptation.

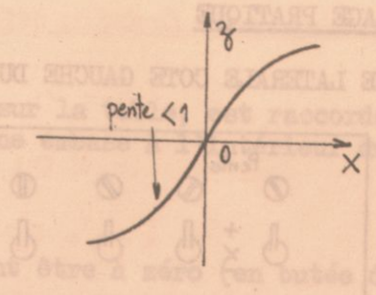
exemple d'utilisation :

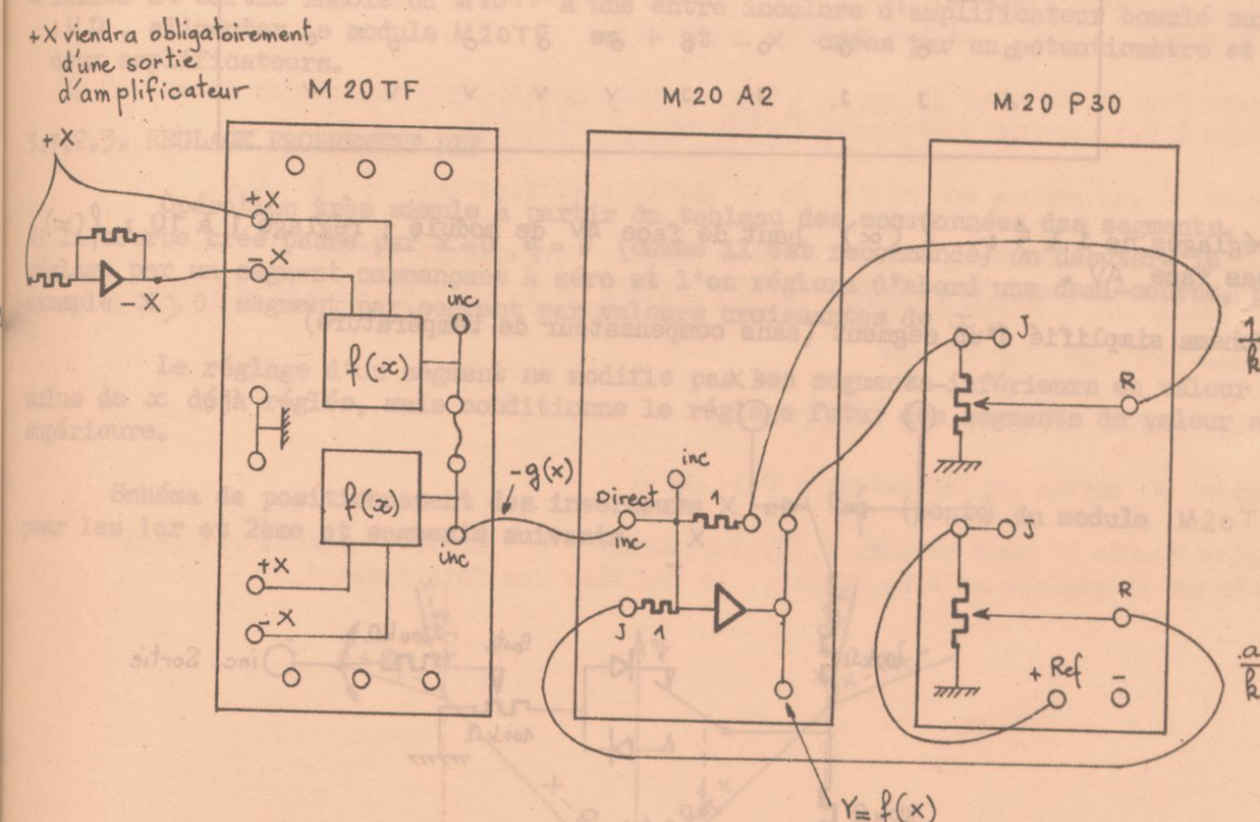
en 10 segments on veut $Y = f(x)$



on génère au traducteur

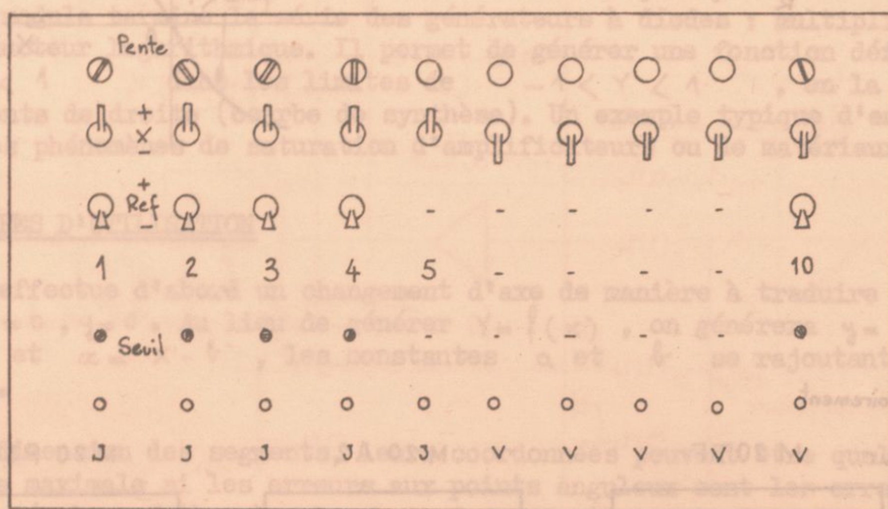
$$(Y - a)\,\frac{1}{k} = z = g(x)$$





3.7.2. REGLAGE PRATIQUE

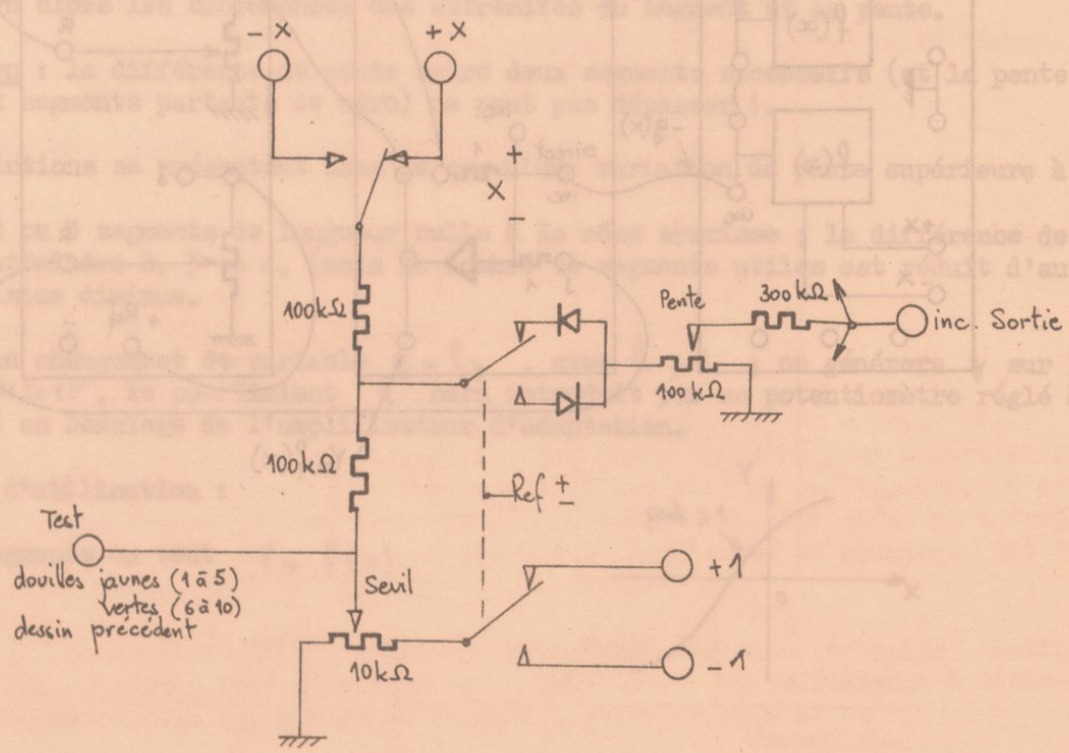
3.7.2.1. VUE LATERALE COTE GAUCHE DU MODULE



X viendra obligatoirement d'une sortie d'amplificateur

réglages de 1 à 5 : $f(x)$ haut de face AV de module ; réglage 1 à 10 : $f(x)$ bas face AV.

Schéma simplifié d'un segment (sans compensateur de température)



3.7,2.2. OPERATIONS PRELIMINAIRES

Le traducteur à régler, placé à plat sur la table, est raccordé à la machine par un cordon prolongateur souple enfiché sur une embase à l'intérieur du logement des modules.

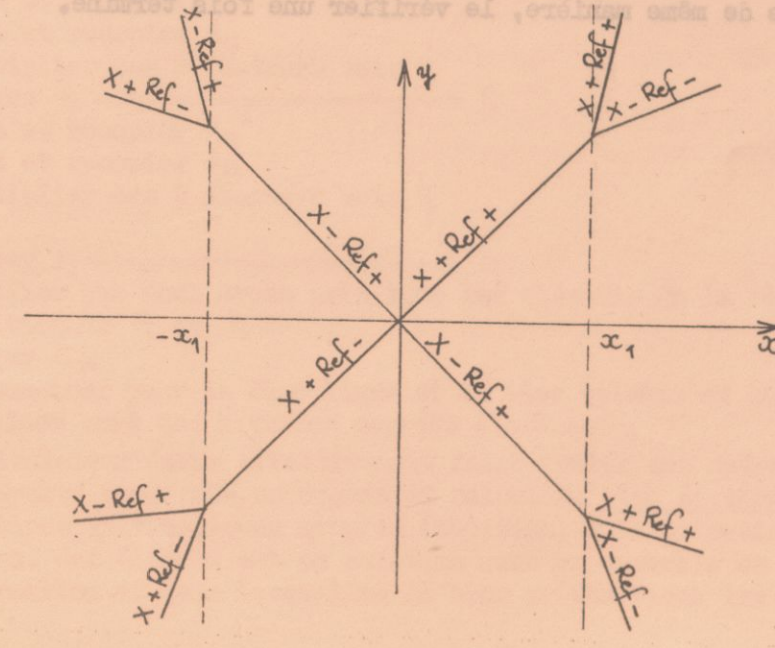
- machine en position V-I.
- tous les potentiomètres marqués "pente" doivent être à zéro (en butée dans le sens trigonométrique)
- tous les potentiomètres marqués "seuil" doivent être au maximum (en butée sens horaire) (ce qui reporte les segments vers les abscisses + 1 ou − 1).
- relier la douille test-jaune ou verte (sous le potentiomètre seuil) du segment à régler au voltmètre douille rouge du panneau de commande.
- relier la sortie incole du M20TF à une entre incolore d'amplificateur bouclé sur 1MΩ, alimenter le module M20TF en + et − X crées par un potentiomètre et deux amplificateurs.

3.7,2.3. REGLAGE PROPREMENT DIT

Opération très simple à partir du tableau des coordonnées des segments. Si la courbe crée passe par $x=0$, $y=0$ (comme il est recommandé) on débutera le réglage par un segment commençant à zéro et l'on réglera d'abord une demi-courbe, par exemple $X > 0$ segment par segment par valeurs croissantes de x.

Le réglage d'un segment ne modifie pas les segments inférieurs en valeur absolue de x déjà réglés, mais conditionne le réglage futur des segments de valeur absolue supérieure.

Schéma de positionnement des inverseurs X et Ref. (pente) du module M20TF pour les 1er et 2ème et segments suivants.



<u>Exemple de réglage de la fonction définie par les segments ;</u>

S_1 ($x_1 = 0$, $y_1 = 0$, $x_2 = 0{,}1$ $y_2 = 0{,}05$)

S_2 ($x_2 = 0{,}1$, $y_2 = 0{,}05$, $x_3 = 0{,}2$ $y_3 = 0{,}09$)

S_3 ($x_3 = 0{,}2$ $y_3 = 0{,}09$, $x_4 = 0{,}3$ $y_4 = 0{,}125$)

régler "seuil segment 1" au minimum (zéro) placer "X" sur + "Ref 1" sur + (voir schéma de positionnement)

régler X par mesure en AFF + de la sortie amplificateur X à 0,1 .

régler le potentiomètre "pente 1" du module pour obtenir 0,05 sur la sortie d'amplificateur Y (AFF +) le 1er segment est réglé.

placer " X 2" sur -, Ref 2" sur - (voir schéma positionnement) régler "seuil 2" du module à 0,1, mesuré sur "test 2" en AFF- .

régler X par mesure en AFF + de la sortie amplificateur X à 0,2.

régler le potentiomètre "pente 2" pour obtenir 0,09 sur la sortie Y (AFF +)
Le deuxième segment est réglé.

placer "X3" sur "Ref 3" sur -

régler "seuil 3" à 0,2 sur test (AFF-)

régler X sortie amplificateur à 0,3 (AFF+)

régler "pente 3" pour obtenir 0,125 sur sortie Y le 3ème segment est réglé.
Terminer le réglage de même manière, le vérifier une fois terminé.

TROISIEME PARTIE

PROGRAMMATION SUR CALCULATEUR NUMERIQUE P.B. 250

Chapitre I

LE CALCULATEUR NUMERIQUE

I.1. GENERALITES

1.1.1. <u>LES DIVERS ELEMENTS D'UN CALCULATEUR ELECTRONIQUE</u>

Un calculateur est destiné à remplacer l'homme dans l'élaboration des calculs numériques. Voyons, par exemple, comment un opérateur humain procède pour effectuer le produit de deux matrices carrées :

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Il sait que $C_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$

et exécute les opérations suivantes :

- lire et recopier A_{11}
- lire et recopier B_{11}
- multiplier ces 2 facteurs soit P_1
- ranger $P_1 \Longrightarrow P_1$
- lire et recopier A_{12}
- lire et recopier B_{21}
- multiplier ces 2 facteurs soit P_2
- ranger $P_2 \Longrightarrow P_2$
- vérifier que nous avons pris tous les éléments de la 1ère ligne.
- additionner P_1 et P_2
- ranger C_{11}
- recommencer pour la 2ème ligne et la 1ère colonne et cela N^2 fois si les matrices sont des matrices carrées d'ordre N.

Un calculateur devra effectivement faire toutes ces opérations, l'une après l'autre. Il comporte pour cela un organe de calcul ou <u>bloc arithmétique</u> pouvant effectuer des opérations arithmétiques simples (addition, multiplication...) et des opérations logiques. Cet élément est en relation avec un ensemble de <u>mémoires</u> qui conservent l'information et dans lesquelles le bloc arithmétique ira "lire" ou "ranger".

Pour communiquer avec l'opérateur, un calculateur comporte également des organes d'entrée-sortie destiné par exemple, à introduire les valeurs numériques des éléments des matrices A et B et sortir les valeurs des éléments de la matrice C.

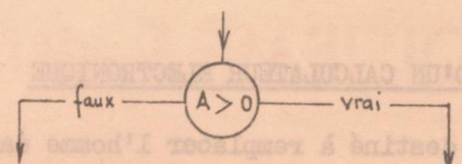
Le programme fixe à la machine la suite de toutes les opérations effectuées ci-dessus.

1.1,2. UTILISATION DES CALCULATEURS

D'une façon générale, la mise en oeuvre d'un calculateur pour un calcul simple, n'est pas recommandée. Par contre, le gain de temps est appréciable si le problème comporte des formes arithmétiques assez compliquées, et surtout si un même calcul doit être effectué un grand nombre de fois avec des valeurs différentes des variables (création de tables numériques).

1.1,2.1. NOTION DE PROGRAMME

En plus des calculs arithmétiques, un calculateur peut suivre un chemin ou un autre pré-déterminé d'après la réponse à un polynôme logique : vrai ou faux.



La suite (caractère séquentiel) des différentes instructions, opérations arithmétiques ou logiques introduction de données, sortie de résultats, etc... constitue le programme.

1.1,2.2. CONCEPTION D'UN LANGAGE SYMBOLIQUE

Le problème qui se pose est d'entretenir un dialogue avec la machine. Il faut pour cela disposer d'un langage de programmation qui sera le lien entre la machine et l'utilisateur.

Cette notion peut se comprendre à plusieurs niveaux :

- au niveau "technologie", la machine traite les informations sous leur forme la plus simple c'est-à-dire la configuration binaire.

Ex: Additionneur

- au niveau "machine", les constructeurs des calculateurs mettent au point des ordres plus évolués qui correspondent à des ensembles d'éléments simples (quelquefois des circuits câblés).

Ex. :
- vider la mémoire M
- loger le contenu de la mémoire M dans la mémoire A
- multiplier les contenus de la mémoire A par le contenu de la mémoire M

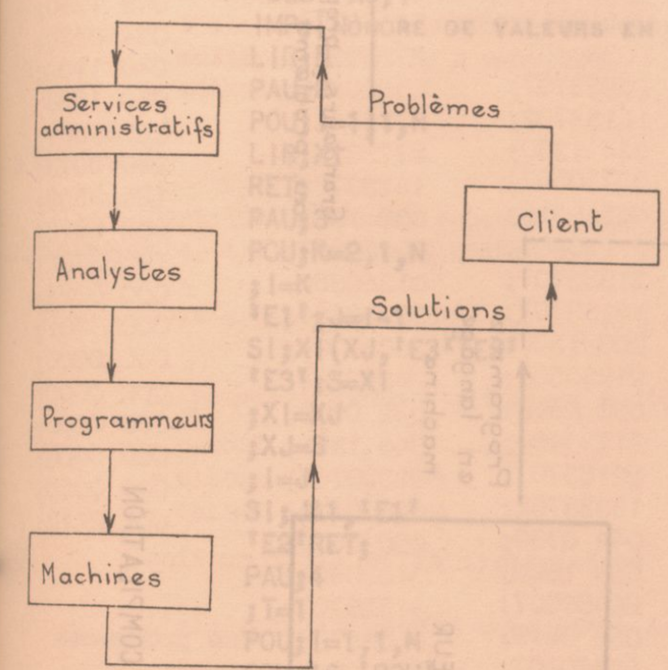
- au niveau "utilisateur", les différentes opérations ont conduit à la conception de langage symbolique type ALGOL, FORTRAN, PAF, COBOL, MAGE....etc.. qui sont très proches des expressions arithmétiques usuelles.

Ex.: Faire I = 2

Si A = 0 , alors B = C sinon aller à tel calcul.

Malheureusement, les langages symboliques sont inintelligibles par les calculateurs. Ils exigent une manipulation supplémentaire qui sera vue plus loin. En effet, la machine ne comprend qu'une seule sorte de langage appelé "langage machine" particulier à chaque constructeur et à chaque calculateur, dont l'écriture est longue et difficile.

1.1,2.3. GENERALISATION DE L'EMPLOI DES LANGAGES SYMBOLIQUES



Les calculateurs numériques ont d'abord été utilisés de la façon suivante : peu nombreux, ils étaient exploités par le constructeur dans des Centres de Calcul comprenant en plus des Services Administratifs, des analystes qui étudient les problèmes soumis par les clients et des programmeurs qui écrivaient les programmes à introduire dans les machines (généralement en langage machine).

Actuellement, les constructeurs lancent sur le marché de plus en plus de machines de petite et moyenne puissance, utilisables directement chez le client. Celui-ci est donc amené à analyser et à écrire lui-même ses programmes.

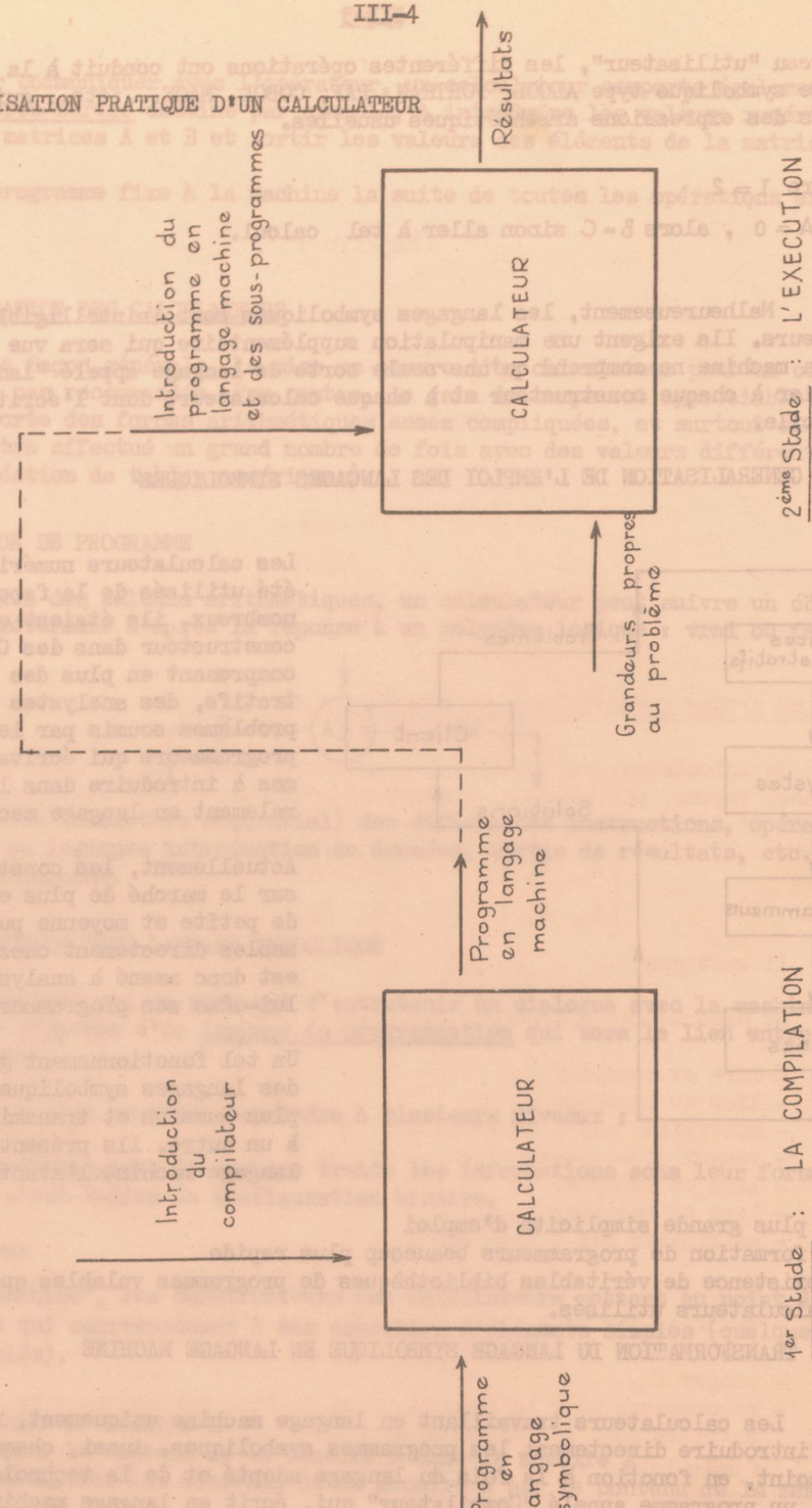
Un tel fonctionnement généralise l'emploi des langages symboliques considérés comme plus humains et transmissibles d'un pays à un autre. Ils présentent par rapport au langage machine l'avantage :

- d'une plus grande simplicité d'emploi
- d'une formation de programmeurs beaucoup plus rapide
- de l'existence de véritables bibliothèques de programmes valables quels que soient les calculateurs utilisés.

1.1,2.4. TRANSFORMATION DU LANGAGE SYMBOLIQUE EN LANGAGE MACHINE

Les calculateurs travaillant en langage machine uniquement, il est impossible d'introduire directement les programmes symboliques. Aussi, chaque constructeur met au point, en fonction à la fois du langage adopté et de la technologie de la machine, un programme appelé "Compilateur" qui, écrit en langage machine, traduit automatiquement le langage symbolique en langage machine.

1.1,2.5. UTILISATION PRATIQUE D'UN CALCULATEUR



PROGRAMME EN LANGAGE SYMBOLIQUE AVANT COMPILATION.

```
,1340
IMP;.
DIM;1001,XI,XJ
IND;N
'DEB'PAU;1
IMP;[NOMBRE DE VALEURS EN FIXE : ]
LIR;N
PAU;2
POU;I=1,1,N
LIR;XI
RET;
PAU;3
POU;K=2,1,N
;I=K
'E1';J=I-1
SI;XI(XJ,'E3''E2'
'E3';S=XI
;XI=XJ
;XJ=S
;I=J
SI;J&1,'E1'
'E2'RET;
PAU;4
;T=1
POU;I=1,1,N
COM;16,'ROU'
PER;(7,2)XI
'ROU'SI;T=5,'E4'
IMP;(7,2)XI
;T=T+1
VER;'E5'
'E4'IMP;(7,2)XI
;T=1
'E5'RET;
VER;'DEB'
FIN;
```

"Classement d'une suite de nombres dans l'ordre croissant".

```
00013C001 2613;
036S3707;
057 0013;
220S5250;
151 0013;
000S5200I
200 0013;
000S5244I
244 0013;
000S5240;
305 0013;
044 1415;
344 0013;
000S5201;
361 0013;
000S5205I
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
037S0507;
024S3705;
041S0407;
043S3707;
363S3706;
045S0607;
000 0053;
044 0441I
047S0407;
051S3707;
355S3703;
054S3707;
000S0177;
054 0530;
057S3707;
000S0177;
052 0530;
000 0021;
074S0607;
360 1256;
030 5050;
044S5204;
220S5204;
224S0240I
324S5002I
```

```
07013C024 4055;
344S5615I
100S4004;
360S6400;
061 5441I
076S0407;
100S3707;
355S3703;
102S0607;
000 0004;
101 0441I
104S0407;
106S3707;
355S3703;
107S0407;
111S3707;
025S3703;
050 1130;
000 0022;
114S0507;
000 0000I
120S3707;
117S1407;
000 0000I
054 1130;
122S0407;
124S3707;
025S3703;
000 1300;
126S0507;
052 0613;
130S0407;
132S3707;
163S3702;
054 0530;
040 1106;
050 1530;
136S0607;
116 0013;
126 3502;
000 0023;
142S0507;
000 0040;
146S3707;
145S1407;
000 0000I
046 1130;
046 0530;
054 1130;
054 0530;
153S1507;
000 0000I
052 1130;
156S0507;
052 0613;
160S0407;
```

```
162S3707;
161S3702;
040 1306;
164S0507;
055 0613;
166S0407;
170S3707;
161S3702;
171S0407;
173S3707;
061S3706;
176 3507;
006 0613;
126S3702;
010 0613;
126S3702;
201S0507;
052 0613;
203S0407;
205S3707;
161S3702;
044 1330;
207S0507;
055 0613;
211S0407;
213S3707;
161S3702;
000 1300;
215S0507;
052 0613;
217S0407;
221S3707;
163S3702;
044 0730;
000 1300;
224S0507;
055 0613;
226S0407;
230S3707;
163S3702;
052 0530;
054 1130;
052 0530;
104 0605;
040 1306;
000 0702;
237S0407;
241S3707;
071S3706;
244 3507;
004 0613;
126S3702;
046 0530;
040 1106;
050 1530;
250S0607;
```

```
144 0013;
126 3502;
000 0024;
000 0702;
042 1330;
256S0507;
000 0000I
262S3707;
261S1407;
000 0000I
054 1130;
264S0507;
200 0000;
267 7716;
000 4500;
272 3507;
012 0613;
126S3702;
273S0507;
052 0613;
275S0407;
277S3707;
161S3702;
000 1300;
301S0607;
013S0360;
303S0407;
305S3707;
243S3704;
042 0730;
040 1306;
310S0707;
000 0101;
050 0000;
313S0407;
315S3707;
065S3706;
320 3507;
014 0613;
126S3702;
321S0507;
052 0613;
323S0407;
325S3707;
161S3702;
000 1300;
327S0607;
017S0360;
331S0407;
333S3707;
304S3704;
042 0730;
000 1300;
000 0702;
337S0407;
341S3707;
```

```
011S3705;
042 1330;
016 0613;
126S3702;
345S0507;
052 0613;
347S0407;
351S3707;
161S3702;
000 1300;
353S0607;
013S0360;
355S0407;
357S3707;
304S3704;
000 0702;
042 1330;
054 0530;
040 1106;
050 1530;
365S0607;
260 0013;
126 3502;
002 0613;
126S3702;
000 0026;
000S3701;
000 0000;
000 0000;
000 0000;
000 0000;
000 0000;
001 2613;
```

Même programme en langage machine, après compilation. (Nous donnons ici le programme machine implanté dans la ligne 13)

1.2. LES DIFFERENTS STADES DE L'ETUDE D'UN PROBLEME

1,2,1. 1ère PHASE : ETUDE PREALABLE DU PROBLEME

Le problème étant posé sous la forme d'un énoncé complet, comportant éventuellement quelques formules, le premier travail consiste, par une recherche mathématique, à trouver un algorithme. On rappelle qu'un algorithme est une méthode de calcul se ramenant à une chaine d'opérations élémentaires, par exemple :

- l'algorithme d'HORNER (cf. plus loin)
- l'algorithme d'EUCLIDE (Rechercher du P.G.C.D. de 2 nombres)
- multiplication de 2 matrices
- méthodes de tri etc...

APPLICATION

Certaines méthodes peuvent amener des gains de temps appréciables. Citons comme exemple, l'algorithme d'HORNER :

Soit à calculer $P(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_n = \sum_{i=0}^{n} a_i x^{n-i}$

La première méthode qui vient à l'esprit est la suivante :

- calcul de $x^2, x^3, \ldots, x^n$ $\longrightarrow$ $n-1$ multiplications
- calcul des $a_i x^{n-i}$ $\longrightarrow$ n multiplications
- calcul de la somme $\longrightarrow$ n additions

soit un total de $(3n-1)$ opérations.

L'algorithme d'HORNER consiste à écrire le polynome sous la forme :

$$P(x) = (--(((a_0 \cdot x + a_1)x + a_2)x + ----+a_{n-1})x + a_n$$

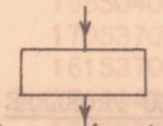
$\Longrightarrow$ n multiplications + n additions $\longrightarrow$ $2n$ opérations.

1,2,2. 2ème PHASE : ETABLISSEMENT D'UN ORGANIGRAMME

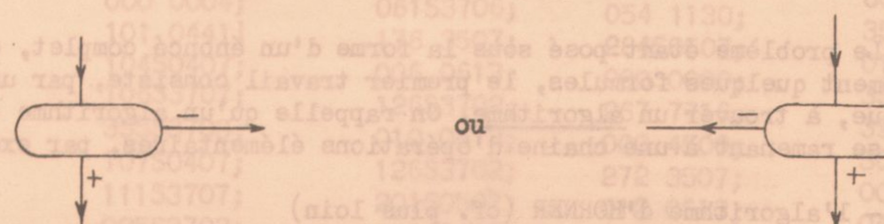
Cette phase n'est pas obligatoire. Elle facilite cependant l'écriture d'un programme car elle est en quelque sorte l'ossature des opérations. Un organigramme est la description schématisée de la suite des opérations arithmétiques et logiques qui conduisent à la résolution d'un problème.

REGLES D'ECRITURE :

- Une opération arithmétique est représentée par le symbole :



- L'indication d'un test est représenté par :

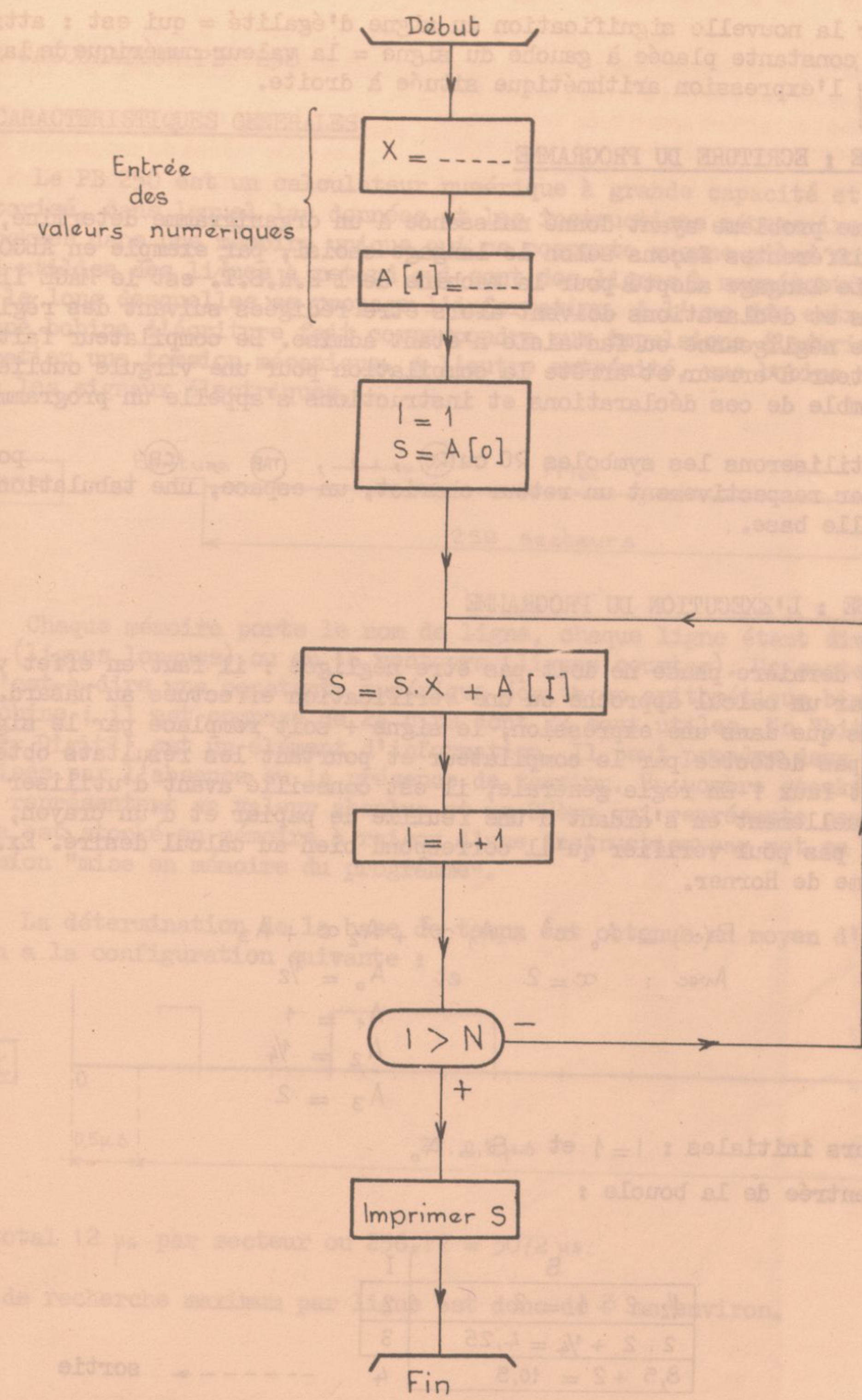


\+ correspondant à la réponse "OUI" ou "VRAI"

\- " " " "NON" ou "FAUX"

- L'indication d'une variable indicée A_1 est donnée par :

A [1]



Organigramme de l'Algorithme d'Horner

REMARQUE : Noter la nouvelle signification du signe d'égalité = qui est : attribuer à la constante placée à gauche du signe = la valeur numérique de la constan[te] ou de l'expression arithmétique située à droite.

1.2,3. 3ème PHASE : ECRITURE DU PROGRAMME

Un même problème ayant donné naissance à un organigramme déterminé, il peut être écrit de différentes façons selon le langage choisi, par exemple en ALGOL ou en FORTRAN (IBM). Le langage adopté pour la machine de l'E.N.S.T. est le MAGE II. Les instructions et déclarations doivent alors être rédigées suivant des règles très strictes, aucune négligeance ou fantaisie n'étant admise. Le compilateur fait d'ailleurs office de détecteur d'erreur et arrête la compilation pour une virgule oubliée, par exemple. L'ensemble de ces déclarations et instructions s'appelle un programme.

REGLES : Nous utiliserons les symboles RC ou (RC) , ␣ , (TAB) , (CB) pour désigner respectivement un retour chariot, un espace, une tabulation ou une corbeille base.

1.2,4. 4ème PHASE : L'EXECUTION DU PROGRAMME

Cette dernière phase ne doit pas être négligée : il faut en effet vérifier les résultats par un calcul approché ou une vérification effectuée au hasard. En effet, supposons que dans une expression, le signe + soit remplacé par le signe x, l'erreur n'est pas détectée par le compilateur et pourtant les résultats obtenus au calculateur sont faux ! En règle générale, il est conseillé avant d'utiliser la machine de suivre éventuellement en s'aidant d'une feuille de papier et d'un crayon, le programme pas à pas pour vérifier qu'il correspond bien au calcul désiré. Ex.: Toujours avec l'algorithme de Horner.

$$P(x) = A_0 x^3 + A_1 x^2 + A_2 x + A_3$$

Avec : $x = 2$ et $A_0 = 1/2$

$A_1 = 1$

$A_2 = 1/4$

$A_3 = 2$

Valeurs initiales : $I = 1$ et $S = A_0$

A l'entrée de la boucle :

S	I
$\frac{1}{2} \cdot 2 + 1 = 2$	2
$2 \cdot 2 + \frac{1}{4} = 4{,}25$	3
$8{,}5 + 2 = 10{,}5$	4

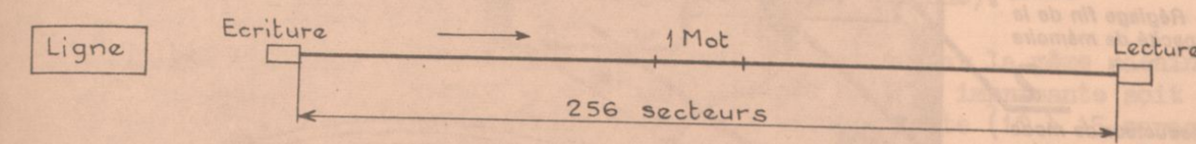
-------→ sortie

Avant d'écrire le programme sur cet exemple adopté depuis le début, il va falloir maintenant étudier le langage utilisé : MAGE II et le calculateur : PB 250.

I.3. LE CALCULATEUR PB 250

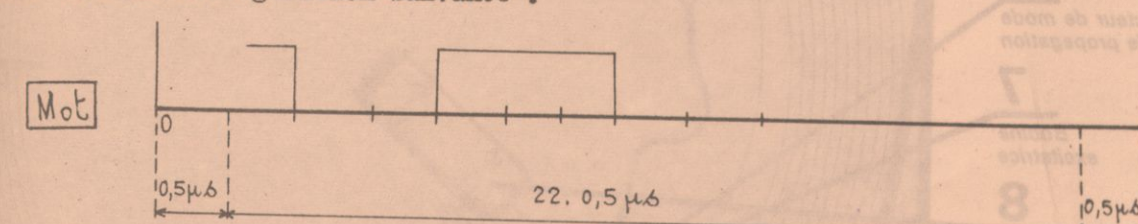
1.3,1. CARACTERISTIQUES GENERALES

Le PB 250 est un calculateur numérique à grande capacité et entièrement transistorisé, dans lequel les données et les instructions nécessaires au calcul sont stockées dans une mémoire unique qui ne comporte aucune pièce en mouvement. Celle-ci utilise des lignes à retard qui sont des lignes à magnétostrictions en nickel, le long desquelles se propage l'information. A l'une des extrémités de chaque ligne, une bobine d'écriture fait correspondre aux impulsions électriques support de l'information une torsion mécanique. A l'autre extrémité, une bobine de lecture restitue les signaux électriques.



Chaque mémoire porte le nom de ligne, chaque ligne étant divisée en 256 secteurs (lignes longues) ou en 16 secteurs (lignes courtes). Un secteur représente un mot c'est-à-dire une constante numérique écrite en arithmétique binaire ou un ordre machine ; il est composé de 24 bits dont 22 sont utiles. Un "bit" (contraction de "Binary Digit") est un élément d'information. Il peut prendre deux états qui sont caractérisés par l'absence ou la présence de tension. Un nombre stocké comporte 21 bits qui représentent sa valeur absolue et un 22ème qui représente son signe. Le programme est stocké en mémoire à raison d'une instruction par mot ce qui justifie l'expression "mise en mémoire du programme".

La détermination de la base de temps est obtenue au moyen d'une horloge de 2 MHz. On a la configuration suivante :



soit au total 12 µs par secteur ou 256.12 = 3072 µs.

le temps de recherche maximum par ligne est donc de 3 ms environ.

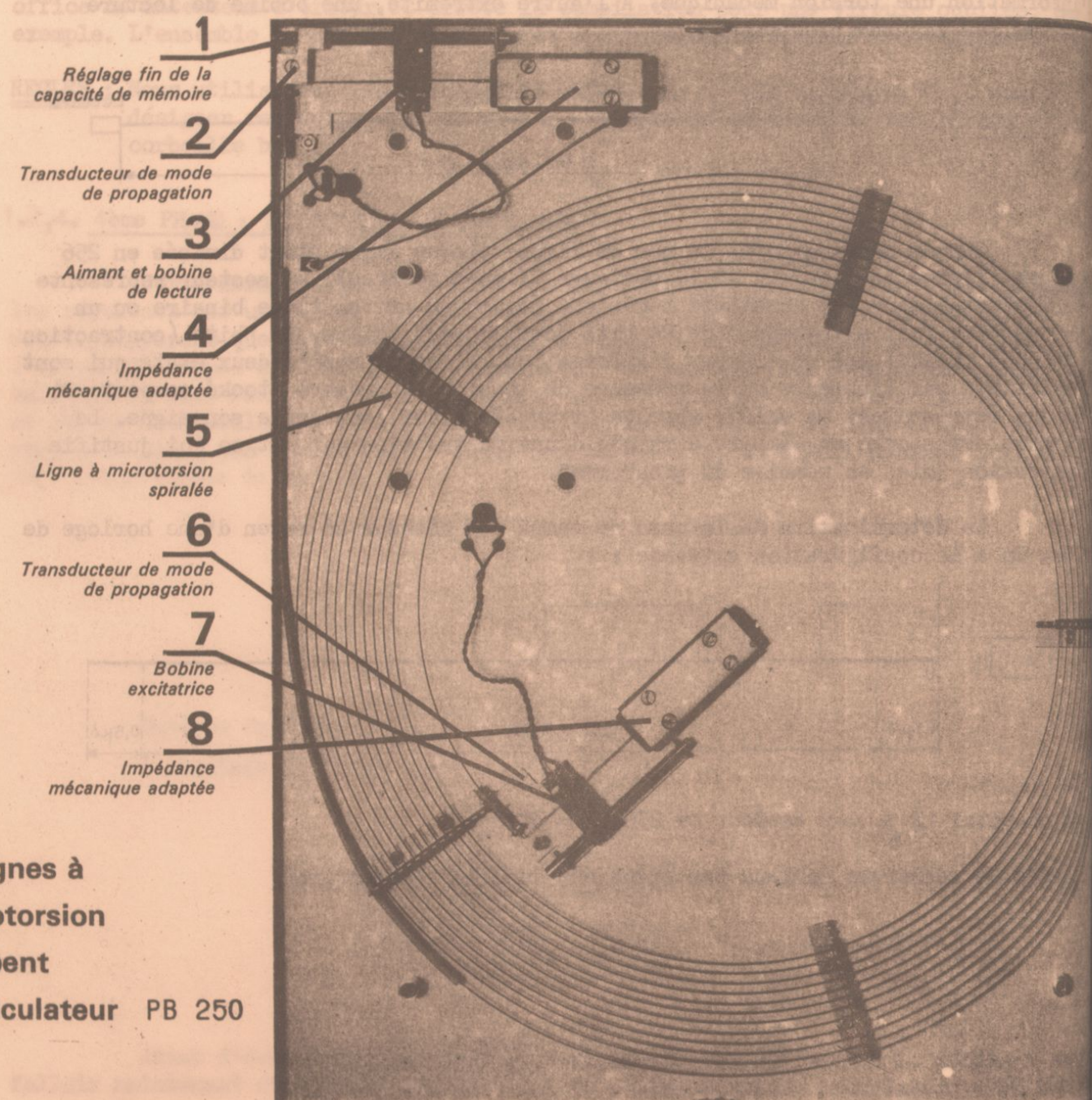
ligne à microtorsion

Mémoire électronique rapide de haute fiabilité utilisée dans tous les cas délicats où les conditions physiques rendent très difficiles le bon fonctionnement des mémoires classiques : engins embarqués, satellites, fusées sous-marines, avions supersoniques, analyseurs de vidéo radar, etc.

Endurance aux chocs et vibrations : jusqu'à 20 g et 2.000 Hz
Consommation : 40 mW pour 1.000 bits d'information
Température : pratiquement sans effet
Capacité : 7.000 bits
Fréquence de travail : jusqu'à 4 M Hz.

A été sélectionné pour :

La mémoire du satellite Mariner II
Les sous-marins du programme SUBROC
Le système RADFAC de l'OTAN
Les balises maritimes TACAN
Les radars Sperry F P S
Les radars du système RAYTHEON ASSR-2



les lignes à microtorsion équipent le calculateur PB 250

1.3.2. <u>LE CALCULATEUR NUMERIQUE DE L'E.N.S.T.</u>

Actuellement le calculateur de l'E.N.S.T. est équipé avec une extension de mémoire, ce qui donne une capacité d'environ 8000 mots.

Les éléments périphériques sont :

<u>Unités d'entrées</u> : les organes sont :

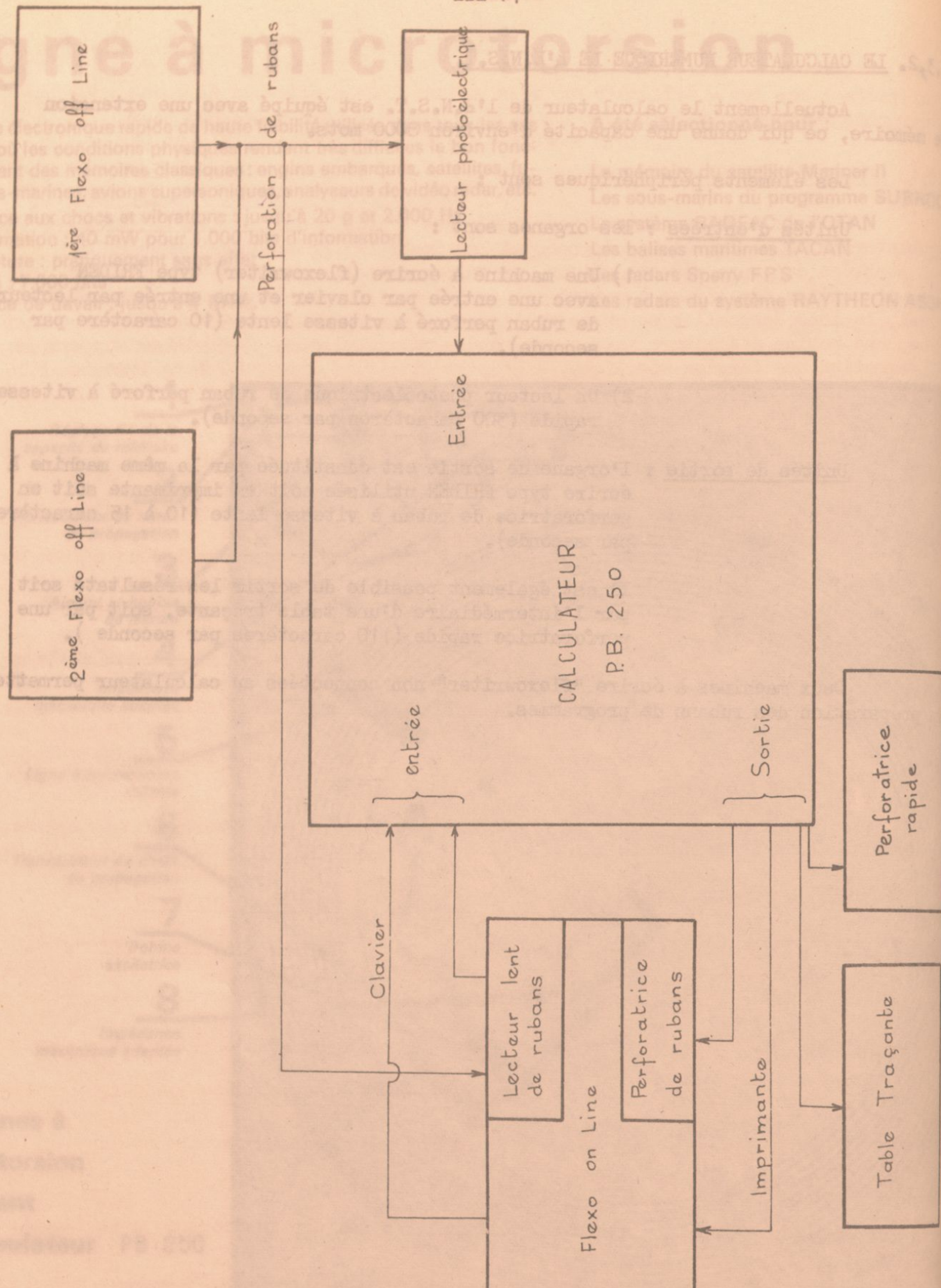
1) Une machine à écrire (flexowriter) type FRIDEN avec une entrée par clavier et une entrée par lecteur de ruban perforé à vitesse lente (10 caractère par seconde).

2) Un lecteur photoélectrique de ruban perforé à vitesse rapide (300 caractères par seconde).

<u>Unités de sortie</u> : l'organe de sortie est constituée par la même machine à écrire type FRIDEN utilisée soit en imprimante soit en perforatrice de ruban à vitesse lente (10 à 15 caractères par seconde).

Il est également possible de sortir les résultats soit par l'intermédiaire d'une table traçante, soit par une perforatrice rapide (110 caractères par seconde).

Deux machines à écrire "flexowriter" non connectées au calculateur permettent la préparation des rubans de programmes.



Chapitre II

LE LANGAGE MAGE II

(METHODE D'ASSEMBLAGE DE GRANDE EFFICACITE)

2.1. LA SYNTAXE

Un programme écrit en langage symbolique "MAGE" se présente sous la forme d'une suite de phrases qui sont soit des instructions soit des déclarations.

2.1.1. LA PHRASE MAGE

Elle se décompose en :

"ETIQUETTE"	TYPE DE PHASE	:	CORPS DE PHASE	RC

2.1.1.1. L'ETIQUETTE

Elle permet de localiser une phrase et sert d'adresse si on désire, à un certain moment, aller directement à une partie déterminée du programme : elle est donc facultative. Une étiquette s'écrit toujours entre guillemets : "ETI"et se place en début de phrase.

2.1.1.2. LE TYPE DE PHRASE

C'est un mot qui précise la nature du traitement effectué ou la nature des variables qui suivent. Le type de phrase se termine toujours par deux points ou un point virgule (: ou ;).

2.1.1.3. LE CORPS DE LA PHRASE

Il consiste soit en une suite d'opérations à effectuer soit en une liste de variables.

2.1,1.4. RETOUR CHARIOT (RC)

C'est un code qui termine obligatoirement toute phrase MAGE.

2.1,2. LE VOCACULAIRE

Le vocabulaire utilisé pour désigner les :

- variables
- étiquettes
- type de phrases
- fonctions

est formé d'un groupe de caractères alphanumériques, de longueur quelconque mais dont seuls les trois premiers permettent de les différencier. Exemple : x, ARG, SIN, "UN", mais ALPHA et ALPHABET représentent une seule et même variable.

Les types de phrases et un certain nombre de fonctions sont fixés par le langage, le reste étant au choix du programmeur. Les identificateurs des quatre ensembles précédents sont considérés séparément par le compilateur de sorte qu'on peut utiliser le même nom pour désigner par exemple, une variable, une étiquette et un type de phrase.

2.1,3. LES VARIABLES

2.1,3.1. NOTE SUR L'ECRITURE DES NOMBRES DANS LE CALCULATEUR

Différence entre virgule flottante et virgule fixe

Nous avons vu qu'un "mot" comprenait 22 bits. En fait, on peut grouper deux mots pour augmenter la précision de la machine et disposer ainsi de 44 bits. L'écriture d'un nombre peut se faire de deux façons différentes :

1) Ecriture en virgule fixe :

Il s'agit d'un nombre entier traduit en binaire et précédé d'un signe + ou -.

	2^{20}					2^{1}	2^{0}
+	1	0	0	1		1	0

La valeur maximale de ce nombre est $2^{21} - 1 = 2\,097\,151$

2) Ecriture en virgule flottante :

L'écriture normalisée d'un nombre en virgule flottante est :

$\pm$ Mantisse $\pm$ Exposant

La mantisse ne comportant que les chiffres significatifs et l'exposant étant une puissance de 10.

Exemples :			
	+ 3,14159	s'écrit	+ . 314 159 E + 1
	+ 0,015	s'écrit	+ . 15 E - 1
	- 1534,2	s'écrit	- . 15342 E + 4

Cette manière d'écrire s'effectue généralement sur 44 bits (2 mots) de la façon suivante :

±	Mantisse	±	Exposant
	← 35 bits →		← 7 Bits →

Comme $2^{35} \# 10^{10}$, la mantisse s'écrit avec dix chiffres. Comme $2^{7} = 128$, l'exposant est compris entre $+ 2^{-128}$ et $+ 2^{128}$ ce qui correspond en décimal à un intervalle compris entre :

$$10^{-38} \quad \text{et} \quad 10^{+38}$$

Cette différence entre les nombres écrits en fixe ou en flottante est d'une grande importance dans l'écriture d'un programme.

2.1,3.2. VARIABLES ECRITES EN VIRGULE FLOTTANTE

A - Les variables simples

Elles sont identifiées par 1 à 3 caractères alphanumériques, le premier étant obligatoirement une lettre :

X , THE , R3 , K2Z

N.B. : Les chiffres 0 à 9 peuvent être utilisés comme variables simples ; ils représentent alors leur propre valeur numérique.

B - Les variables indicées

Elles peuvent avoir un ou deux indices.

1) Variables à un indice :

L'ensemble doit être composé de trois caractères alphanumériques au plus, le dernier représentant l'indice d'où les différentes formes :

A_I s'écrit AI

AB_J s'écrit ABJ

2) Variables à deux indices (éléments d'une matrice, par exemple) Elles sont obligatoirement de la forme :

$A_{IJ} \Longrightarrow$ AIJ

2.1,3.3. VARIABLES ECRITES EN VIRGULE FIXE

Ces variables s'appellent des "indices" et sont toujours désignées par une seule lettre. Elles ne peuvent représenter que des nombres entiers (positif, négatif ou nul).

Remarques : Une même lettre ne peut servir à désigner un indice et une variable simple même s'il s'agit de la même valeur. Il faut alors utiliser deux identificateurs différents (par exemple, I et IF).

2.1,3.4. UTILISATION DES VARIABLES INDICEES

Lorsqu'on utilise une variable indicée, par exemple AIJ et J sont des indices au sens du paragraphe 2.1,3.3. qui de plus, ne peuvent avoir que des valeurs supérieures ou égales à 1, en aucun cas, nulles. Il est interdit d'utiliser directement une variable indicée sous la forme A_{23} par exemple ; il faut d'abord donner à I la valeur 2, à J la valeur 3 et utiliser ensuite la variable AIJ .

2.1,4. LES CONSTANTES

Comme pour les variables, les constantes peuvent s'écrire en virgule fixe ou en virgule flottante mais elles sont toujours positives.

2.1,4.1. CONSTANTES EN VIRGULE FLOTTANTE

1) Les chiffres 0 à 9 déjà décrits plus haut. On les écrit tels quels.

2) Toutes les autres valeurs s'écrivent toujours avec une virgule décimale et non un point, et entre deux espace (␣).

Exemples : ␣ 3,14 ␣ $\longleftrightarrow$ + 3,14
- ␣ 17,5 ␣ $\longleftrightarrow$ - 17,5 et non pas ␣ - 17,5 ␣
␣ 50, ␣ $\longleftrightarrow$ + 50

Remarquons qu'il faut écrire la virgule décimale même s'il s'agit d'un nombre entier.

2.1,4.2. CONSTANTES EN VIRGULE FIXE

Elles s'écrivent sans point ni virgule décimal.

Exemples : 5
117

En résumé :

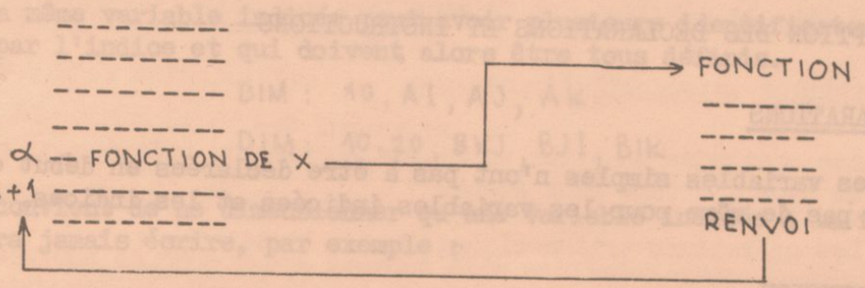
	Variables	Constantes
Virgule flottante	Simples : X, SIN, AR3 Indicées : AI BIJ	0 1 2 .... 9 ␣ 3,14 ␣ - ␣ 2,7 ␣ ␣ 50, ␣
Virgule Fixe	(Indices) A, B, ... Z	5 117

2.1,5. LES FONCTIONS

Les fonctions fixées pour le langage MAGE sont :

FONCTIONS	REPRESENTATION "MAGE"
- Racine carrée	√
- Exponentielle	EXP
- Logarithme népérien	LNE
- Logarithme décimal	LDE
- Sinus	SIN
- Cosinus	COS
- Tangente	TAN
- Arc tangente	ATG
- Valeur absolue	ABS

Les fonctions sont en fait des sous-programmes rédigés en langage machine et logés de façon constante dans le calculateur. Leur utilisation est la suivante : une instruction de "fonctions" étant située à une certaine adresse α , le calculateur transfère la valeur du paramètre au sous-programme voulu qui calcule la valeur numérique de la fonction et se renvoie à l'adresse $\alpha + 1$.



2.1.6. OPERATEURS

A - Opérateurs arithmétiques :

- Addition	+
- Soustraction	-
- Multiplication	.
- Division	/
- Puissance entière positive	*

B - Opérateurs logiques et de relations :

- Supérieur	)
- Inférieur	(
- Egal	=
- Différent	&
- Fonction ET	Ω
- Fonction OU	╥

2.1.7. TYPES DE PHRASES

Cette catégorie contient les instructions d'entrée et de sortie, de calcul, de rupture de séquence. Elle contient également les diverses autres déclarations.

2.2. DESCRIPTION DES DECLARATIONS ET INSTRUCTIONS

2.2.1. DECLARATIONS

Les variables simples n'ont pas à être déclarées en début de programme mais il n'en est pas de même pour les variables indicées et les indices.

2.2.1.1. DIMENSION

A) Ecriture en MAGE : DIM : N , AI où N est obligatoirement un nombre. Dans un problème, il est fréquent de rencontrer des variables indicées.

Ex.: - les composantes d'un vecteur $\vec{A}\ (A_1, A_2, \ldots A_i, \ldots)$
- les éléments d'une matrice $\|M\|\ (M_{11}, M_{12}, \cdots M_{ij}, \cdots)$

Nous avons vu comment sont utilisées les variables indicées dans un programme. En fait, dans la mémoire principale d'un calculateur, les différentes valeurs numériques d'une variable indicée sont toujours dans une série de mémoires successives afin d'être transmises au bloc de calcul, au moment opportun.

Il est donc nécessaire dès le début du programme de réserver un certain nombre de mémoires. C'est le but de l'instruction DIM : dans laquelle le corps de phrase précise le nom de la variable indicée et le nombre maximum de mémoires à réserver. Il est à noter qu'il n'est pas obligatoire de les remplir toutes.

Ex.: DIM : 100 , AI réserve cent mémoires. Si AI n'a que 20 composantes, les mémoires 21 à 100 sont inutilisées. La 14ème mémoire, par exemple, contient la valeur numérique de AI pour I = 14 .

La variable indicée AI n'est plus disponible pour désigner une autre variable. Par contre, on peut utiliser la variable A et l'indice I .

Ex.: DIM : 100 , AI

DIM : 100 , BI

B) Par extension aux variables à deux indices, l'écriture MAGE est :

DIM : M.N.AIJ où M et N sont <u>obligatoirement</u> des nombres qui sont séparés par un point.

Ex.: DIM : 20.10, AIJ avec $1 \leqslant I \leqslant 20$, $1 \leqslant J \leqslant 10$

La machine réserve dans ce cas 20.10=200 mémoires.

C) Une variable indicée AI est toujours écrite en virgule flottante, alors que l'indice est un INDICE au sens M.A.G.E.

Ex.: AI = 2,7 mais I = 1,2,3,...

D) La même variable indicée peut avoir plusieurs identificateurs qui ne diffèrent que par l'indice et qui doivent alors être tous définis.

Ex.: DIM : 10 , AI , AJ , AK

DIM : 10.20 , BKJ , BJI , BIK

Remarque : Il convient de ne dimensionner qu'une variable indicée à la fois. On ne devra jamais écrire, par exemple :

DIM : 50 , AI , BI où AI et BI sont deux variables indicées distinctes.

2.2.1.2. INDICE

Contrairement aux variables ordinaires, les indices doivent faire l'objet d'une déclaration si l'instruction DIM : ne les a pas déjà identifiés comme tels parmi la ou les deux dernières lettres d'une variable indicée. Nous verrons plus loin deux autres instructions qui permettent d'identifier des indices.

Cependant, il existe deux cas pour lesquels les indices ne sont pas identifiés et où il faut alors les déclarer en tête de programme, par la déclaration IND : I , J , L ...

Ces deux cas sont :

a) introduction d'un indice par un organe d'entrée.
b) calcul d'indice. Par exemple, soient I et J deux indices déclarés dans une déclaration DIM : Pour calculer leur somme L , L devra être déclaré en indice.

2.2.2. <u>INSTRUCTIONS DE CALCUL</u>

Cette instruction a pour type de phrase <u>CAL</u> : mais on peut omettre CAL et écrire simplement les deux points

2.2.2.1. SIGNIFICATION DYNAMIQUE DU SIGNE =

Le corps de la phrase à pour syntaxe :

nom du résultat = expression arithmétique

Dans ce cas particulier, le signe = signifie que l'expression de droite permet de calculer une certaine valeur arithmétique et cette valeur est attribuée à la variable de gauche.

<u>EX.</u>: : Y = A

L'expression arithmétique peut être du type flottant ou du type fixe mais dans les deux cas, le résultat est du même type que dans l'expression arithmétique.

2.2.2.2. EXPRESSION ARITHMETIQUE FLOTTANTE

- Les opérandes sont des variables simples ou indicées ou bien des constantes du type flottant sans signe.
- Les opérateurs arithmétiques sont : l'addition +
  la soustraction -
  la multiplication .
  la division /
  la puissance entière (au sens mathématique), positive *

Lorsqu'on écrit A^N, le calculateur effectue en fait A.A.A...A (N-1) fois, donc le calculateur ne peut calculer A^0 (A puissance zéro). Pour calculer

$A^{p/q}$ il faut écrire : $A^{p/q} = e^{p/q \log_e A}$ $(A > 0)$

- <u>Priorité des facteurs</u> :

Une expression arithmétique est lue et exécutée de gauche à droite, chaque opérateur arithmétique ayant pour premier opérande tout ce qui précède et pour deuxième opérande celui qui lui succède immédiatement.

Ex.: $a.x+b$ s'écrit A.X+B mais B+A.X signifie $(b+a)x$

De même X*2 ne signifie pas Ax^2 mais $a^2.x^2$ cette forme d'écriture permet d'ailleurs d'écrire très simplement un polynôme suivant la méthode de Horner :

$$P_4(x) = a_0 x^4 + a_1.x^3 + a_2.x^2 + a_3.x + a_4$$

s'écrit :

A0.X + A1.X + A2.X + A3.X + A4

Dans certains cas, il sera nécessaire de décomposer l'expression arithmétique.

<u>Ex.1</u>: calculer $\Delta = \dfrac{b^2 - 4ac}{2a}$ s'écrit :

: DEL = 4.A.C
: DEL = B.B - DEL/2/A

(à noter l'utilisation de la variable DEL)

<u>Ex.: 2</u>

$$R = \frac{a}{C - 3,7}$$

; IN1 = C - ⊔ 3,7 ⊔

: R = A/IN1

1. On utilise la variable intermédiaire IN 1.
2. On n'écrit pas ⊔-3,7⊔ mais -⊔3,7⊔
3. La forme ; Y= A. -⊔10,⊔ est interdite.

<u>Utilisation des fonctions</u> :

Les fonctions et procédures peuvent être utilisées comme opérandes dans une expression arithmétique. Les règles auxquelles obéissent les fonctions sont les suivantes :

1. La fonction est suivie de l'argument ou paramètre effectif entre crochets.

Ex.: EXP [X]

2. Les arguments des fonctions trigonométriques sont exprimés en radians.
3. L'argument de la fonction peut être une variable simple ou indicée mais non une constante ou une expression arithmétique ou un indice.

Ex.:

MAUVAIS	BON
; Y= LDE [2,7]	; INT = ⊔ 2,7 ⊔ : Y= LDE [INT]
: Z= SIN [OME.T+PHI]	: ARG = OME.T+PHI : Z= SIN [ARG]

Les variables en virgule flottante ne doivent pas être déclarées en début de programme mais toute variable utilisée dans la partie droite de la phrase doit avoir été utilisée auparavant au moins une fois :

- dans une déclaration DIM :
- à gauche du signe =
- dans la liste des variables d'une instruction d'entrée.

2.2.2.3. EXPRESSION ARITHMETIQUE EN FIXE

A) Les opérandes écrits à droite comme à gauche du signe = sont des expressions en fixe de deux types :

- les indices désignés par une seule lettre

Ex.: I, J, K...

- les constantes en fixe, sans point (ni virgule) décimal.

Ex.: 3 21 101

Les indices doivent avoir été définis comme tels par une des déclarations suivantes :

- DIM : 20, ABI — I est un indice
- IND : N
- POU : K = ...
- ENT :

(Ces deux instructions seront définies plus loin)

Ex.: IND : N
DEM : N
Instruction de calcul en fixe portant sur N

B) Les opérations

Les seules opérations en virgule fixe sont :

l'addition +
la soustraction –
la multiplication.

- La règle de priorité est la même qu'en virgule flottante.
- Une expression arithmétique en fixe ne peut pas commencer par le signe -

Ex.: R= (I_L) K_21

s'écrit en MAGE : R= I_L .K_21

REMARQUES : 1. Exemple : I = I+1 peut s'écrire : I+1
2. Attention : R = I/J est interdit
3. R = – 10 . G s'écrit : R = 0 – 10 . G

2.2.3. CONVERSION DES NOMBRES ECRITS EN VIRGULE FLOTTANTE EN INDICES ET INVERSEMENT

Il existe deux instructions qui permettent de passer d'une écriture à une autre.

2.2.3.1. ENT :

qui signifie : prendre la partie entière d'une expression flottante et la convertir en fixe.

Ex.: ENT : I = ALP+ INJ. DEL

- I est un indice.
- I peut ne pas avoir encore figuré dans le programme. Après l'instruction ENT : il sera considéré comme un indice.
- l'expression arithmétique est écrite en convention virgule flottante.

Supposons que la valeur numérique de l'expression arithmétique soit 23,87. I prendra la valeur numérique 23. Le calculateur ne fait pas d'arrondi à l'entier supérieur ; il convient d'écrire alors :

ENT : I = ALP + INJ. DEL + ␣ 0,5 ␣

auquel cas I prendra la valeur 24.

REMARQUE : Le sous-programme ENT : a été écrit de façon telle qu'on ne peut convertir en fixe les nombres supérieurs à

$2^{19} - 1$ c'est-à-dire supérieurs à 524287

2.2.3.2. FLO :

La syntaxe de cette expression est :

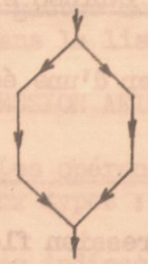
variable simple ou indicée = expression arithmétique
(convention virgule flottante) (convention virgule fixe)

Ex.: FLO : ALP = I + 21 . J

REMARQUE : Les expression arithmétiques à droite du signe = obéissent dans les instructions ENT : et FLO : aux mêmes règles que dans l'instruction CAL :

2.2,4. INSTRUCTIONS DE RUPTURE DE SEQUENCE

Au cours de l'exploitation, l'orientation d'un problème peut dépendre d'une certaine condition attachée aux valeurs des variables.



Ex.: Résolution de l'équation du 2ème degré :

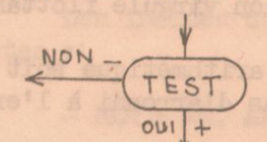
Si $b^2 - 4ac > 0$ 2 racines réelles et distinctes

Si $b^2 - 4ac = 0$ 2 racines réelles et confondues

Si $b^2 - 4ac < 0$ 2 racines imaginaires.

On doit alors effectuer un choix entre les différentes séquences de calcul : on dit qu'il y a rupture de séquence conditionnelle. Dans d'autres cas, on peut vouloir sauter systématiquement une partie du programme ou se renvoyer à un certain point situé en avant, dans le programme : on dit qu'il y a rupture de séquence inconditionnelle.

2.2,4.1. RUPTURE DE SEQUENCE CONDITIONNELLE



Rappelons l'utilisation de cette instruction déjà utilisée dans la définition d'un organigramme.

Le test est en fait une expression logique qui ne peut donc avoir que deux réponses : vrai ou faux, oui ou non. Selon la nature de la réponse à l'expression logique, le calculateur exécute le programme à partir de deux adresses différentes, c'est-à-dire à partir de deux étiquettes différentes.

I - INSTRUCTION SI :

L'écriture de cette instruction en MAGE est la suivante :

SI : Polynôme logique, "1ère étiquette" "2ème étiquette"

A) la virgule (,) qui indique la fin du polynôme logique est indispensable.

B) Il n'existe par contre aucun séparateur entre la 1ère et la 2ème étiquette.

C) Les étiquettes sont toujours encadrées par des guillemets.

Si la réponse au polynôme logique est vrai (OUI), le calculateur exécute le programme à partir de la phrase précédée de la 1ère étiquette, sinon à partir de la 2ème étiquette.

LE POLYNOME LOGIQUE

Un polynôme logique est constitué par une ou plusieurs relations logiques séparées par des opérateurs logiques.

Les opérateurs logiques sont :

ET (caractère typographique PB 250 : Ω)

OU (caractère typographique PB 250 : π)

Un opérateur logique sépare deux relations logiques, toute relation logique étant constituée par deux expressions arithmétiques séparées par un opérateur de relation.

Les opérateurs de relation sont :

- supérieur)
- inférieur (
- égal =
- différent &

(les deux premiers correspondent aux signes de parenthèse fermée ou ouverte).

Les expressions arithmétiques figurant dans les relations peuvent être en fixe ou en flottant. Elles peuvent être même de types différents, l'une en fixe et l'autre en flottant mais elles ne peuvent débuter par une constante.

Ex. : SI : A) ␣ 12 ␣ , ——— est interdit

Exception à cette règle : les chiffres entiers de 0 à 9.

Ex. : Si : A = 0 , ——— est autorisé

Si : B = -1 , ——— est autorisé

REMARQUES :

A) Attention :

SI : A = ␣ 0,25 ␣ est interdit

Il faut écrire :

Si : MAX = ␣ 0,25 ␣
Si : A = MAX , ———

B) Les expressions arithmétiques peuvent comporter des fonctions standards ou des procédures si celles-ci ne possèdent qu'un seul paramètre.

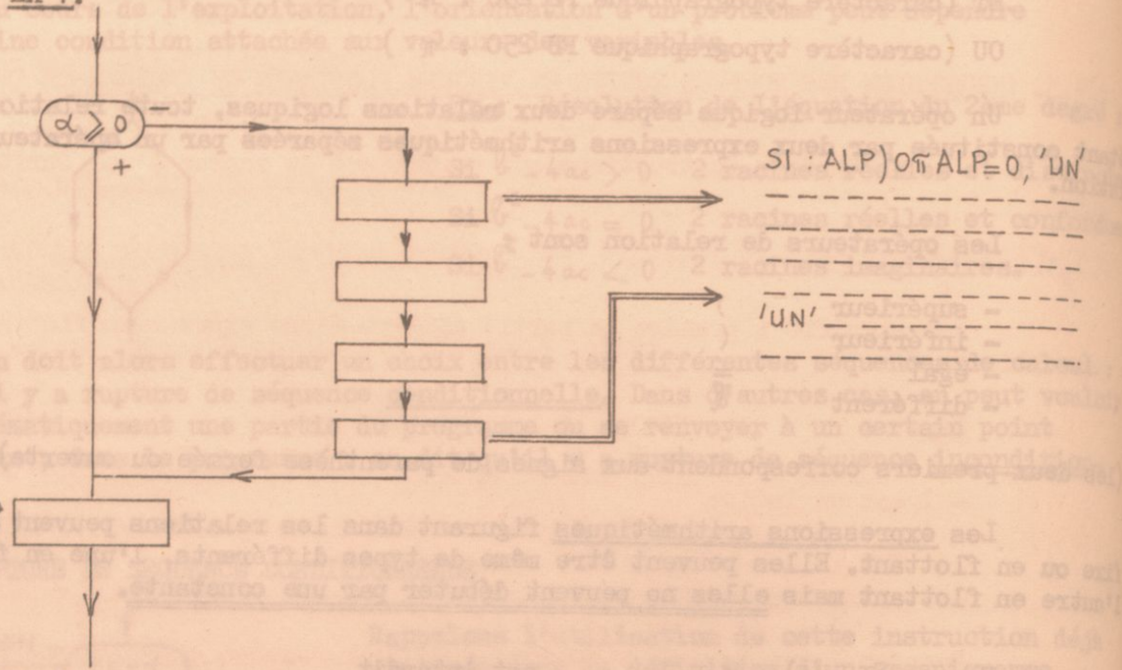
C) Règle de priorité pour les opérateurs logiques :

Une opération logique porte sur le résultat du polynôme logique qui précède et la relation qui suit. Une priorité est donc accordée d'abord aux opérateurs logiques, ensuite aux opérateurs de relation (voir exemple plus loin).

D) Les étiquettes utilisées dans une expression SI : peuvent ne pas avoir encore été utilisées mais elles devront l'être dans le cours du programme sinon l'exploitation ne pourrait se dérouler normalement.

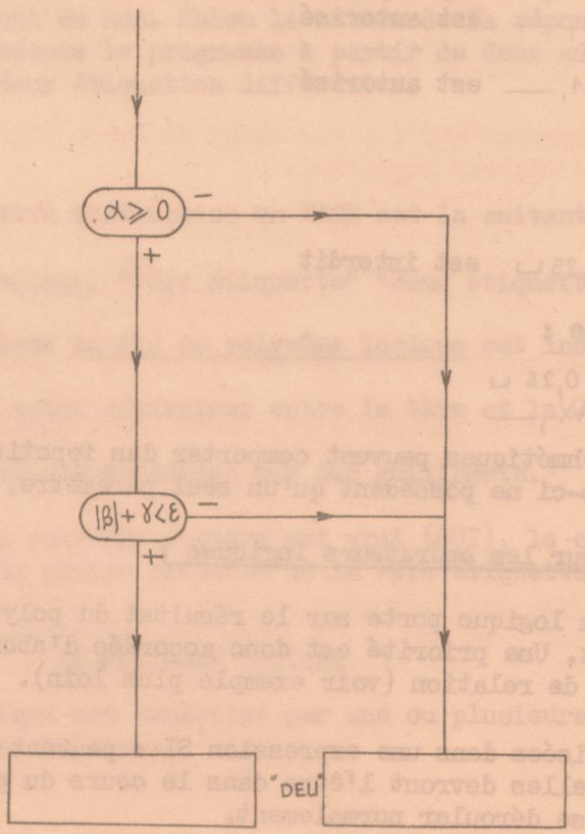
E) La 2ème étiquette peut être omise. Dans ce cas, le calculateur exécute la phrase qui suit l'instruction SI : lorsque la réponse au polynôme logique est non.

EX 1:



EX. 2

Si ; ALP) OT ALP = 0   Ω ABS [BET] + GAM(EPS ,   "UN" "DEU"



II - INSTRUCTION PAIR : Ecriture MAGE : PAI :

Cette instruction permet d'aiguiller le calcul vers une séquence déterminée suivant la parité d'une expression arithmétique <u>en fixe.</u>

PAI : Expression arithmétique en fixe, "1ère étiquette" "2ème étiquette"

Si le résultat de l'expression est pair, le calculateur exécute la phrase repérée par la première étiquette. Si le résultat est impair, il exécute la phrase repérée par la deuxième étiquette ou continue en séquence s'il n'y a qu'une seule étiquette.

III - INSTRUCTION COMMUTATEUR : Ecriture MAGE : COM :

Les éléments périphériques du PB 250 comportent entre autres, un pupitre d'interrogation appelé PI 8 comportant 8 commutateurs numérotés de 10 à 17.

Cette instruction permet de modifier le déroulement d'un programme sur une intervention manuelle du programmeur, extérieure au calculateur.

<u>Ex.</u>: COM : 15, "U N" "D E U"

Si le voyant <u>rouge</u> du commutateur n° 15 est visible, le calculateur effectue la phrase repérée par l'étiquette "UN" sinon, la phrase repérée par l'étiquette <u>"DEU"</u>.

<u>Remarque</u> : Les commutateurs 10 et 17 sont utilisés pour le choix de l'organe d'entrée des données, au cours de l'exploitation.

2,2,4.2. RUPTURE DE SEQUENCE INCONDITIONNELLE

INSTRUCTION VERS   Ecriture en MAGE : VER :

Cette instruction renvoie le programme vers la phrase repérée par l'étiquette figurant dans le corps de l'instruction.

<u>Ex</u> .:   VER : "DEB"

Remarquons que la phrase repérée par l'étiquette peut n'avoir pas encore été écrite.

Exercices :

I-Calculer les 1000 premiers termes de la série suivante :

$$\sum_{n=1}^{1000} \frac{1}{n^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots + \frac{1}{1000^2}$$

```
      : MAX = ⊔1000,⊔
      : S = 0
      : N = 0
`E1'  : N = N+1
      : S = 1/N*2+S
SI    : N < MAX , `E1'
      : (
```

II- Calculer la somme des carrés des 100 premiers nombres entiers.

$$\sum_{n=1}^{100} n^2 = 1 + 2^2 + 3^2 + \dots + 100^2 .$$

Remarque : Les nombres étant des entiers on pourra utiliser des indices :

```
IND   : K, I
      : MAX = ⊔100,⊔
      : K = 0
      : I = 0
`E1'  : I = I+1
      : K = I.I+K
SI    : I > MAX, `SUI'`E1'
`SUI' : (
```

(le signe * n'est pas autorisé en calcul virgule fixe)

III-CALCULER

$\sum_{n=1} 1/n!$ avec $U_n = 1/n!$ tant que Un

```
      : S = 0
      : A = 0
      : UN = 1
`E1'  : A = A+1
      : UN = UN/A
      : S = S+UN
SI    : UN > EPS ↑ UN = EPS , `E1'
      : (
```


2.2.5. INSTRUCTION D'ITERATION

On est amené très souvent dans un problème, à exécuter un certain nombre de fois le même calcul d'une fonction $Y = F(x)$ pour des valeurs de x comprises dans un intervalle $[A, B]$, x variant par pas de DX par exemple.

Les instructions POUR : et RETOUR : (écriture en MAGE POU : RET:) encadrent cette séquence.

```
POU : - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
- - - - - - - - - -
RET :
```

Le corps de la phrase POUR est composé comme suit :

Indice de comptage = valeur initiale, pas, valeur finale

2.2.5.1. L'indice de comptage est un indice (c'est-à-dire une seule lettre). Cet indice peut n'avoir jamais figuré dans le programme mais après l'instruction POU : il sera considéré comme un indice.

2.2.5.2. Par contre, les 3 grandeurs : valeur initiale, pas, valeur finale doivent être des indices connus du calculateur donc :

- soit des constantes sans signe écrites en virgule fixe.

Ex. POU : I = 1 , 1 , 10

- soit des indices déjà déclarés.

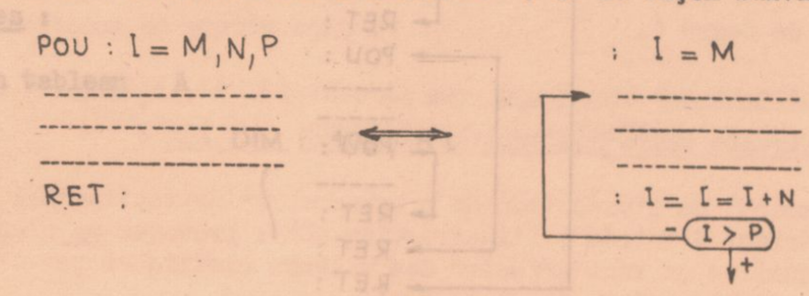
Ex.

```
IND : M, N, P
POU : I = M, N, P
```

2.2.5.3. Ces 3 grandeurs ne peuvent être des expressions arithmétiques et ces indices sont strictement positifs.

Ex. POU : I = 0,1,10 est formellement interdit.

2.2.5.4. Le calculateur exécute l'instruction POU : de la façon suivante :



L'indice de comptage progresse d'un pas puis I est comparé à P : on dit que l'on a fait un test. Si $I \leqslant P$ le calculateur recommence une nouvelle séquence de calcul, depuis son début. Si $I > P$, le calculateur exécute l'instruction suivante.

Par conséquent :

A) l'indice de comptage conserve sa valeur durant tout un passage en séquence et sa valeur ne soit pas être modifiée par une instruction de calcul.

B) L'itération I = M,N,P s'effectuera un nombre de fois égal à la partie entière de

$$\frac{P - M}{N} + 1$$

C) Une itération I = M,N,P est toujours effectuée au moins une fois, même si M est supérieur à P

Ex. : POU : I = 2,1,1 sera effectué une fois

D) Si le pas de progression n'est pas un entier, on ne peut utiliser l'instruction POU: RET:. Il faut écrire en détail la séquence ci-dessus : initialisation-séquence de calcul-progression-test.

E) Si on utilise des variables indicées dans la boucle POU : comme par exemple

```
POU : I = 10,1,19
    : VI =
    ..................
RET :
```

les variables indicées sont appelées après un calcul d'indices. Ce sont donc les VI qui se trouvent de la 10ème à la 19ème mémoire. L'instruction DIM : n, VI devra donc avoir un n égal au moins à 19.

2.2,5.5. L'instruction RET : indique la fin d'une itération débutée par un POU : Le corps de phrase reste vide mais il ne faut pas omettre les deux points.

2.2,5.6. Les instructions POU : RET permettent des boucles imbriquées, chaque instruction RET : renvoyant sur la dernière instruction POU :

Ex.:

```
POU :
-----
-----
POU :
-----
RET :
POU :
-----
-----
POU :
-----
RET :
RET :
RET :
```


A) Le nombre d'instruction POU : doit être le même que celui de RET :

B) Le nombre de boucles imbriquées est limité à 6.

C) L'organisation suivante est <u>interdite</u>.

```
POU :
-----
-----
POU :
-----
RET :
-----
RET
```

Ex : $\sum_{n=1}^{100} n^2$

```
INP : K
    : K = 0
PØU : I = 1,1,100
    : K = I.I + K
RET :
  (
```

Exercice :

Dresser le tableau des 50 premiers nombres entiers, puis leur somme.

```
DIM : 50, T1
PØU : I = 1,1,50
FLØ : IF = I
    : TI = IF
RET :
    : SØM = 0
PØU : I = 1,1,50
    : SØM = SØM + TI
RET :
```

Exercices :

1 On considère un tableau A

```
DIM : 100, A1
  (
```

Combien de nombres du tableau sont $>$ B ?

```
      : K = 0
  PØU : I = 1,1,100
   SI : AI > B,`E1' `E2'
 `E1' : K = K+1
 `E2' RET :
```

Dans un tableau A (100), trouver l'indice et la valeur du plus grand élément.

```
DIM : 100, AI          : I = 1
                       : MAX = A1
                   POU : I = 2,1,100
                    SI : AI - PCO, `RET'
                       : P = AI
                       : J = I
                 `RET' RET :
```

P contient l'élément et J l'indice.

2.2,6. INSTRUCTIONS D'ENTREE - SORTIE

2.2,6.1. ENTREE

DEMANDER ET LIRE

Reprenons l'exemple du polynôme de Horner : le programme décrit une suite d'opérations effectuées sur des valeurs littérales $A_0, A_1, \ldots,$ X . Le calculateur effectuera, par exemple, le produit de la valeur numérique de A_1 par la valeur numérique de X . Ou bien ces valeurs sont attribuées par le programme aux variables, auquel cas le problème n'est utilisable que pour une seule série de valeurs numériques ou bien au moment de l'exploitation du programme, on attribue une valeur numérique à chacune des variables par un organe d'entrée.

Dans ce dernier cas, le programme est utilisable pratiquement indéfiniment et c'est donc cette solution qui doit être adoptée. On utilise pour cela les instructions DEM :et LIR : dont l'effet est le suivant : lorsqu'on exploite le programme le calculateur s'arrête sur ces instructions et se met en attente sur un organe extérieur. L'opérateur introduit les valeurs qu'il désire utiliser immédiatement et le calculateur reprend de suite des instructions du programme.

Ex.: Polynôme de degré 4.

LIR : A0, A1, A2, A3, A4, X

Différence entre les instructions DEM : et LIR :

Au moment de l'exécution du programme, l'instruction LIR : met en attente un organe extérieur d'entrée ; l'instruction DEM : provoque en plus, le rappel des variables auxquelles on veut affecter des valeurs numériques par l'impression du nom de cette variable suivie du signe égal.

Ex.: A0 =

Format d'introduction des valeurs numériques :

A) Indices

signe - valeur - caractère terminal

Le caractère terminal peut être : -un espace
-une tabulation
-un retour chariot

Ex.: - 13 ⊔

B) Variables simples ou indicées :

1) signe - valeur suivie d'un point ou d'une virgule - caractère terminal

Ex.: 13,4 ⊔
-50. RC
-0.25 TAB

2) signe - mantisse - E - signe - exposant - caractère terminal

Ex.: Diverses manières d'introduire la valeur 1

$1 = 1.10^0 \Longrightarrow$ + 1 E0 ⊔
ou 1,E0 ⊔
ou 1, ⊔
$1 = 0,1.10^1 \Longrightarrow$ 0,1 E1 ⊔
$1 = 10.10^{-1} \Longrightarrow$ 10. E-1 ⊔

NOTA : a) L'absence de signe est interprétée par la machine comme le signe +
b) Le point (ou la virgule) décimal n'est pas obligatoire. S'il n'y en a pas, la machine considère qu'il est à droite du dernier chiffre de la mantisse. (La mantisse est considérée comme un nombre entier).
c) L'exposant est un entier sans point (ni virgule) décimal.
d) Les espaces et les zéros précédant le nombre et l'exposant sont éliminés tant qu'un chiffre (ou un signe) n'a pas été lu.

2.2,6.2. SORTIE

Les instructions de sortie sont :

IMPRIMER [IMP]: sortie sur imprimante (machine à écrire).
[PER]: sortie sur ruban perforé.

L'instruction IMP : permet d'imprimer :

- des valeurs numériques,
- des textes quelconques, y compris les signes de ponctuations.

A) Impression des valeurs numériques :

1) d'un indice N

Instruction	Format de sortie
IMP : N	**** (4 chiffres)

2) d'une variable simple ou indicée VAR

Instruction	Format de sortie
- IMP : VAR	$\underbrace{**********}_{10\ \text{chiffres}}$ E **
Ex.: 23,57 0,2357.10^{-2}	2357000000 E 2
- IMP :(M , N) VAR	$\underbrace{****.**}$ M chiffres avant la virgule et N chiffres après
EX.: (2,3) VAR	23,570
- IMP : (P) VAR	$\underbrace{*******}_{P\ \text{chiffres}}$ E **

L'identificateur de variable est toujours suivi d'un des caractères de ponctuation suivant, ce caractère étant effectué à l'impression.

Les caractères de ponctuation utilisés sont :

␣ espace
RC retour chariot
TAB tabulation

REMARQUES :

1) On doit avoir $M \neq 0$
$N \neq 0$
$M + N \leqslant 10$

2) Si le format de sortie est incorrect (M trop petit), le nombre est sorti automatiquement sous forme mantisse et exposant.

3) On peut de plus imprimer le nom de la variable ou de l'indice.
Pour cela, on écrit le nom de la variable à faire imprimer en minuscules, suivi du caractère de ponctuation espace :

Ex.: IMP : var ␣ (M , N) VAR RC ⟹ VAR = ␣ ****, *** RC
IMP : (2,1) X ␣ ⟹ ××, × ␣
IMP : (1,2) VA1 TAB (2,2) VA2 ␣ (5,5) VA3 ␣ ⟹ ×, ×× TAB ××, ×× ␣ ×××××, ××××× ␣

B) Impression d'un texte :

IMP : [..........]

Le texte entre crochets sera imprimé en majuscules, même s'il a été écrit en minuscules. Si on désire donc une impression en minuscules, il convient de faire précéder le texte d'une "corbeille haute". D'autre part, le nombre de caractères y compris les corbeilles doit être inférieur à 30.

IMP : [.....30 CAR ....] [.......30 CAR...]

C) Changement de ligne :

Le caractère retour chariot sert de code "fin de phrase" pour le compilateur MAGE.

Le retour chariot avec changement de ligne n'est effectif à l'impression que dans les cas suivants :

1) Impression d'une variable

Ex.: IMP : (5,5) VAR RC

2) Impression d'un texte ou d'un caractère de ponctuation suivi d'un .

Ex.: IMP :[.......] . RC
IMP : .... TAB .

3) Instruction : IMP : . RC

Le retour chariot n'est pas effectué dans les exemples suivants :

1 - IMP : [---------] RC
2 - IMP : ---------- TAB RC
3 - IMP : ----- - VAR . (à l'impression, la machine imprime la valeur de VAR suivie d'un point)
4 - IMP : VAR ␣ RC

2.2,6.3. INSTRUCTION COPIE [COP :]

Uniquement sur les calculateurs de plus de 4000 mots.

Cette instruction permet de recopier une ligne de texte (Date de l'exploitation ou indicatif quelconque) introduite par un lecteur de ruban perforé au moment de l'exploitation.

<u>EX.</u>: COP : Impression d'un texte.
COP : P Perforation du texte.

2.2,7. <u>INSTRUCTIONS DIVERSES</u>

2.2,7.1. FIN : RC

L'instruction FIN : RC sans oublier les deux points et le retour chariot, doit toujours être la dernière d'un programme MAGE. Dans la phase compilation, elle provoque l'arrêt du compilateur. Dans la phase exécution, elle arrête le déroulement du programme sauf si elle est précédée d'une instruction VER : "DEB" par exemple.

On peut toujours mettre dans le corps de la phrase un chiffre de 1 à 7 qui sera visualisé sur le pupitre du calculateur.

2.2,7.2. HALTE : RC

L'instruction HAL : suivie éventuellement d'un chiffre de 1 à 7 joue le même rôle que FIN : en phase exécution mais n'arrête pas la compilation. Il peut y avoir plusieurs HAL : dans un programme mais <u>ce n'est jamais</u> la dernière instruction.

HAL : ⟶ équivaut à : VER:'FIN'

'FIN'FIN

2.2,7.3. PAUSE

Cette instruction permet de suspendre provisoirement le déroulement du programme, pour permettre une vérification des données, par exemple. La reprise du traitement est commandée par l'opérateur, au moyen d'une manipulation sur le pupitre du calculateur.

On peut si on le désire, indiquer un chiffre de 1 à 7 dans le corps de cette instruction ; ce chiffre sera affiché au pupitre pendant toute la durée de la pause.

Dans tous les cas, les deux points doivent être écrits.

<u>Ex.</u>: PAU : 3
PAU :

<u>Nota</u>: Lorsque le corps de phrase des instructions FIN : HAL : et PAU : est vide, le chiffre 6 est visualisé.

2.2,7.4. TEXTE

Cette instruction sert à placer un commentaire dans l'écirture du programme MAGE. Le tout sera éliminé à la compilation et n'apparaîtra donc pas à l'exploitation du programme. Il faut se servir de l'ordre IMP : si on désire faire imprimer un commentaire au cours de l'exécution.

<u>Ex.</u>:

<u>Programme MAGE</u>	<u>Exploitation</u>
IMP : [Problème 1]	Problème 1
TEX : Equation du 1er degré	A=―――――
DEM : A	

2.2,7.5. RENVOI

Cette instruction permet de faire appel à un sous-programme écrit en langage machine. Le corps de cette instruction est constitué par l'adresse machine de la mémoire où débute le sous-programme : un numéro de 5 chiffres en octal, les trois premiers désignant le numéro de secteur de la première instruction du sous-programme et les deux derniers indiquant le numéro de ligne ($\leqslant 17$) de cette instruction. A la fin de ce sous-programme, le calculateur exécute la phrase qui suit immédiatement l'instruction RENVOI :

<u>Ex.</u>: REN : 00610

2.2,7.6. AFFECTATION

Cette déclaration est équivalente à IND : mais porte sur des variables simples et non indicées.

Elle est utile lorsqu'on désire transmettre des informations entre deux programmes MAGE compilés indépendamment.

Son utilisation, jointe à celle des déclarations DIM : IND : permet de désigner les mêmes varaibles par les mêmes noms dans plusieurs programmes MAGE indépendants, tout en assurant la transmission des valeurs correspondantes.

Il suffit pour cela de grouper en tête de chaque programme toutes les déclarations AFFECTATION, DIMENSION, INDICE. Cette partie déclaration ainsi constituée doit être répétée identiquement, dans le même ordre, en tête de chaque programme. Le corps de la déclaration AFF : est constitué par la liste des noms des variables simples, séparés par des virgules.

Ex.: AFF : ALP, BET, GAM

La partie déclaration permet aussi d'assurer l'échange de valeurs entre un programme MAGE et un programme langage machine.

2.2,8. LES PROCEDURES

L'existence des "procédures" en programmation symbolique a plusieurs justifications :

I - Rôle de "fonction"

Au cours du déroulement d'un problème, une même expression arithmétique peut se présenter plusieurs fois, avec des opérandes différents. La procédure, joue alors le rôle d'une "fonction" écrite sur mesure.

II - Fractionnement d'un problème

Lors de la résolution d'un problème assez long, il peut être extrêmement agréable de fractionner l'étude en plusieurs parties. Chacune de ces parties formant une procédure, le problème est évidemment plus simple à étudier et mieux encore, il peut ainsi exister une bibliothèque de programmes (Communications de l'A.C.M. ou Revue française du Traitement de l'Information, par exemple).

L'utilisation le plus classique est celle où la procédure joue le rôle d'une fonction. Prenons un exemple très simple pour en étudier le fonctionnement : supposons que l'on soit amené à calculer plusieurs fois le polynôme du 2ème degré.

```
:Y1 = A1.X1 + B1.X1 + C1
:Y2 = A2.X2 + B2.X1 + C2
------------------------
------------------------
------------------------
:YN = AN.XN + BN.XN + CN
```

Pour éviter la répétition de ce calcul qui se réduit ici à une ligne mais qui pourrait être beaucoup plus long, on écrit en début de programme une procédure effectuant le calcul

: Y = A.X + B.X + C

Les paramètres utilisés n'ont pour l'instant, aucune valeur numérique, ils servent à décrire la "forme" du calcul; ce sont des paramètres formels.

Au moment ou nous auront effectivement à attribuer des valeurs numériques aux paramètres formels, nous "appelerons" la procédure en lui joignant des paramètres effectifs ayant une valeur numérique. La procédure sera donc définie par son nom et par un certain nombre de paramètres.

Etudions d'une façon plus précise la procédure MAGE au point de vue

- écriture
- utilisation

2.2,8.1. ECRITURE D'UNE PROCEDURE

En MAGE, une procédure débute par l'instruction PROCEDURE : et se termine par l'instruction REPRISE :

Elle est définie par son nom et éventuellement par ses paramètres d'où la phrase :

PRO : NOM [P1, P2, ...]

a) le nom est un ensemble alphanumérique dont les 3 premiers caractères seuls sont pris en compte par le calculateur. Il débute obligatoirement par une lettre.

La seule restriction au sujet de ce nom est qu'il doit être différent des noms de fonctions standards (SIN, COS, LNE, ...) existant en MAGE.

b) les paramètres s'ils existent, sont des paramètres formels au nombre maximum de 8. Ils sont écrits entre crochets, séparés par une virgule, à la suite du nom de la procédure.

Les paramètres formels ne peuvent être que des variables simples. Il faut cependant remarquer que :

1) Il n'est pas obligatoire de déclarer en tant que paramètres toutes les variables utilisées dans la procédure.

2) Ces variables ne sont pas locales à la procédure mais ont une portée qui s'étend sur tout le programme. Donc, si par inadvertance, deux grandeurs différentes portent le même nom et sont utilisées l'une dans la procédure, l'autre dans la suite du programme, le calculateur les considérera comme une seule et même valeur. Les risques d'erreur sont très importants. Il est donc vivement conseillé d'utiliser des noms de variables différents dans la procédure et dans le programme principal.

3) Ces variables, comme toutes les variables, doivent être connues du calculateur, c'est-à-dire qu'elles ne peuvent apparaître à droite du signe = sans avoir été déclarées ou utilisées à gauche d'un signe = auparavant.

<u>Ex.</u>:

```
PRO : POL[X]
:Y= A.X+B.X+C
REP :
```

Dans cette procédure, A, B et C sont des inconnues pour le calculateur et une erreur sera donc détectée par le compilateur à cet endroit (opérande inconnu : OIE)

Il convient d'écrire :

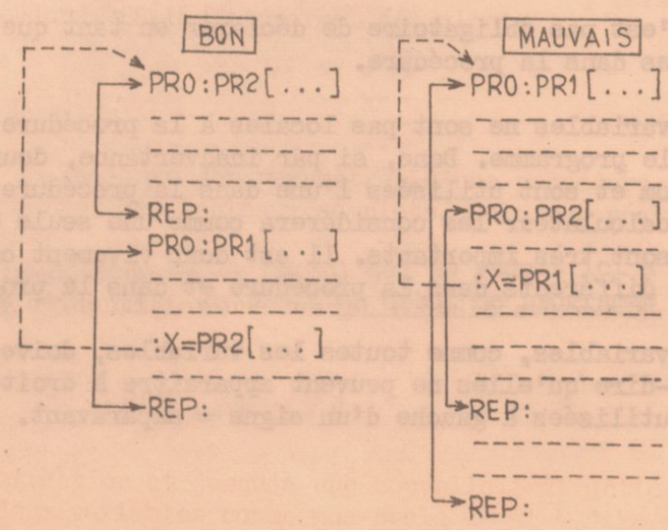
```
PRO : POL[A,B,C,X]
:Y= A.X+B.X+C
REP :
```

c) Une procédure est un sous-programme qui ne sera exécuté que par un "appel" précis et non pas en séquence. Par conséquent, la description de la procédure devra être obligatoirement sautée par une instruction VER :

<u>Ex.</u>:

```
VER : 'E1`
PRO : BIZ[A,B,C]
---------------
---------------
---------------
REP :
'E1`----------- ==> Début de programme.
---------------

VER : 'DEB`
PRO : RES [A,B]
-------------
REP :
PRO : ARS [X]
-------------
REP :
`DEB'---------
-------------
```

d) Dans une procédure, on peut faire appel à une procédure <u>déjà écrite</u> mais on ne peut pas imbriquer plusieurs procédures.



e) Les procédures s'écrivent en début de programme. En effet, le corps d'une procédure ne doit pas être au-delà de la ligne 17 du calculateur (Impression de EEC par le compilateur).

2.2,8.2. UTILISATION D'UNE PROCEDURE

Il y a deux moyens d'appeler une procédure précédemment écrite

- par l'instruction EXECUTER
- comme une fonction à l'intérieur d'une expression arithmétique.

Dans les 2 cas, on indique le nom de la procédure, suivi de la liste des paramètres effectifs, si la déclaration de procédure correspondante comporte une liste de paramètres formels.

La liste des paramètres effectifs à la même syntaxe que la liste des paramètres formels. Elle doit comporter autant de noms que la liste correspondante des paramètres formels.

<u>Ex.</u>: - déclaration de procédure : PRO : POL[A,B,C,X]

- utilisation de la procédure : POL [A1 , B1 , C1 , X1]

Lorsqu'on utilise une procédure, les valeurs des paramètres effectifs sont affectées, avant exécution de la procédure, aux paramètres formels correspondants dans l'ordre. Ainsi, dans l'exemple choisi, avant exécution de la procédure POL, A prend la valeur de A1 , B celle de B1 , etc... Contrairement aux paramètres formels, dans la liste des paramètres effectifs, les deux premiers paramètres et les deux premiers seulement peuvent être indicés. Il doit leur correspondre, dans la liste des paramètres formels, deux variables simples. Seul l'élément du tableau désigné par la valeur de l'indice au moment de l'appel de la procédure est transmis à la procédure et non le tableau complet. De plus, les paramètres effectifs sont toujours écrits en convention <u>virgule flottante</u>.

Les deux moyens d'appeler une procédure dépendent de l'utilisation de celle-ci.

A) <u>La procédure calcule un seul résultat</u>

<u>Ex.</u>:
$$F(x) = Y = ax^2 + bx + c$$

Dans ce cas, l'instruction REP: qui termine la description de la procédure doit rappeler le <u>nom</u> du résultat.

```
VER:'E1'
PRO:POL[A,B,...]
--------
--------
--------
:Y=.....
REP:Y
'E1`------
```

La procédure sera alors utilisée comme une fonction c'est-à-dire soit à l'intérieur d'une expression arithmétique soit dans un test de la façon suivante :

Ex.: :Z=M.POL[A1,B1,C1,X1]+N

ou SI:POL[A1,B1,C1,X1](0,'UN''DEU'

Si la procédure ne possède pas de paramètres, ce qui est assez exceptionnel, le nom de la procédure sera suivi d'une virgule.

B) La procédure comporte plusieurs résultats qui doivent être utilisés dans la suite du programme principal :

Ex.:

```
PRO:POL[A,B,C,X]
   :Y=A.X+B.X+C          y=ax²+bx+c
   :YP= 2.A.X+B          y'=2ax+b
   :YS=2.A               y"=2a
REP:
```

Dans ce cas, le corps de la phrase de l'instruction REP: est vide et on appelle la procédure par l'intermédiaire de l'instruction EXECUTER :

Ex. : EXE : POL[A1,B1,C1,X1]

Cette instruction a pour effet de rendre disponibles les variables utilisées dans la procédure et ce jusqu'à une nouvelle instruction EXE :

Ex. :

```
EXE : POL[A1,B1,C1,X1]
    : Z=ALP.Y
    : T=BET.YP
    : U=GAM.YS+Z+T
EXE : POL[A2,B2,C2,X2]
```

Procédure à écrire :

Etant donnés deux nombres N et D, soit Q et R tels que : $N = Q \cdot D + R$

Ecrire une procédure RES[N,D] permettant d'obtenir R

2.2.8.3. DECLARATION POSER

L'organisation de la liste des paramètres d'une procédure ne permet pas de transmettre comme paramètres des tableaux complets. Cela est réalisé au moyen de de la déclaration POSER :

Celle-ci permet d'identifier une variable indicée à une autre variable indicée.

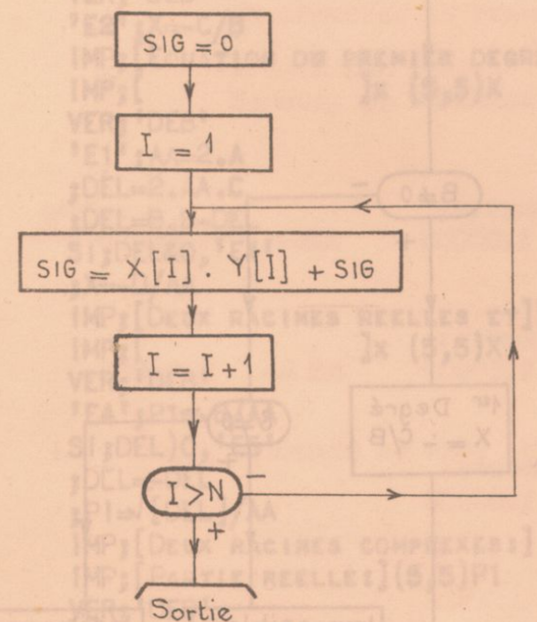
Ex.: POS : XI = VJ

Après cette instruction et jusqu'à une nouvelle instruction POSER : le calcultateur traite ces deux variables comme une seule c'est-à-dire que pour la même valeur de I et de J les valeurs numériques attribuées à XI et à VJ sont égales.

Autrement dit, XI paramètre formel sert à décrire la procédure mais ne possède pas de réalité numérique. Avant l'appel de la procédure, il faut attribuer à XI non pas une seule valeur mais un ensemble de valeurs, l'ensemble des valeurs de VJ, tableau déclaré dans le programme principal.

Il va de soi que les composantes de XI n'ayant pas d'existence réelle, il est inutile de leur réserver des mémoires et l'ordre DIM : aura une dimension nulle. Un exemple simple illustrera mieux cet exposé :

Soit à effectuer le produit scalaire de deux vecteurs U_A et V_B



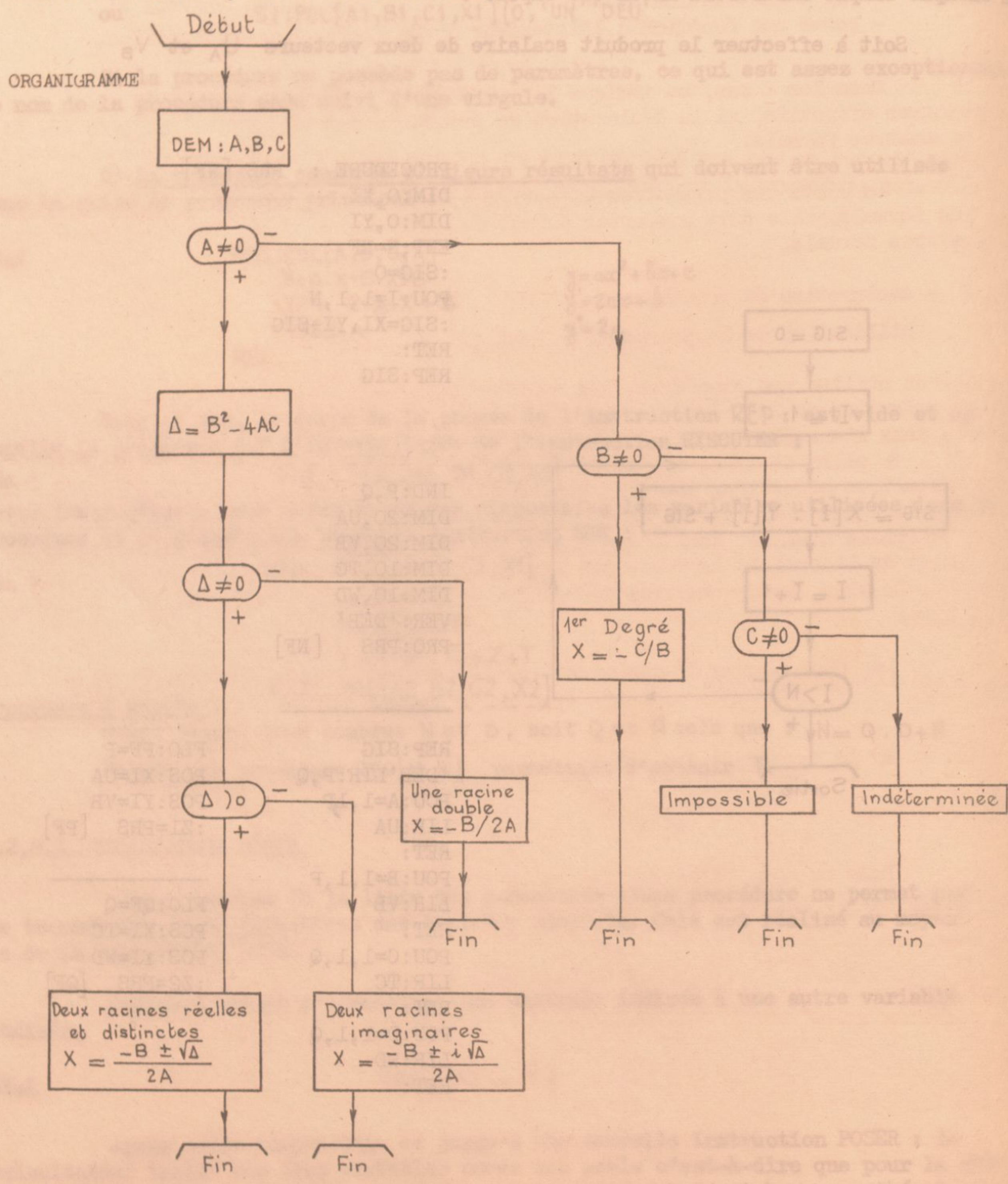

```
PROCEDURE :  PRS [NF]
DIM:0,XI
DIM:0,YI
ENT;N=NF
:SIG=0
POU:I=1,1,N
:SIG=XI.YI+SIG
RET:
REP:SIG


IND:P,Q
DIM:20,UA
DIM:20,VB
DIM:10,TC
DIM:10,WD
VER:'DEB'
PRO:PRS   [NF]
______________

REP:SIG
'DEB'LIR:P,Q
POU:A=1,1,P
LIR:UA
RET:
POU:B=1,1,P
LIR:VB
RET:
POU:C=1,1,Q
LIR:TC
RET:
POU:D=1,1,Q
LIR:WD
RET:

FLO:PF=P
POS:XI=UA
POS:YI=VB
:Z1=PRS  [PF]
______________
FLO:QF=Q
POS:XI=TC
POS:YI=WD
:Z2=PRS  [QF]
```

Chapitre III

EXEMPLES DE PROBLEMES

RESOLUTION DE L'EQUATION DU SECOND DEGRE :

ORGANIGRAMME



PROGRAMME

```
,1340
TEX;RESOLUTION DE L'EQUATION DU SECOND DEGRE
'DEB'IMP;.
DEM;A,B,C
IMP;.
SI;A&O,'E1'
SI;B&O,'E2'
SI;C&O,'E3'
IMP;[INDETERMINATION].
VER;'DEB'
'E3'IMP;[IMPOSSIBILITE].
VER;'DEB'
'E2';X=-C/B
IMP;[EQUATION DU PREMIER DEGRE:].
IMP;[                  ]x (5,5)X
VER;'DEB'
'E1';AA=2.A
;DEL=2.AA.C
;DEL=B.B-DEL
SI;DEL&O,'E4'
;X=-B/AA
IMP;[DEUX RACINES REELLES ET][ CONFONDUES:].
IMP;[                  ]x (5,5)X
VER;'DEB'
'E4';P1=-B/AA
SI;DEL)O,'E5'
;DEL=-DEL
;PI=√[DEL]/AA
IMP;[DEUX RACINES COMPLEXES:].
IMP;[PARTIE REELLE:](5,5)P1                [PARTIE IMAGINAIRE:](5,5)PI
VER;'DEB'
'E5';P2=√[DEL]/AA
;X1=P1+P2
;X2=P1-P2
IMP;[DEUX RACINES REELLES ET DISTINCTES:].
IMP;[                  ]x1 (5,5)X1            x2 (5,5)X2
VER;'DEB'
FIN;
```

EXPLOITATION

A=0, B=0, C=0,
INDETERMINATION

A=0, B=0, C=9,
IMPOSSIBILITE

A=0, B=2, C=1,
EQUATION DU PREMIER DEGRE:
X = - .50000

A=1, B=0, C=-1,
DEUX RACINES REELLES ET DISTINCTES:
X1 = 1.00000 X2= - 1.00000

A=1, B=2, C=-3
DEUX RACINES REELLES ET DISTINCTES:
X1 = 1.00000 X2 = - 3.00000

A=1, B=2, C=1,
DEUX RACINES REELLES ET CONFONDUES:
X = - 1.00000

A=2, B=1, C=3,
DEUX RACINES COMPLEXES:
PARTIE REELLE:- .25000 PARTIE IMAGINAIRE: 1.19895

PROBLEME 2.

CLASSEMENT D'UNE SUITE DE NOMBRES QUELCONQUES DANS L'ORDRE CROISSANT

METHODE DE SHELL

ETANT DONNE UNE SUITE DE NOMBRES $A_1, A_2, A_3, \ldots, A_N$, ON UTILISE LE PRINCIPE SUIVANT POUR ORDONNER CETTE SUITE:

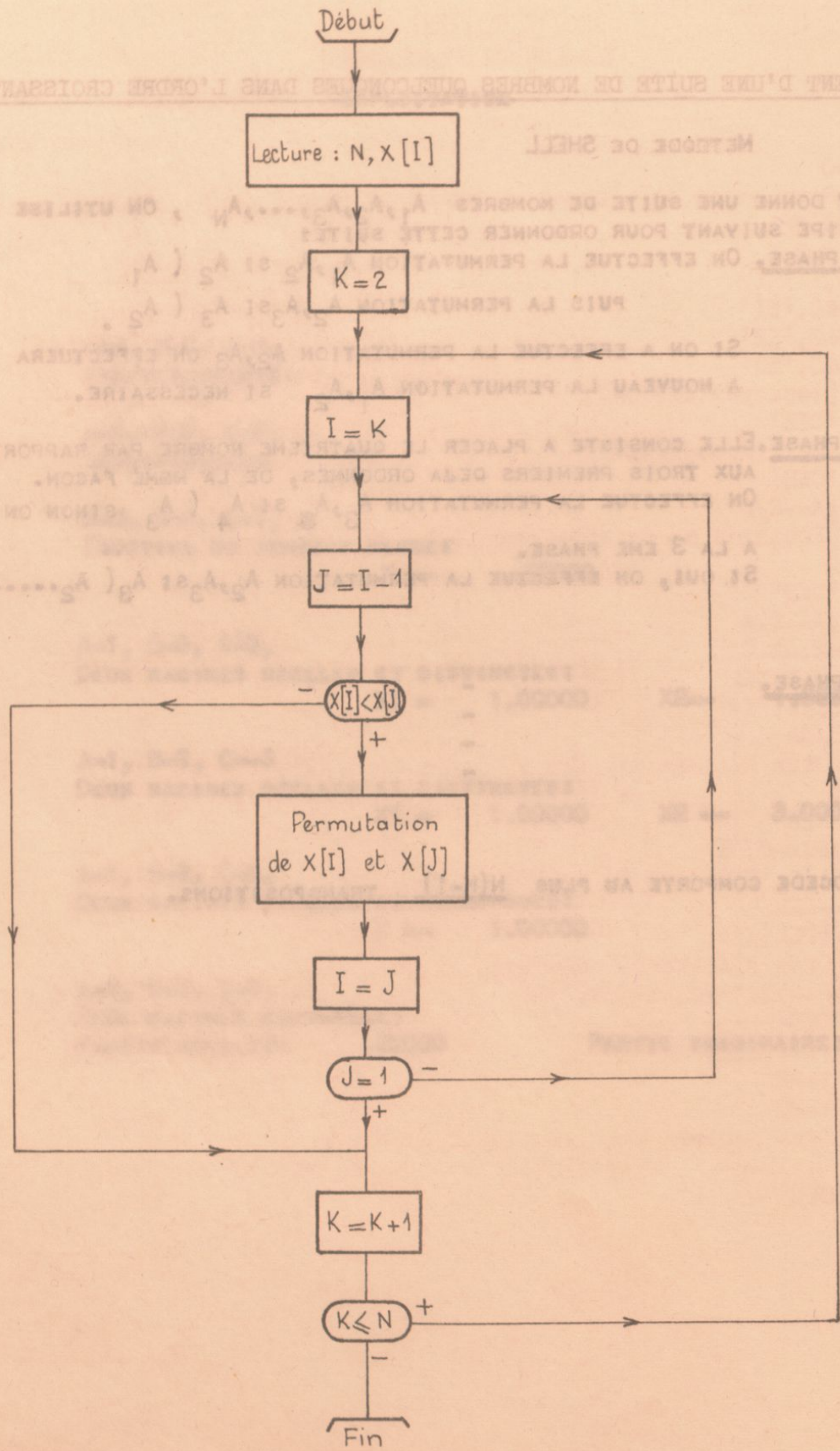
1ERE PHASE. ON EFFECTUE LA PERMUTATION A_1, A_2 SI $A_2 < A_1$
PUIS LA PERMUTATION A_2, A_3 SI $A_3 < A_2$.
SI ON A EFFECTUE LA PERMUTATION A_2, A_3 ON EFFECTUERA A NOUVEAU LA PERMUTATION A_1, A_2 SI NECESSAIRE.

2EME PHASE. ELLE CONSISTE A PLACER LE QUATRIEME NOMBRE PAR RAPPORT AUX TROIS PREMIERS DEJA ORDONNES, DE LA MEME FACON.
ON EFFECTUE LA PERMUTATION A_3, A_4 SI $A_4 < A_3$ SINON ON PASSE A LA 3 EME PHASE.
SI OUI, ON EFFECTUE LA PERMUTATION A_2, A_3 SI $A_3 < A_2$........

3ème PHASE. -
-
-
-

LE PROCEDE COMPORTE AU PLUS N(N-1) TRANSPOSITIONS.

ORGANIGRAMME



PROGRAMME

```
,1340
IMP;.
DIM;1001,XI,XJ
IND;N
'DEB'PAU;1
IMP;[NOMBRE DE VALEURS EN FIXE : ]
LIR;N
PAU;2
POU;I=1,1,N
LIR;XI
RET;
PAU;3
POU;K=2,1,N
;I=K
'E1';J=I-1
SI;XI(XJ,'E3''E2'
'E3';S=XI
;XI=XJ
;XJ=S
;I=J
SI;J&1,'E1'
'E2'RET;
PAU;4
;T=1
POU;I=1,1,N
COM;16,'ROU'
PER;(7,2)XI
'ROU'SI;T=5,'E4'
IMP;(7,2)XI
;T=T+1
VER;'E5'
'E4'IMP;(7,2)XI
;T=1
'E5'RET;
VER;'DEB'
FIN;
```

EXPLOITATION

Les nombres donnés sont :

6, 6, 12, 13, 10, 9, 10, 17, 16, 5, 9, 12, 12, 13, 12, 12, 9, 11, 7, 16, 10, 16, 9, 11, 11, 13, 13, 11, 8, 13, 17, 7, 6, 14, 15, 16, 17, 10, 14, 13, 14, 13, 10, 17, 16, 11, 11, 16, 12, 12, 10, 13,

---

Nombre de valeurs en fixe : 52

5.00	6.00	6.00	6.00	7.00
7.00	8.00	9.00	9.00	9.00
9.00	10.00	10.00	10.00	10.00
10.00	10.00	11.00	11.00	11.00
11.00	11.00	11.00	12.00	12.00
12.00	12.00	12.00	12.00	12.00
13.00	13.00	13.00	13.00	13.00
13.00	13.00	13.00	14.00	14.00
14.00	15.00	16.00	16.00	16.00
16.00	16.00	16.00	17.00	17.00
17.00	17.00			

Durée du classement : 3MN

ME 3

CLASSEMENT DANS L'ORDRE CROISSANT D'UNE SUITE DE NOMBRES ENTIERS

* * *

PRINCIPE: ON SE DONNE UN TABLEAU XI DE DIMENSION 100, CHACUN DE SES ELEMENTS ETANT VIDE (POIDS NUL: XI=0).
QUANT ON LIT UN NOMBRE VAL, ON AUGMENTE D'UNE UNITE LE POIDS DE L'ELEMENT XI DONT L'INDICE I EST EGAL A VAL. TOUS LES NOMBRES AYANT ETE LUS, IL SUFFIT DE RELIRE LE TABLEAU DES XI EN IMPRIMANT LA VALEUR DE I UN NOMBRE DE FOIS EGAL AU POIDS DU XI CORRESPONDANT. POUR LES I POUR LESQUELS CE POIDS EST NUL, ON N'IMPRIME RIEN.

PROGRAMME:

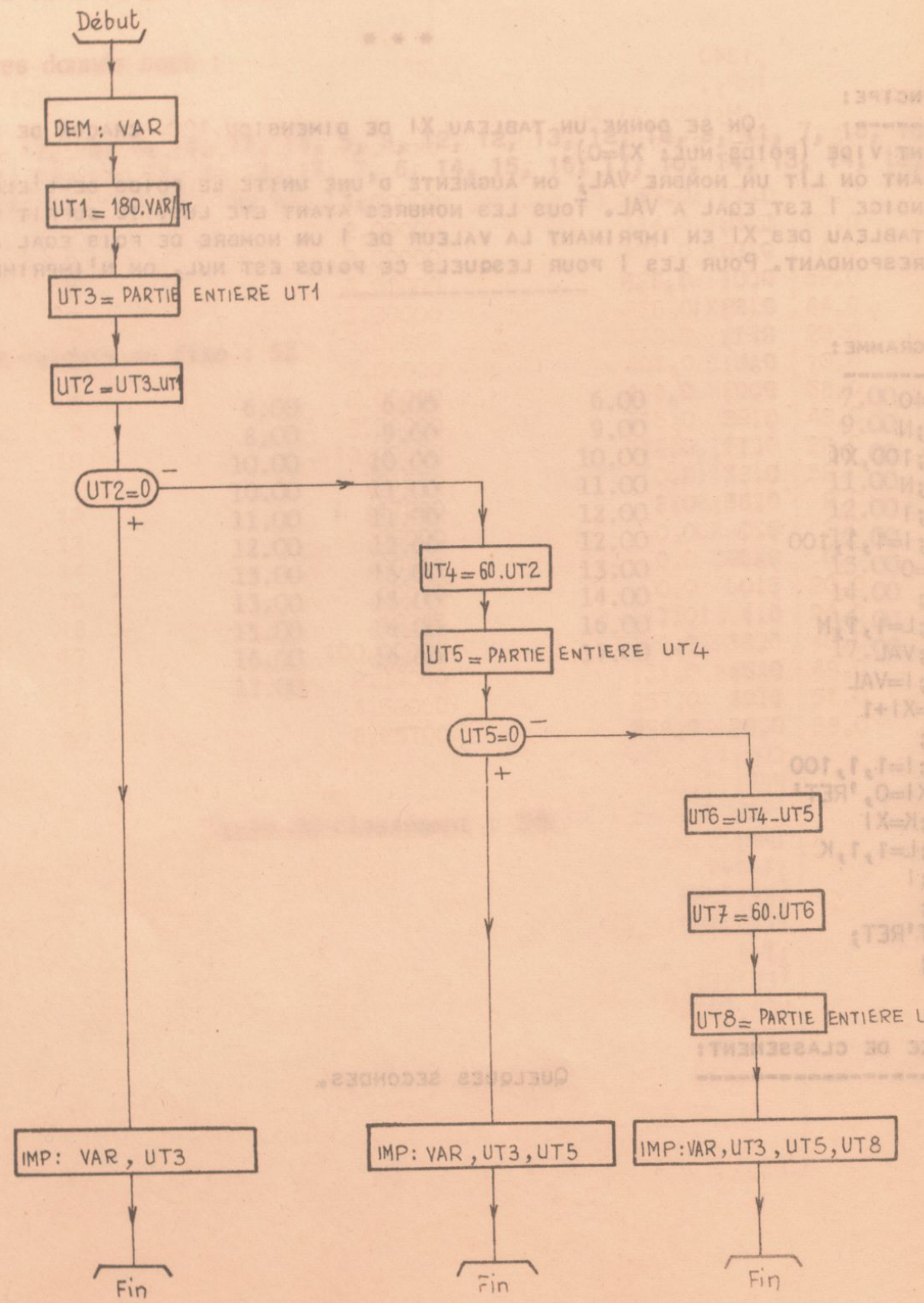
```
,1340
IND;N
DIM;100,XI
DEM;N
PAU;1
POU;I=1,1,100
;XI=0
RET;
POU;L=1,1,N
LIR;VAL
ENT;I=VAL
;XI=XI+1
RET;
POU;I=1,1,100
SI;XI=0,'RET'
ENT;K=XI
POU;L=1,1,K
IMP;I
RET;
'RET'RET;
FIN;
```

DUREE DE CLASSEMENT: QUELQUES SECONDES.

PROBLEME 4

Ecriture d'un programme avec procédure pour transformer des radians en degrés, minutes, secondes

ORGANIGRAMME



PROGRAMME

```
,1340
IND;N
;N=1
;PI= 3,14159265
IMP;[
   N                    RADIANS][              DEGRES   MN   SE
].
VER;'MO1'
PRO;CHE[AL1]
ENT;W=AL1
FLO;AL2=W
REP;AL2
PRO;MIN[VAR]
;UT1= 180, .VAR/PI
;UT3=CHE[UT1]
;UT2=UT1-UT3
;UT5=0
;UT8=0
SI;UT2=0,'MO2'
;UT4= 60, .UT2
;UT5=CHE[UT4]
SI;UT5=UT4,'MO2'
;UT6=UT4-UT5
;UT7= 60, .UT5
;UT8=CHE[UT7]
'MO2'REP;
'MO1'LIR;VAL
EXE;MIN[VAL]
IMP;N              (3,5)VAL              (5,1)UT3 (2,1)UT5 (2,1)UT8
;N=N+1
VER;'MO1'
FIN;
```

EXPLOITATION

13$,

N	RADIANS	DEGRES	MN	SE
1	1.00000	57.0	17.0	44.0
2	2.00000	114.0	35.0	29.0
3	3.00000	171.0	53.0	14.0
4	4.00000	229.0	10.0	59.0
5	5.00000	286.0	28.0	44.0
6	6.00000	343.0	46.0	28.0
7	7.00000	401.0	4.0	13.0
8	8.00000	458.0	21.0	58.0
9	9.00000	515.0	39.0	43.0
10	10.00000	572.0	57.0	28.0
11	.01570	0.0	53.0	58.0
12	.17280	9.0	54.0	2.0
13	.00000	0.0	0.0	1.0
14	.00611	0.0	21.0	0.0
15	.00010	0.0	0.0	20.0
16	.12345	7.0	4.0	24.0
17	100.00000	5729.0	34.0	40.0
18	2.30000	131.0	46.0	49.0
19	4.50000	257.0	49.0	51.0
20	6.25700	358.0	29.0	58.0

PROBLEME . 5

PRODUIT D'UNE MATRICE A DE DIMENSION M.P PAR UNE MATRICE B DE DIMENSION P.N

-.-.-.-.-

PROGRAMME

LA MATRICE PRODUIT C DE DIMENSION M.N A POUR ELEMENTS:

$$C_{ij} = \sum_{k=1}^{p} a_{ik}\, b_{kj}.$$

```
,1340
IMP;.
DIM;10.10,AIK
DIM;10.10,BKJ
DIM;10.10,CIJ
IND;M,N,P
'E1'IMP;[DIMENSION DES MATRICES: ][A(M.P) ET B(P.N) EN FIXE.].
DEM;M,N,P
PAU;1
POU;I=1,1,M
POU;K=1,1,P
LIR;AIK
RET;
RET;
POU;K=1,1,P
POU;J=1,1,N
LIR;BKJ
RET;
RET;
IMP;[MATRICE PRODUIT:].
POU;I=1,1,M
POU;J=1,1,N
;CIJ=0
POU;K=1,1,P
;CIJ=AIK.BKJ+CIJ
RET;
COM;11,'ROUGE'
PER;CIJ
'ROUGE'IMP;(5,2)CIJ
RET;
IMP;.
RET;
VER;'E1'
FIN;
```

Exploitation

Produit des matrices :

$$A = \begin{bmatrix} 1, & 0, & 1, \\ 0, & 0, & 0, \\ 1, & 0, & 1, \end{bmatrix} \quad \text{et } B = \begin{bmatrix} 1, & 0, & 1, \\ 0, & 0, & 0, \\ 1, & 0, & 1, \end{bmatrix}$$

Dimension des matrices : A(M.P) et B(P.N) en fixe.

M=3 N=3 P=3

Matrice produit :

2.00
0.00
2.00

0.00
0.00
0.00

2.00
0.00
2.00

Produit des matrices :

$$M_1 = \begin{bmatrix} 1, & -2, & 3 \\ 3, & 1, & 2, \end{bmatrix} \quad M_2 = \begin{bmatrix} 2, & 0, & 1, \\ 1, & 1, & -2, \\ -1, & 3, & 1, \end{bmatrix} \quad M_3 = \begin{bmatrix} 1, & 0, & 3, & 2, \\ 3, & -1, & 2, & -1, \\ 2, & 0, & 1, & -1, \end{bmatrix}$$

Dimension des matrices A(M.P) et B(P.N) en fixe.
M=2 N=3 P=3

Matrice produit $M_1 \cdot M_2 =$

3.00
7.00
8.00

5.00
7.00
3.00

- Perforation d'un ruban donnant les valeurs de la matrice $(M_1\ M_2)$

Dimension des matrices : A(M.P) et B(P.N) en Fixe.

M=2 N=4 P=3

Matrice produit $(M_1 \cdot M_2) \cdot M_3 =$

34.00
\- 7.00
13.00
\- 21.00

32.00
\- 7.00
32.00
\- .00

Problème 6

Calcul du P.G.C.D. de deux nombres A et B par l'algorithme d'Euclide.

On suppose que A)B

$$A = B \cdot Q_0 + R_1$$

Si $R_1 = 0$, B est P.G.C.D. sinon on écrit :

$$B = R_1 \cdot Q_1 + R_2$$

Si $R_2 = 0$, R_1 est P.G.D.C. sinon on écrit :

$$R_1 = B_2 \cdot Q_2 + R_3$$

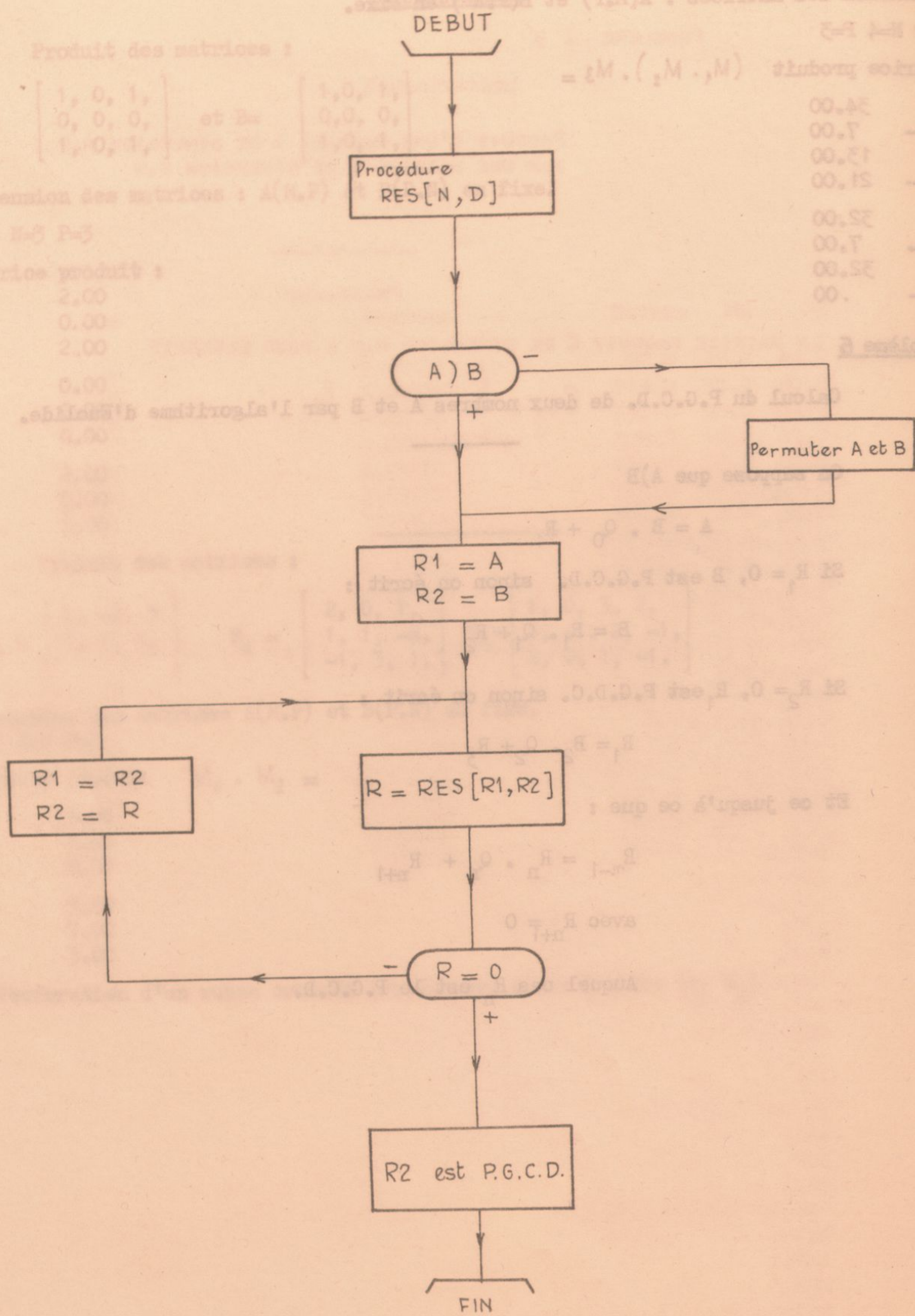
Et ce jusqu'à ce que :

$$R_{n-1} = R_n \cdot Q_n + R_{n+1}$$

avec $R_{n+1} = 0$

Auquel cas R_n est le P.G.C.D.

ORGANIGRAMME



Chapitre IV

NOTICE D'UTILISATION DU CALCULATEUR PB 250 ET DE SES ORGANES PERIPHERIQUES

4.1. IMPLANTATION DU COMPILATEUR

Le "Compilateur" occupe les ligne 02 à 26 soit presque la totalité des mémoires. Aussi, un ruban programme symbolique est souvent compilé en plusieurs fois : le programme généré est rangé au fur et à mesure dans les lignes libres. Lorsque ces mémoires sont remplies, le Compilateur arrête sa traduction, vide ces lignes en les sortant sur ruban perfoé et reprend sa compilation.

4.2. IMPLANTATION DES SOUS-PROGRAMMES ET DU PROGRAMME GENERE

Les sous-programmes sont implantés jusqu'à la ligne 11 incluse et le compilateur range alors le programme généré entre les lignes 12 et 40. Cependant, il est possible de faire varier ces limites ; pour cela, il suffit d'indiquer <u>en tête de programme</u> le numéro de la première ligne soit LL et celui de la dernière soit L'L' sous la forme :

, LL L'L' (virgule LL L'L')

<u>Restrictions</u>: il faut que :

$$10 \leqslant LL \leqslant 17$$

$$LL < L'L' \leqslant 40$$

En aucun cas, on ne peut avoir L L ou L'L' égal à 37.

L'introduction des rubans se faisant à l'aide d'un lecteur rapide, la ligne 07 comporte un programme - HSR 07 - qui gère le lecteur rapide. Ce programme est normalement détruit lorsqu'on passe en exploitation aussi on le transfère en ligne 12. Il faut alors implanter les programmes compilés entre les lignes 13 et 40. Pour d'autres raisons, il est recommandé de commencer les programmes par le caractère "corbeille basse" (CB) et de les terminer par le code "Stop".

En conclusion, les programmes se présentent toujours sous la forme :

```
CB ,      1340
        ------
        ------
        ------
        ------
        FIN : RC
        CODE "STOP"
```

Remarque : Si on connaît l'encombrement exact d'un programme, par exemple, entre les ligne 13 et 15, on peut loger plusieurs programmes dans le calculateur. Ils seront exploités à tour de rôle, lors de l'appel de chaque programme.

Ex.:

```
1er programme                2ème programme
 CB,    1315                  CB,    1640
 --------                     --------
 --------                     --------
 FIN :    RC                  FIN :    RC
 CODE "Stop"                  CODE "Stop"
```

Implantation en mémoire

```
00                    Ligne courte                                   16 mots
01                    OUP (ordres initiaux)                         256 mots
02 ---------------------------┐                                        "
03                            |                                        "
04                            |                                        "
05                            ├-> Sous-programmes généraux (entrée et sortie   "
                              |   décimale) et opérations arithmétiques        "
06 ---------------------------┘                                        "
07                    HSR 07 - Ligne de travail                        "
10 --------┐                                                           "
           ├-> Sous-programme fonctions                                "
11 --------┘                                                           "
12 --------------------------------------------------------------      "
13                                       Programme                     "
---                                          |                         "
---                                          v                         "
17                                                                     "
---                                                                    "
37                                           ^                         "
40 ----------------------------------------- Données                   "
```


4.3. ENCOMBREMENT EXACT D'UN PROGRAMME ET DES DONNEES

En fin de compilation, on peut connaître l'encombrement exact d'un programme et des données, en interrogeant certains secteurs par l'O.U.P. et en abaissant la touche marquée " Bouton Programmeur".

En 065 06, on peut lire en format donnée le numéro de la dernière ligne contenant le programme.

En 301 02, on peut lire en format commande le numéro du premier secteur libre après le programme (les adresses du programme vont en croissant).

En 011 00, on peut lire en format donnée la première adresse libre au-dessus des données (les adresses des données vont en décroissant).

Cette adresse est cadrée à 21, les 8 derniers bits à droite indiquent le numéro de secteur, les 6 bits précédant le numéro de secteur indiquent le numéro de ligne.

Ex.:

```
06506D+0000014
30102C110 0000;
01100D+0016502          +0016502 => 000 000 001 110 1|01 000 010
                                          \___/   5 | 1   0   2
                                            3
```

```
LIGNE        14           35        40
   ----------+------------+---------+
SECTEUR     110          102
```

4.3,1. IMPRESSION DE MISE AU POINT

On peut terminer une phrase de calcul en virgule flottante, comportant au moins une opération arithmétique, par le sign =.A l'exploitation il y aura impression du résultat de la phrase arithmétique si la touche "Bouton Programmeur" est abaissée. Les trois premières lettres du nom du résultat, suivies du signal "égal", suivi du résultat en virgule flottante, sont imprimées.

MAGE	EXPLOITATION

: VAR=2	
: VAR=VAR+1=	VAR=3000000000 E 1
: VD=VAR.1=	VD =3000000000 E 1

4.3,2. CORRECTION DU RUBAN PROGRAMME SYMBOLIQUE AU MOMENT DE LA COMPILATION

Il est possible de modifier un programme symbolique sans avoir à dupliquer le ruban. Il suffit d'introduire, avant de compiler, toutes les corrections que l'on veut faire dans un ordre quelconque.

Pour cela, on doit :

1. situer la correction dans le programme. On compte les instructions c'est-à-dire les lignes de programmes de 10 en 10, la ligne CB ,1340 non comprise. Ainsi, la nième phrase se voit affecter le numéro 10.N

```
    CB , 1340
10  ------
20  ------
30  ------
      ----
      ----
120  :A1=A1+1
130  :RA=√[A1]
     -----
```

2. perforer un ruban de correction. Une correction est constituée par une phrase MAGE selon la syntaxe suivante :

CB	Numéro de la correction	.	phrase complète correcte	RC

On peut ainsi :

A) Annuler une phrase du ruban

Ex.: CB 130 . TEX : RC

B) Remplacer une phrase erronée sur le ruban par une phrase correcte.

Ex.: CB 130 . :RA=A1.A1 RC

C) intercaler une ou plusieurs corrections entre deux phrases du ruban perforé. Si on désire intercaler une phrase supplémentaire entre la 12ème et la 13ème ligne, il suffit de perforer un ruban de correction portant le numéro 121. S'il y a deux instructions à intercaler, on leur donnera les numéros 121 et 122, etc....

Ex.:
```
CB 120 . : A1=A1+1/2   RC
CB 121 . : RA=A1*3     RC
CB 130 . : RA=√[RA]    RC
```


4.4. LIMITATION DU COMPILATEUR MAGE II PB 250

1. Le nombre d'identificateurs de variable est limite à 128.
2. Le nombre d'étiquettes est normalement limité à 14. Si ce nombre est insuffisant, il suffit d'indiquer le nombre d'étiquettes désiré en début de programme, sous forme d'un point suivi du nombre voulu d'étiquettes, en décimal. On est alors limité à 120.

Ex.: Nombre d'étiquettes désiré : 50.

```
CB, 1340. 50          CB . 50,  1340
-------               ---------
-------               ----------
```

3. Le nombre de boucles imbriquées est limité à 6.
4. Le nombre de procédures est limité à 10.
5. Le nombre de corrections (sous forme de ruban de corrections) est limité à 5. L'encombrement des corrections est limité à 765 caractères (non compris le numéro et le point qui suit).

4.5. DESCRIPTION DE LA MACHINE A ECRIRE

La machine à écrire est une flexowriter FRIDEN Modèle FX1.Elle est munie d'une perforatrice et d'un lecteur de ruban. Elle peut être utilisée connectée au calculateur (ON LINE) ou non (OFF LINE).

4.5,1. FONCTIONNEMENT DE LA MACHINE A ECRIRE

1. Interrupteur POWER sur ON : machine à écrire ordinaire.
2. Interrupteur PUNCH sur ALL: voyant lumineux allumé - les caractères tapés sur le clavier sont perforés suivant le code ci-joint.

- au point de vue typographique, les minuscules sont en fait de petites majuscules. Un programme peut être écrit indifféremment en majuscules (corbeille basse) ou en minuscules (corbeille haute) sauf dans les corps de phrase IMP : ou PER :
- Les caractères . , " ' : ; peuvent être utilisés indifféremment en corbeille haute ou basse.
- Pour faire avancer la bande sous la perforatrice, toujours utiliser la touche "avance bande".
- Lecture de rubans :

Les rubans perforés sur 6 canaux (code flexo) peuvent être relus par la machine à écrire. Pour cela, placer le ruban sur le lecteur lent et appuyer sur la touche "DEPART LECT". Pour arrêter la lecture, il suffit d'appuyer sur la touche "ARRET LECT".

Si, de plus, l'interrupteur PUNCH est sur ALL pendant la lecture d'un ruban, on perf[o]
en même temps un ruban identique. Cette méthode sert couramment pour corriger un rub[an]
symbolique.

- <u>Duplication de rubans</u> :

La touche "DUPL" permet de reproduire un ruban 6 canaux (en symbolique) ou 8 canaux (en machine) mais sans impression de contrôle.

- <u>Correction de bande au moment de la perforation</u> :

Lorsqu'on perfore un ruban symbolique, il arrive souvent que l'on fasse des erreurs de frappe.

A) Si on s'aperçoit tout de suite de l'erreur, reculer le ruban sous la per[fora]trice d'un ou plusieurs caractères, au besoin en relisant ces caractères au moyen du code flexo, puis perforer des codes "ERREUR" en appuyant sur la touche "CODE ERREUR" [les] perforations obtenues ne seront pas prises en compte par le calculateur.

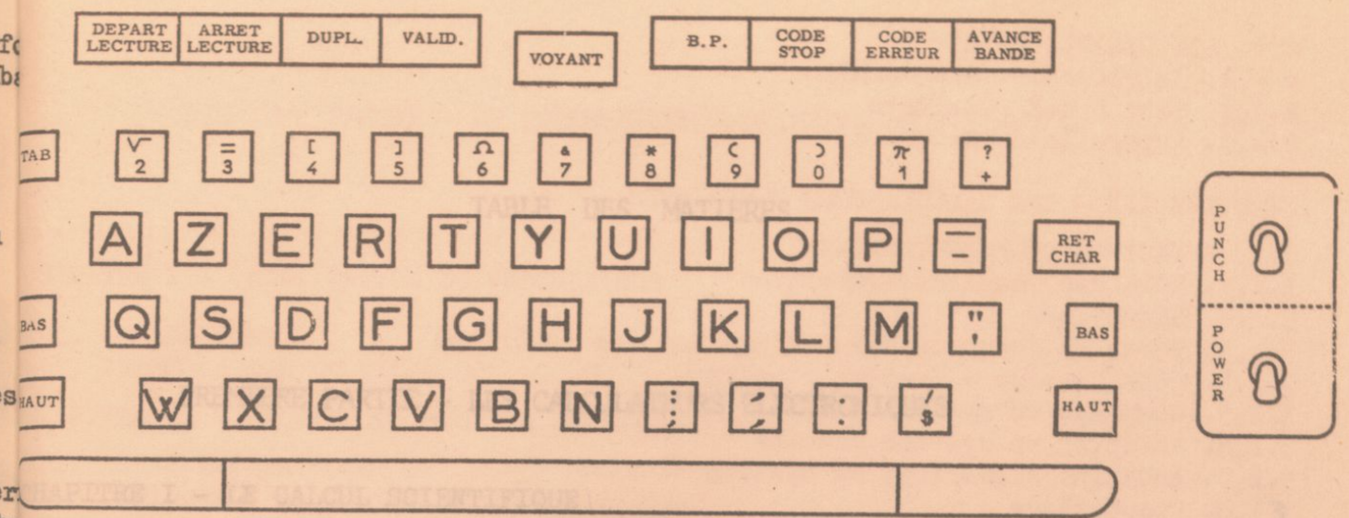
b) Si on s'aperçoit de l'erreur trop loin mais si le retour chariot termina[nt] l'instruction fausse n'est pas perforé, taper le code $ (dollar) qui annule la phras[e] en cours et recommencer la perforation au début de la phrase annulée.

<u>Ex.</u>: IMO: [LATUS] $IMP: [LATUS] (RC)
0 pour P

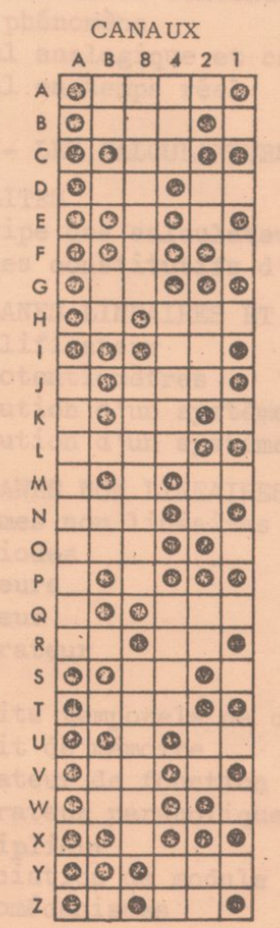
4.5,2. <u>FONCTIONNEMENT ON LINE</u>

Tout ce qui a été dit sur le fonctionnement OFF LINE reste valable pour le fonctionnement ON LINE.

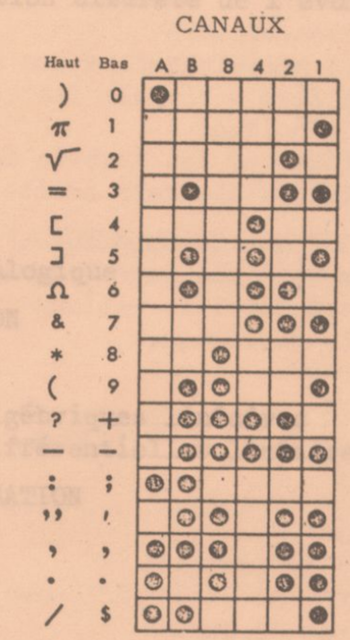
De plus, la perforatrice peut sortir des rubans en langage généré (sur 8 canaux) ou des résultats (PER :).



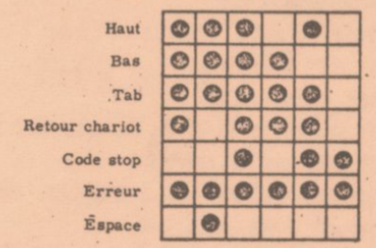
Clavier de la Flexowriter



CARACTERES ALPHABETIQUES
DISPONIBLES EN HAUT ET BAS



CARACTERES NUMERIQUES
ET SPECIAUX



CARACTERES DE
COMMANDE

CODE FLEXOWRITER

TABLE DES MATIERES

PREMIERE PARTIE - LES CALCULATEURS ELECTRONIQUES

DEUXIEME PARTIE - LE CALCULATEUR ANALOGIQUE NADAC 20

TROISIEME PARTIE - PROGRAMMATION SUR CALCULATEUR NUMERIQUE P.B.250

°°°