

LA STRUCTURE MATHÉMATIQUE SIMPLE DES GRANDES MACHINES À CALCULER

par J. VILLE

Professeur à la Faculté des Sciences de Paris

L'apparition, puis le développement des grandes calculatrices électroniques offrent aux Mathématiques des possibilités nouvelles, mais posent également des exigences nouvelles aux mathématiciens.

Les possibilités sont connues ; la plus commune est la rapidité et le bon marché avec lesquels on peut procéder à des opérations classiques, ce qui permet d'entreprendre des calculs auxquels on aurait jadis renoncé faute de temps et de moyens. Plus une machine va vite, plus elle remplace de calculateurs, et par conséquent plus elle est économique. Mais ce n'est pas tout ; à supposer qu'une machine travaille au même prix que des calculateurs manuels, si elle en remplace un grand nombre, elle rend possibles des calculs qui restaient irréalisables parce que le travail simultané d'un grand nombre de calculateurs manuels sur un même problème posait des problèmes d'organisation presque insurmontables. Une autre ressource des machines est apparue lorsque l'on a constaté que l'on pouvait leur faire traiter des problèmes non classiques ; ce fut le cas des problèmes où interviennent de très nombreuses inégalités ; lorsqu'on essaie de résoudre par un calcul « à la main », on est surpris par le développement qu'il prend, tellement il est compliqué.

Parlons maintenant des exigences qui vous intéressent particulièrement, puisque vous aurez sans doute à préparer des élèves qui devront y satisfaire, même si peu d'entre vous sont appelés, en tant que professeurs, soit à utiliser directement les machines, soit à être les bénéficiaires de leurs possibilités. Ces exigences sont dues à la structure des machines, structure que le titre de cet exposé annonce comme mathématiquement simple, mais qui, justement à cause de cette simplicité, requiert d'assez grands efforts pour être mise effectivement en œuvre.

Le premier principe de structure est que les machines ne savent actuellement travailler qu'en système binaire, c'est-à-dire dans le système de numération à base deux où chaque entier n est mis sous la forme :

$$n = \sum \alpha_k 2^k \quad \text{où } \alpha_k = 0 \text{ ou } 1 \text{ et } k \in \mathbb{N}.$$

Second principe de structure : chaque chiffre binaire est représenté par une impulsion électrique, c'est-à-dire par un courant pendant un temps très court. Si le courant est présent, cela représente le chiffre 1 ; s'il est absent, cela représente le chiffre 0.

Ces deux premiers principes n'ont pas, en eux-mêmes, de grandes conséquences. Mais un troisième principe intervient alors : un chiffre binaire ne peut se trouver qu'en deux états : soit *en circulation*, pour être traité par le calcul, soit *enregistré*, pour attendre d'être traité.

Ce dernier principe, qui paraît une évidence, a des conséquences importantes, parce qu'il fait intervenir une notion nouvelle, et même entièrement nouvelle, celle d'*adresse*. En effet, si nous admettons qu'une opération doit intervenir, telle que :

$$(1) \quad a + b = c,$$

l'essentiel n'est pas de connaître les valeurs numériques de a et b , puisque la machine sait faire l'addition, mais de savoir où il faut aller chercher a et b .

Pour comprendre ce qui va suivre, il faut insister sur la signification de l'égalité (1), qui paraît tout à fait simple. Elle signifie en réalité : « J'ai appelé a un certain nombre ; j'ai appelé b un certain nombre ; je calcule leur somme ; cette somme, je l'appelle c . »

Or, pour une machine numérique, il n'existe que des nombres. Donc, la façon dont on appelle un nombre ne peut être qu'un nombre. Ceci est d'ailleurs beaucoup plus souple. En effet, s'il était imposé de représenter des nombres par des lettres toujours de la même façon, un traité d'algèbre ne pourrait porter que sur des nombres en nombre égal à celui des lettres de l'alphabet. A partir du moment où l'on fait intervenir des indices, on commence à entrer dans un système de numération. Ces faits passent souvent inaperçus parce que, dans les Mathématiques classiques, l'esprit complète l'exposé. Par exemple, la phrase : « Je considère la suite des x_i , i variant de 1 à n », est parfaitement claire, mais pour une machine elle est terriblement compliquée. Elle serait également très compliquée pour quelqu'un qui refuserait de faire l'effort d'interprétation qu'on lui demande, et qui exigerait qu'on l'exprime à partir d'axiomes logiques.

Pour en revenir au « nom » des nombres, le plus simple est de nommer un nombre par l'emplacement qu'il a dans la machine. Par exemple, l'égalité :

$$\langle 15 \rangle + \langle 37 \rangle = \langle 203 \rangle$$

signifie : le nombre qui est à l'emplacement numéro 203 doit être la somme du nombre qui est à l'emplacement n° 15 et du nombre qui est à l'emplacement n° 37. Ce n'est qu'une généralisation des procédés employés couramment pour établir les feuilles de calcul ou même les déclarations d'impôt ; on y lit : inscrire dans la colonne 4 la somme de ce qui est inscrit dans la colonne 1 et dans la colonne 2, ce qui peut donc se noter :

$$\langle 1 \rangle + \langle 2 \rangle = \langle 4 \rangle .$$

Une adresse est une notion assez facile à comprendre sur le schéma suivant, où nous voyons apparaître deux éléments : a) un registre R ; b) un élément arithmétique EA (voir fig. 1).

Sur le registre sont inscrits les éléments (les nombres). Chacun est à un emplacement. L'élément arithmétique sait faire un certain nombre d'opérations. La conduite de la machine consiste à prélever des couples

de nombres dans le registre (choix β), à choisir l'opération (choix γ), — après quoi celle-ci est effectuée —, et à choisir l'endroit où l'on place le résultat (choix α).

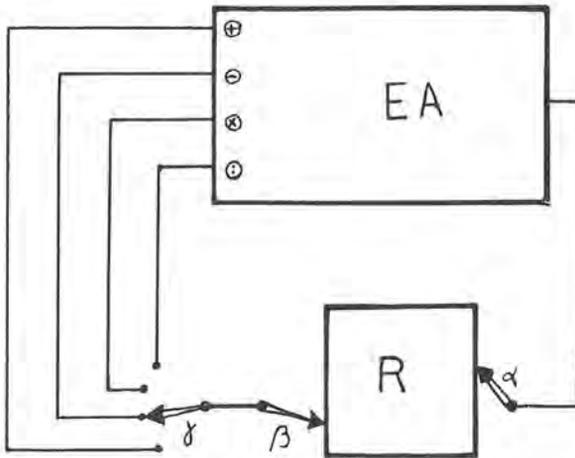


FIG. 1

Un ordre se présenterait ainsi sous la forme (β, γ, α) et, pour un choix convenable des adresses des données, on peut toujours trouver une suite d'ordres qui amène le résultat à des adresses données. Si, par exemple, j'écris :

$$\begin{aligned} ax + by &= c \\ a'x + b'y &= c' \end{aligned}$$

sous la forme :

$$\begin{aligned} \langle 1 \rangle \langle 7 \rangle + \langle 2 \rangle \langle 8 \rangle &= \langle 3 \rangle \\ \langle 4 \rangle \langle 7 \rangle + \langle 5 \rangle \langle 8 \rangle &= \langle 6 \rangle . \end{aligned}$$

ce qui revient à mettre les données dans les cases numérotées de 1 à 6 et à porter les inconnues dans les cases de numéros 7 et 8, on peut trouver une chaîne de calculs et d'ordres amenant au résultat.

Ceci suppose que l'on adopte une méthode de calcul (c'est-à-dire un ordre d'opérations) et que l'on choisit à l'avance toutes les adresses. C'est là qu'interviennent les servitudes propres aux machines et les exigences relatives aux calculs. Il est un fait que l'on peut résoudre sur machine des systèmes de cent équations du premier degré à cent inconnues. On sait que le déterminant d'un tel système contient $100!$ termes. Si rapide que soit une machine et si grande que soit la capacité de ses registres, il est impossible de calculer tous ces termes et de les enregistrer à des adresses différentes.

Il faut donc mener le calcul de la façon la plus économique, tant en nombre d'opérations qu'en nombre de symboles intermédiaires. Ces deux exigences font appel à des notions simples, mais parfois négligées. Par exemple, s'il s'agit de calculer la valeur numérique d'un polynôme :

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n,$$

pour une valeur α de x , on constate que l'on a besoin de moins d'opérations en procédant à la division de $f(x)$ par $x - \alpha$ qu'en procédant à la substitution, et que l'on a moins de registres à utiliser. De plus l'opération se met sous forme séquentielle simple, en posant :

$$\begin{aligned} f_0 &= a_n \\ f_1 &= f_0\alpha + a_{n-1} \\ f_2 &= f_1\alpha + a_{n-2} \\ &\dots\dots\dots \\ f_n &= f_{n-2}\alpha + a_0 = f(\alpha). \end{aligned}$$

Nous avons procédé à n multiplications et n additions et nous n'avons eu besoin que de deux mémoires, car nous n'avons pas besoin de conserver les valeurs f_i pour $i < n$; dès que f_i est calculé, f_{i-1} peut être effacé.

En ce qui concerne la résolution d'un système d'équations linéaires, la méthode que l'on peut utiliser consiste à tirer une inconnue d'une équation et à porter sa valeur dans les autres équations. Par exemple, le système :

$$\left\{ \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \dots\dots\dots \\ a_{n1}x_1 + \dots\dots\dots &= b_n \end{aligned} \right.$$

est ramené au système :

$$\left\{ \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ \left(a_{22} - \frac{a_{21}a_{12}}{a_{11}} \right) x_2 + \dots &= b_2 - \frac{a_{21}b_1}{a_{11}} \\ \dots\dots\dots \end{aligned} \right.$$

ce qui exige $(n - 1)^2$ fois le calcul de termes tels que :

$$a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}} ;$$

les quotients $\frac{a_{ij}}{a_{11}}$ sont au nombre de $n - 1$. Nous aurons donc $(n - 1)$ divisions, $(n - 1)^2$ multiplications, $(n - 1)^2$ soustractions, c'est-à-dire $(n - 1)(2n - 1)$ opérations.

Pour les seconds membres, nous aurons une division $\left(\frac{b_1}{a_{11}}\right)$, $(n - 1)$ multiplications, $(n - 1)$ soustractions, soit $(2n - 1)$ opérations. Cette première étape de la réduction du système aura donc exigé $n(2n - 1)$ opérations. La réduction totale ne fera donc intervenir que :

$$n(2n - 1) + (n - 1)(2n - 3) + \dots + 6 \sim \frac{n^3}{3} \text{ opérations.}$$

La remontée du procédé en fait intervenir la moitié. En tout, le système est résolu en $\frac{n^3}{3}$ opérations. On est loin des $n!$ termes du déterminant. Quant au déterminant lui-même, son calcul peut également se faire par récurrence.

Si l'on veut donner des ordres à la machine pour effectuer le calcul, et que l'on écrive *tous* ces ordres, on s'aperçoit que cela est compliqué et le choix des mémoires est également compliqué. Il y a évidemment beaucoup d'intermédiaires, leur nombre est du même ordre que celui des opérations. Mais si on écrit les relations de récurrence sous la forme :

$$a_{ij}^{(k)} = \frac{a_{ik}^{(k-1)} a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

k désignant le numéro du pas de réduction, on remarque que les $a_{ij}^{(k)}$ n'ont pas besoin d'être tous conservés, et que leur adresse naturelle est (i, j) . Quant aux ordres, ils ne dépendent pas de (k) en réalité, puisque l'opération est toujours la même ; ce qui dépend de k , c'est l'adresse des nombres à opérer. Et, d'un pas de réduction à l'autre, ces adresses varient d'une unité. On conçoit donc qu'à côté des opérations purement numériques, on ait à opérer sur des choses qui ne sont pas des nombres, mais des *noms* (écrits sous forme numérique), ces noms pouvant désigner des emplacements, l'emplacement étant occupé par un nombre proprement dit, un ordre, ou même le nom d'un autre emplacement.

Les ordres eux-mêmes peuvent ne pas être des ordres d'opération, mais des ordres à aspect logique. Pour calculer une suite convergente avec une certaine précision, par exemple, on pourra convenir de l'ordre suivant : « Si $x_n = x_{n-1}$, s'arrêter. Si $x_n \neq x_{n-1}$, passer à l'exécution de l'ordre $(n + 1)$ ». C'est là qu'intervient une algèbre logique binaire, les alternatives étant simples, en général.

Ces choix sont assez rares dans les calculs classiques, mais ils peuvent intervenir à tout instant dans les calculs non classiques pour lesquels des machines très puissantes deviennent nécessaires. Parmi ces calculs, il faut citer ceux où interviennent des inégalités ; elles demandent des opérations de choix très nombreuses. Les systèmes d'inégalités linéaires de la forme

$$\sum a_{ij} x_j \leq b_i$$

sont encore très peu étudiés. Les méthodes classiques d'élimination conduisent à un nombre énorme d'opérations ; de plus, elles obligent à conserver des éléments trop nombreux.

L'emploi des machines contraint donc à des études spéciales portant principalement sur l'algèbre et où interviennent un très grand nombre de décisions. Ces problèmes d'algèbre sont élémentaires, mais ils n'ont pas encore été beaucoup étudiés.