MÉMOIRES DE LA S. M. F.

JACQUES CALMET

Utilisation des langages formels en théorie des champs

Mémoires de la S. M. F., tome 49-50 (1977), p. 31-39

<http://www.numdam.org/item?id=MSMF_1977__49-50__31_0>

© Mémoires de la S. M. F., 1977, tous droits réservés.

L'accès aux archives de la revue « Mémoires de la S. M. F. » (http://smf.emath.fr/Publications/Memoires/Presentation.html) implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

Calculateurs en Math. (1975 - LIMOGES)
Bull. Soc. math. France
Mémoire 49-50, 1977, p. 31-39

UTILISATION DES LANGAGES FORMELS EN THEORIE DES CHAMPS

par

Jacques CALMET

I.- INTRODUCTION -

Depuis quelques années, les langages formels ou systèmes de manipulations algébriques sont utilisés pour effectuer des calculs de théorie des champs et plus particulièrement d'Electrodynamique Quantique. Il existe actuellement un grand nombre de tels langages. Par exemple, en se référant aux Comptes-Rendus de SIGSAM 2 [1], EUROSAM 74 [2], et des Colloques de Marseille [3], on peut citer REDUCE, FORMAC, ALTRAN, MACSYMA, SAC-1, MATHLAB, SYMBAL, SCRATCHPAD, CAMAL, IAM, ALAM, (devenu aujourd'hui CLAM), PORT-ALG, ANALITIK, SCHOONSCHIP, ASHMEDAI. Cette liste n'est évidemment pas exhaustive. On trouve, en outre, dans la littérature, de nombreux programmes écrits en LISP qui résolvent des problèmes particuliers dans des disciplines diverses allant de la théorie des groupes à la chimie organique.

Parmi les langages cités ici seuls les huit premiers sont largement utilisés. Ceci est dû au fait qu'ils sont en général :

(i) efficaces (temps de calcul, nombre de mémoires occupées, facilité de programmation, valeur des algorithmes utilisés ...).

(ii) généraux en ce sens qu'ils permettent le traitement de différents type de problèmes.

(iii) transportables d'un ordinateur à d'autres (excepté FORMAC qui est écrit en PL1).

Le caractère de généralité ne peut évidemment pas être sans limite, c'est la raison pour laquelle des langages plus spécialisés ont vu le jour. On constate en effet une double évolution : certains langages sont de plus en plus utilisés mais en même temps d'autres langages sont développés afin de résoudre des problèmes particuliers.

La discipline qui nous intéresse ici est une bonne illustration de cette évolution. En effet, nous avons à effectuer certains calculs algébriques connus sous le nom d'algèbre des matrices de Dirac [4] que REDUCE [5] permet de faire. Il est à noter que REDUCE a pour motivation originelle ce type de calcul mais qu'il a évolué de telle façon que la partie consacrée à l'algèbre de Dirac ne constitue plus qu'un module du langage. SCHOONSCHIP a été écrit à la même époque mais n'a pas évolué en un langage " général " et ne reste utilisable que sur CDC car il est en code machine Enfin récemment un nouveau système de programmes propre à cette discipline a vu le jour; il s'agit d'ASHMEDAI écrit en FORTRAN.

Ces trois " langages " ont en commun un certain nombre d'algorithmes qui rendent possibles les calculs d'algèbre de Dirac. Un de ces algorithmes est décrit dans la première partie de l'exposé. La deuxième partie sera consacrée au problème de l'intégration formelle. Actuellement ce problème, bien que de remarquables progrès ont été faits durant ces dix dernières années, n'a pas de solution qui permette le traitement effectif de problèmes non triviaux. Nous essaierons de montrer comment un utilisateur peut enrichir un langage donné en créant un outil destiné à intégrer une certaine classe de fonctions.

Le but de cet exposé n'est pas de décrire de façon détaillée les difficultés techniques que l'on rencontre en Théorie quantique des champs traitée de manière informatique. Il est de montrer que bien souvent un utilisateur doit, dans le cadre d'un langage choisi de manière adéquate, développer lui-même les méthodes et programmes dont il a besoin.

L'appendice contient un bref exposé du problème physique auquel ces méthodes sont appliquées.

II.- ALGEBRE DE DIRAC -

Les matrices de Dirac [4] sont un ensemble de 4 matrices 4 × 4, appelées γ_μ (μ = 1, 2, 3, 4). Elles sont définies par la relation d'anticommutation suivante :

$$\gamma_\mu \gamma_\nu + \gamma_\nu \gamma_\mu = 2\, g_{\mu\nu}\, I \qquad (1)$$

I est la matrice unité, $g_{\mu\nu}$ est le tenseur de la métrique utilisée ($g_{ij} = 0$, $g_{11} = g_{22} = g_{33} = -g_{44} = 1$).

Les éléments de ces matrices ne sont jamais utilisés lors des calculs. Ce sont des nombres réels ou imaginaires. Ces matrices formelles apparaissent en fait sous forme de sommes de produits du type

$$S_n = \gamma_{\mu_1} \gamma_{\mu_2} \cdots \gamma_{\mu_n} \qquad (2)$$

Deux types d'opérations sont à effectuer sur ces produits :

1°) Calculer leur trace

2°) Sommer sur les indices répétés qui peuvent apparaître dans S_n .

Le calcul des traces est extrêmement simple en principe. Il suffit d'utiliser les relations suivantes :

$$\begin{aligned}
&\mathrm{Tr}(I) = 4 \\
&\mathrm{Tr}(\gamma_\mu \gamma_\nu) = 4\, g_{\mu\nu} \qquad (3)\\
&\mathrm{Tr}(S_{2n+1}) = 0 \qquad \forall\, n \geqslant 0 \\
&\mathrm{Tr}(S_{2n}) = g_{\mu_1 \mu_2}\, \mathrm{Tr}(\gamma_{\mu_3} \cdots \gamma_{\mu_{2n}}) \\
&\qquad - g_{\mu_1 \mu_3}\, \mathrm{Tr}(\gamma_{\mu_2} \gamma_{\mu_4} \cdots \gamma_{\mu_{2n}}) \\
&\qquad \cdots\cdots\cdots\cdots\cdots \\
&\qquad + g_{\mu_1 \mu_{2n}}\, \mathrm{Tr}(\gamma_{\mu_2} \cdots \gamma_{\mu_{2n-1}})
\end{aligned}$$

$Tr(S_{2n})$ engendre (2n) !! termes. La nécessité d'utiliser une méthode informatique vient de la prolifération de tels termes. Il n'existe pas de très bons algorithmes pour calculer $Tr(S_{2n})$. Les meilleurs sont basés sur une décomposition tensorielle de S_{2n} obtenue en utilisant les propriétés des γ_μ . (On montre que tout produit S_n est décomposable sur une base de 16 éléments).

Les expressions dont on doit prendre la trace peuvent être réduites si l'on effectue d'abord la somme sur les indices répétés. En effet, sauf dans les cas extrêmement simples, on a toujours à faire ce genre de somme. De là vient l'importance de l'algorithme de Kahane [6] qui simplifie considérablement cette opération. Nous allons montrer sur deux exemples comment fonctionne cet algorithme.

Exemple 1 : Soit à calculer

$$X = \sum_{\alpha_1 , \alpha_2} \gamma_{\alpha_1} \gamma_{\mu_1} \gamma_{\alpha_2} \gamma_{\mu_2} \gamma_{\mu_3} \gamma_{\alpha_1} \gamma_{\mu_4} \gamma_{\alpha_2} \quad (4)$$

en utilisant les relations d'anticommutation (1) on obtient :

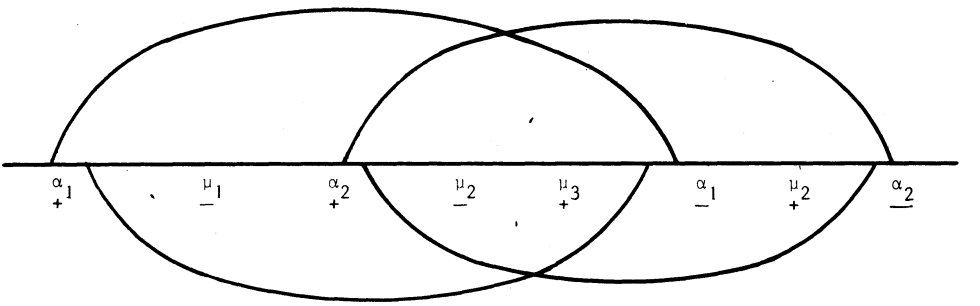
$$X = 4 \gamma_{\mu_4} \gamma_{\mu_2} \gamma_{\mu_3} \gamma_{\mu_1} \quad (5)$$

l'inconvénient de cette méthode est qu'elle engendre plusieurs termes qui après simplification se réduisent à un seul. L'algorithme élimine cet inconvénient. Il s'applique de la manière suivante :

1) On place les indices sur une droite, dans l'ordre où ils apparaissent dans X.

2) On joint les indices sur lesquels on somme par des lignes dans le demi-plan supérieur.

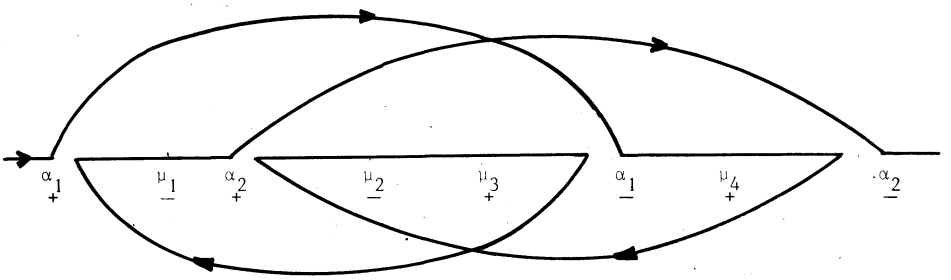
3) On assigne une parité à chaque indice : l'indice le plus à gauche a la parité +, ensuite on alterne évidemment les parités :



4) Lorsqu'un indice de somme est de parité + on déplace cet indice légèrement vers la droite, lorsqu'il est de parité - on le déplace vers la gauche.

5) On joint, dans le demi-plan inférieur les indices de sommes ainsi déplacés.

6) On coupe la droite entre les points originaux et les points déplacés; on obtient ainsi un circuit que l'on parcourt depuis la gauche pour obtenir le résultat cherché : $\gamma_{\mu_4}\ \gamma_{\mu_2}\ \gamma_{\mu_3}\ \gamma_{\mu_1}$



7) Le coefficient est donné par la relation suivante

$$2^K\ (-1)^{K-k} \qquad (6)$$

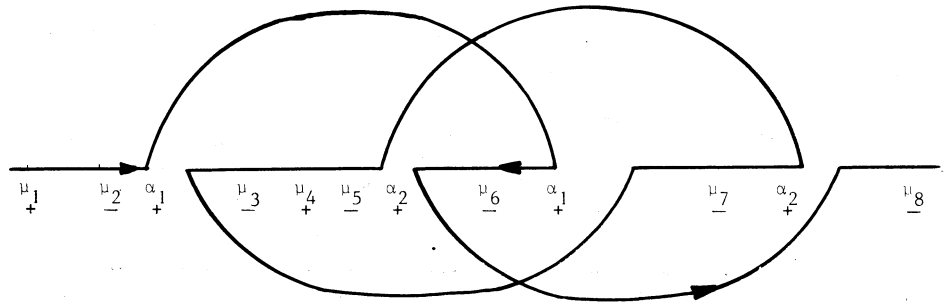
où K est le nombre d'indices de somme (2 ici) et k le nombre de boucles fermées dans le graphe obtenu (0 dans ce cas). D'où

$$2^2\ (-1)^0 = 4$$

Exemple 2 : Nous examinons maintenant un exemple où k est différent de zéro. Soit

$$X' = \sum_{\alpha_1\ ,\ \alpha_2} \gamma_{\mu_1}\ \gamma_{\mu_2}\ \gamma_{\alpha_1}\ \gamma_{\mu_3}\ \gamma_{\mu_4}\ \gamma_{\mu_5}\ \gamma_{\alpha_2}\ \gamma_{\mu_6}\ \gamma_{\alpha_1}\ \gamma_{\mu_7}\ \gamma_{\alpha_2}\ \gamma_{\mu_8} \qquad (7)$$

En appliquant l'algorithme ainsi qu'il est décrit ci-dessus, on obtient le graphe suivant :



dont le résultat est

$$X' = -4\ \gamma_{\mu_1}\ \gamma_{\mu_2}\ \gamma_{\mu_6}\ \gamma_{\mu_8} \times \{\text{boucle}\} \qquad (8)$$

où {boucle} désigne la contribution de la boucle fermée. Cette contribution s'obtient de la façon suivante : on examine la parité de l'indice le plus à gauche dans la boucle (ici μ_3); si la parité est − on décrit la boucle dans le sens trigonométrique, si elle est + on la décrit dans le sens inverse. Pour cet exemple :

$$\{\text{boucle}\} = \{\gamma_{\mu_3}\ \gamma_{\mu_7}\ \gamma_{\mu_5}\ \gamma_{\mu_4}\}$$

cette quantité est appelée " brace ". Elle est définie par :

$$\{S\} = S + S_R \qquad (10)$$

où S_R signifie que l'on renverse l'ordre des facteurs de S

$$\{\gamma_{\mu_3}\ \gamma_{\mu_7}\ \gamma_{\mu_5}\ \gamma_{\mu_4}\} = \gamma_{\mu_3}\ \gamma_{\mu_7}\ \gamma_{\mu_5}\ \gamma_{\mu_4} + \gamma_{\mu_4}\ \gamma_{\mu_5}\ \gamma_{\mu_7}\ \gamma_{\mu_3} \qquad (11)$$

On vérifie par un calcul direct que l'algorithme donne bien la solution exacte de [7] .

Cet algorithme ne s'applique pas au cas où l'on a des expressions de la forme

$$\mathrm{Tr}(\gamma_\mu\ S)\ \mathrm{Tr}(\gamma_\mu\ S') \qquad (12)$$

Chisholm et Hearn [7] l'ont généralisé à ce cas.

Les deux exemples ici décrits montrent que cet algorithme est très facile à formuler et à programmer. On s'attendrait donc à le voir implémenté dans différents langages. Il est permis de supposer que cela n'est pas le cas car sa description dans [6] est trop " littéraire " et par suite seul un spécialiste peut le comprendre. Sans entrer dans une polémique au sujet de la façon de présenter un algorithme on peut tout de même souhaiter que tout algorithme soit présenté sous une forme logique qui permette une implémentation rapide; il peut s'agir de la description de Knuth [8] ou de Loos [9] dans ALDES (Algorithm Descriptor) par exemple.

III.- INTEGRATION FORMELLE -

Le programme d'intégration formelle le plus élaboré est certainement celui de Moses [10]. Il est connu sous le nom de SIN et permettait dans sa première version d'intégrer les fonctions " classiques ", c'est-à-dire, en fait de reproduite une bonne table d'intégrales. Il a été ensuite généralisé aux fonctions de Spence et à la fonction erreur

$$\mathrm{er}\ f(x) = \int_0^\infty \exp\ (-x^2)\ dx$$

La méthode utilisée repose sur des algorithmes dûs à Risch (références dans [10]) qui postulent que si l'intégrant et l'intégrale appartiennent à une même classe de fonctions, l'intégrale est calculable. SIN permet de calculer des intégrales simples. Son utilisation n'est pas assez souple pour que l'on puisse l'employer pour des intégrales multiples nombreuses. De plus la place mémoire requise est très importante. Ceci implique sans doute les efforts faits pour avoir des programmes qui soient sans doute moins généraux mais plus facilement utilisables. Le programme - écrit en REDUCE - de Stoutemyer [11] est à cet égard remarquable : il permet d'intégrer un grand nombre de fonctions simples bien qu'il ne comporte que très peu de code. Il est à souhaiter que ces genres de programmes se multiplient.

La prochaine étape en ce qui concerne les intégrales unidimensionnelles est le calcul des intégrales elliptiques. Ce travail est en cours dans SIN. Une approche

différente, et très intéressante, est due à Ng [2]. Il se propose d'utiliser les algorithmes de Carlson. Cette méthode s'applique lorsque l'intégrant s'exprime en fonction d'un certain nombre de fonctions de bases. Outre les fonctions élémentaires Ng utilise trois intégrales canoniques de Legendre.

Les méthodes que nous venons de citer ne sont pas facilement utilisables pour le type d'intégrales que nous avons à calculer. Ce sont des intégrales multiples d'un type bien déterminé. Il a été remarqué en particulier, que l'on avait à évaluer dans de nombreux cas des intégrales du type suivant :

$$I(k, m, n, r, s) = \int_0^1 dy\, dx\, dw\, dz \qquad (13)$$
$$* \quad \frac{\Delta^m}{\beta^n \gamma^k} \; z^r \; w^s \; x^{2k+n-m-3} \; y^{k+n-2}$$

où
$$\beta = 1 - x\, y(1 - w\, z\, (1 - w z))$$
$$\Delta = 1 - x\, (1 - w\, z\, (1 - w z))$$
$$\gamma = w^2 \; x \; \beta + y \; \Delta^2$$

L'expérience montre que les solutions s'expriment sous forme de fonctions élémentaires et des fonctions de Spence $L_n(x)$ qui sont définies par :

$$L_n(x) = \int_1^x \frac{dt}{t - 1} \, L_{n-1}(t)$$
$$L_o(t) = \frac{t - 1}{t} \qquad (14)$$

Peterman [12] a écrit un programme (SINAC) qui permet l'évaluation des intégrales de ce type qui apparaissent effectivement dans le calcul de certains effets physiques. La méthode utilisée est basée sur les fonctions de Nielsen :

$$S_{n,p}(x) = \frac{(-1)^{n+p-1}}{(n-1)\,!\;p\,!} \int_0^1 dt \, \frac{1}{t} \, \mathrm{Log}^{n-1}(t) \, \mathrm{Log}^{p}(1 - x\, t) \qquad (15)$$

qui peuvent également être exprimées sous forme de série :

$$S_{n,p}(x) = \sum_{s=o}^{\infty} \frac{(-1)^s}{(p + s)\,!} \, St_{p+s}(p) \, \frac{x^{p+s}}{(p + s)^n} \quad (x \in R, \; |x| < 1) \qquad (16)$$

où $St_k(p)$ sont les nombres de Stirling de première espèce :

$$\mathrm{Log}^j(1 + x) = j\,! \sum_{k=j}^{\infty} St_k(j) \, \frac{x^k}{k!} \; ; \quad |x| < 1 \qquad (17)$$

Lorsque l'on soustrait les $(m - p - 1)$ premiers termes au voisinage de $X = 0$ on obtient les fonctions de Nielsen soustraites généralisées ou polylogarithmes :

$$S^{m,k}_{n,p}(x) = \sum_{i=o}^{\infty} \frac{(-1)^{m-p+i} \; x^i}{(m + i)\,!\;(k + i)^n} \, St_{m+i}(p) \qquad (18)$$

On constate que les solutions de (13) sont données - outre des polynômes - par des termes de la forme $S^{m,m}_{n,1}$ lors des calculs effectifs.

Ces termes sont à leur tour exprimables sous forme des constantes transcendantes suivantes :

$$\xi(p) = \sum_{n=1}^{\infty} \frac{1}{n^p}$$

$$a_p = \sum_{n=1}^{\infty} \frac{1}{2^n\, n^p} \qquad (19)$$

$$\mathrm{Log}\ 2 = \sum_{n=1}^{\infty} \frac{1}{n\, 2^n}$$

$\xi(p)$ est la fonction de Rieman.

SINAC a été écrit en SCHOONSCHIP. Il a été adapté par Fox et Hearn [13] en REDUCE. Actuellement, il serait souhaitable d'avoir des programmes permettant le calcul d'intégrale à 6 et 7 dimensions; mais la structure analytique des solutions est encore trop mal connue pour espérer avoir une solution générale. Nous savons tout de même que nous aurons à évaluer des intégrales elliptiques.

IV.- CONCLUSION -

Si l'on doit tirer une leçon de ce qui s'est passé en théorie des champs quantique traitée de façon informatique, il semble qu'elle doit être la suivante : l'utilisateur qui veut résoudre un problème spécifique ne doit pas rester passif et attendre qu'un langage lui apporte toutes les techniques dont il a besoin. Il doit s'efforcer de créer lui-même ces techniques lorsqu'elles n'existent pas encore. Il est juste d'ajouter qu'en règle générale, il recevra toute l'aide possible de l'auteur du langage qu'il utilise, pour peu qu'il se donne la peine de le contacter.

APPENDICE

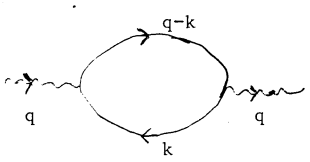
Cet appendice contient une brève description des types de problèmes d'Electrodynamique Quantique qui sont résolus en utilisant des langages formels. Une description plus détaillée est donnée dans [14]. Les effets étudiés viennent de l'interaction des électrons et des photons. Ils sont représentés sous forme de graphes. Les solutions sont obtenues sous forme de séries de perturbations. A chaque ordre de la série correspond un certain nombre de graphes. Pour les ordres les plus bas, les calculs sont triviaux et ne nécessitent aucun langage formel, mais dès que l'on veut évaluer des contributions d'ordre élevé on se heurte à deux difficultés : le nombre de graphes croît très rapidement et chacun de ces graphes est beaucoup plus difficile à calculer. A titre d'exemple, on peut citer les contributions au moment magnétique anormal de l'électron. L'ordre le plus bas comporte 1 graphe, à l'ordre suivant on a 5 graphes et ensuite 40 graphes. L'ordre qui suit n'a pas été calculé; il impliquerait le calcul de plus de 1000 graphes.

Ces calculs peuvent se décomposer en les étapes suivantes :

1) Engendrer les graphes. Pour ce faire, différentes méthodes et techniques sont utilisables. Des références sont données dans [14].

2) Etudier la convergence d'un graphe. Certaines contributions contiennent des quantités infinies qui doivent être soustraites en utilisant une théorie dite de la renormalisation. L'étude de la convergence d'un graphe peut être faite en examinant les propriétés topologiques de ce graphe [15].

3) A chaque graphe, on fait correspondre sa contribution, exprimée de façon algébrique, par un ensemble de règles dites de Feynman. Par exemple à l'ordre 2 le graphe ci-dessous



qui signifie qu'un photon d'impulsion q donne naissance à une paire électron-positron, d'impulsions k et $(q-k)$, qui se recombine ensuite en un photon d'impulsion q, aura pour contribution

$$\Pi_{\mu\nu}(q) \simeq \int d^4 k \, \frac{\mathrm{Tr}\left[\gamma_\mu (i(\not{q} - \not{k}) - m)\, \gamma_\nu (i \not{k} - m)\right]}{\left[(q-k)^2 + m^2\right]\left[k^2 + m^2\right]} \qquad (20)$$

où $\not{k} = \sum_{\rho=1}^{4} \gamma_\rho \, k_\rho$

4) Les dénominateurs sont combinés en utilisant une identité de la forme

$$\frac{1}{a_1 \, a_2 \cdots a_n} = (n-1)\,! \int_0^1 d\alpha_1 \cdots d\alpha_n \, \frac{\delta\left(1 - \sum_1^n \alpha_i\right)}{\left[\sum \alpha_i \, a_i\right]^n} \qquad (21)$$

où les a_i sont les facteurs du dénominateur de (20).

5) L'algèbre de Dirac venant du numérateur ayant été effectuée, on intègre sur les impulsions des boucles internes grâce à des équations du type :

$$\int d^4 k \, \frac{(k^2)^{m-2}}{(k^2 + A^2)^n} = \frac{i \, \pi^2}{(A^2)^{n-m}} \, \frac{(m-1)\,!\,(n-m-1)\,!}{(n-m)\,!} \qquad (22)$$

Lorsque l'on a éliminé les parties divergentes, on obtient des intégrales multiples sur les paramètres introduits par (21). Le calcul d'un processus physique déterminé nécessite d'autres opérations qui peuvent être, par exemple, la décomposition tensorielle des expressions obtenues afin d'isoler un certain élément.

Les langages formels permettent d'effectuer toutes ces manipulations, à l'exception toutefois des intégrales multiples d'ordre élevé qui ne peuvent bien

souvent être calculées que par des méthodes numériques. La situation est la même lorsqu'on utilise des méthodes différentes de celle, très brièvement esquissée ici.

REFERENCES

[1] " Proceedings of the Second Symposium on Symbolic and Algebraid Manipulation" Los Angeles, Mars 1971, édités par l'A.C.M.

[2] " Proceedings of EUROSAM 74 ", SIGSAM Bulletin, 8 (3) (1974)

[3] " Colloques sur les Méthodes Avancées de Calcul en Physique Théorique ".- Marseille 1970, 1971, 1973), édités par A. Visconti, CNRS Marseille.

[4] Voir par exemple ' Théorie Quantique des Champs ", Tome 1, par A. Visconti édité par Gauthier-Villars (1961).

[5] E. TOURNIER, ces comptes-rendus

[6] J. KAHANE.- Journ. Math. Phys. 9, 1732 (1968).

[7] J.S.R. CHISHOLM.- Comp. Phys. Comm. 4, 205 (1972)

[8] D. KNUTH.- The Art of Computing Programming.- vol 1, Addison-Wesley (1969)

[9] R.G.K. LOOS.- SIGSAM Bulletin (ACM), 23, 16 (1972)

[10] J. MOSES.- Comm. A.C.M., 14 (8), 548 (1971)

[11] D. STOUTEMYER.- Prétirage UCP-35.- Université d'Utah, Mai 1975.

[12] A. PETERMAN.- Comp. Phys. Comm., 7, 301 (1974).

[13] J. FOX, A.C. HEARN.- J. Comp. Phys., 14, 301 (1975)

[14] J. CALMET.- Comp. Phys. Comm. 4, 199 (1972).

[15] J. CALMET.- SIGSAM Bulletin, 8(3), 74 (1974).

Jacques CALMET
Centre de Physique Théorique
du C.N.R.S.
31, Chemin J. Aiguier
13274 MARSEILLE CEDEX 2