JOURNAL OF COMPUTATIONAL PHYSICS 89, 246–250 (1990)

Note

Systematic Generation of Linear Graphs—Check and Extension of the List of Uhlenbeck and Ford

1. Introduction

Consider the problem of calculating the partition function of a system of p particles that interact via a pair potential. As is shown in every textbook on statistical mechanics [1], this problem requires the evaluation of a number of integrals each of which is in one-to-one correspondence with a linear graph with p points.

If one wants to obtain an exact result for the partition function one has to evaluate all these integrals analytically. This requires, first of all, a complete list of the corresponding graphs and their invariance groups. The exact calculation of the partition function and the derivation of the resulting equation of state for clusters of hard particles [2] was the motivation for developing an algorithm to generate a list of free graphs. The related virial expansion for the infinite system is discussed in Ref. [3].

2. The Algorithm

Linear graphs may be represented in several ways; here it is done by assigning a number to each graph. The binary representation of this number was constructed in the following way:

(i) First the points of the graph are labelled by $1, 2, ..., p$.

(ii) All possible connections (bonds) (i, j) of two points i and j are numbered according to the sequence $(1, 2), (1, 3), (1, 4), ..., (2, 3), (2, 4), ..., (p-1, p)$. These numbers range from 0 to $p(p-1)/2-1$.

(iii) If the points i and j are connected by a bond the bit assigned to the number of the pair (i, j) is set equal to 1; if they are disconnected it is set equal to 0. (1)

The problem with such a representation is that the resulting numbers depend on the way the points had been labelled in step (i) of scheme (1). Starting from different labelings of the points one gets two different representing numbers for one and the same graph. To decide whether a given number represents a given graph

0021-9991/90 $3.00

it is necessary to consider all possible permutations in step (i) and to compare the resulting numbers with the given one. The drawback of this strategy is that $p!$ permutations have to be performed before two different numbers can be identified as representatives of different graphs.

As the permutations play a crucial role in the whole algorithm the list of all permutations is stored in an array for fast access in a form that allows to perform these permutations in a convenient way.

The list of all graphs with a given number of points is constructed successively. We start from the list of all graphs with p points and one bond, which contains only one graph. For this graph the list of all permutations, which leave the graph invariant is constructed "by hand." To save some computer memory these lists are stored on disk when they are not needed.

The method to construct the list of all graphs with p points and k bonds (k-list) from the list of all graphs of p points and $(k-1)$ bonds ($(k-1)$-list) together with the list of all permutations which leave each of the graphs in the $(k-1)$-list invariant is the following:

For each graph in the $(k-1)$-list we construct a new graph by adding one bond. This new bond is constructed in the way that it has the highest possible value according to step (ii) of scheme (1). Then we go through a loop of all permutations of the points. We check whether the graph obtained by the permutation is already in the k-list. Simultaneously we exclude from further consideration all additional new graphs, which arise from applying those permutations that leave the original graph in the $(k-1)$-list invariant. If a new graph is found in the new k-list we count how often this graph has been found. This number is the order of the subgroup of all permutations which leaves the graph invariant ($G_{\text{graph}} \subset S_p$). For the problem of finding the partition function described in the Introduction it is necessary to know how many representations of one fixed graph exist which are equivalent with respect to permutations of the points. Let us denote this number by w_{graph}; then the following relation holds.

$$w_{\text{graph}} = \frac{p!}{|G_{\text{graph}}|}. \tag{2}$$

This can be proven by group theoretical arguments. We start from a fixed representation of a graph and generate by means of the permutational group S_p a set of equivalent representations. The first step consists of determining the subgroup G_{graph} of S_p that leaves the fixed representation invariant. Thus a coset decomposition of S_p with respect to G_{graph} uniquely fixes this set of equivalent representations and the order of this set of is the weight w_{graph} of the graph.

If one takes into account that the representing numbers of the graphs found in this way form a monotonic decreasing series the expense of finding a new graph in the k-list can be decreased considerably.

If the whole $(k-1)$-list is exhausted we arrive at a complete k-list of graphs. Increasing k by one until the value $p(p-1)/2$ is reached gives the whole list of graphs for a given number p of points.

3. Evaluation of the List

We used the lists derived in Section 2 to check systematically the tables given in [1]. Apart from the errors given in [3] no further errors were discovered. Let us use the notation of appendix 2 of [1]:

$$\begin{aligned} N_{p,k} &= \text{number of labelled graphs (connected or disconnected)} \\ C_{p,k} &= \text{number of labelled connected graphs} \\ S_{p,k} &= \text{number of labelled stars} \\ \pi_{p,k} &= \text{number of free graphs (connected or disconnected)} \\ \gamma_{p,k} &= \text{number of free connected graphs} \\ \sigma_{p,k} &= \text{number of free stars.} \end{aligned} \tag{3}$$

Here p is the number of points and k is the number of bonds in the graph. A graph is said to be *free* if all point are equivalent, and if all points are distinguished from each other we speak of a *labelled* graph. If the graph can be separated into two or more groups so that there is no line joining either part the graph is said to be *disconnected*; otherwise it is *connected*. If for one graph it not possible to find a point with the property that the graph becomes disconnected if all connections to this point are cut the graph is called a *star*. A polygon is a simple example for a star. It is straightforward from these definitions to determine the numbers in list (3) from the list of graphs. The basis of the method is the notion of the graph matrix m(graph) which is constructed infollowing way.

$$m_{ij}(\text{graph}) = m_{ji}(\text{graph}) = \begin{cases} -1, & \text{if the connection } (i, j) \text{ is present} \\ 0, & \text{if the connection } (i, j) \text{ is not present.} \end{cases} \tag{4}$$

The diagonal elements m_{ii}(graph) are chosen such that the sum in each row vanishes. If one graph consists of n disjoint parts the rank of the graph matrix is $p-n$. All principal minors D(graph) of order $p-1$ are equal and are called *graph complexity* (c.f., Ref. [1]). If this graph complexity is zero the graph is not connected.

Our lists of the graphs have been used to check the values for the graph complexities given in [1] and to extend the tables to graphs with eight points (see Table I).

4. Concluding Remarks

The list of all linear graphs with up to eight points has been generated. As the algorithm to generate the graphs with p points involves, in some way, $p!$ operations to be performed on every graph with $p-1$ points, the expense of computer time

TABLE I

Extension of the Table in [1] to Graphs of 8 Points

(p, k)	$N_{p,k}$	$C_{p,k}$	$S_{p,k}$	$\pi_{p,k}$	$\gamma_{p,k}$	$\sigma_{p,k}$
(8,0)	1	0	0	1	0	0
(8,1)	28	0	0	1	0	0
(8,2)	378	0	0	2	0	0
(8,3)	3276	0	0	5	0	0
(8,4)	20475	0	0	11	0	0
(8,5)	98280	0	0	24	0	0
(8,6)	376740	0	0	56	0	0
(8,7)	1184040	262144	0	115	23	0
(8,8)	3108105	1436568	2520	221	89	1
(8,9)	6906900	4483360	84000	402	236	6
(8,10)	13123110	10230360	835800	663	486	40
(8,11)	21474180	18602136	3940440	980	814	161
(8,12)	30421755	28044072	10908688	1312	1169	429
(8,13)	37442160	35804384	20317528	1557	1454	780
(8,14)	40116600	39183840	28032840	1646	1579	1076
(8,15)	37442160	37007656	30526776	1557	1515	1197
(8,16)	30421755	30258935	27217743	1312	1290	1114
(8,17)	21474180	21426300	20285244	980	970	885
(8,18)	13123110	13112470	12777142	663	658	622
(8,19)	6906900	6905220	6830740	402	400	386
(8,20)	3108105	3107937	3096177	221	220	215
(8,21)	1184040	1184032	1182856	115	114	112
(8,22)	376740	376740	376684	56	56	55
(8,23)	98280	98280	98280	24	24	24
(8,24)	20475	20475	20475	11	11	11
(8,25)	3276	3276	3276	5	5	5
(8,26)	378	378	378	2	2	2
(8,27)	28	28	28	1	1	1
(8,28)	1	1	1	1	1	1

Note. The notation is explained in list (3).

increases drastically with the number of points. For instance, up to seven points it is no real effort for a standard personal computer to generate the list, whereas for eight points a rather powerful workstation had to be used.

ACKNOWLEDGMENTS

This research was supported by the Fonds zur Förderung der wissenschaftlichen Forschung, Grant P6199, and by the Österreichische Nationalbank, Grant 3204.

References

1. G. E. Uhlenbeck and G. W. Ford, "The Theory of Linear Graphs with Applications to the Theory of the Virial Development of the Properties of Gases," *Studies in Statistical Mechanics*, Vol. I, edited by J. de Boer and G. E. Uhlenbeck (North-Holland, Amsterdam, 1962).
2. Ch. Foidl and P. Kasperkovitz, in preparation.
3. W. G. Hoover and A. G. De Rocco, *J. Chem. Phys.* **36**, 3141 (1962).
4. J. S. Rowlinson, *J. Stat. Phys.* **42**, 931 (1986).

Received: March 6, 1989; revised: June 16, 1989

Ch. Foidl
P. Kasperkovitz

Institut für Theoretische Physik
Technische Universität
Wiedner Hauptstrasse 8–10
A-1040 Vienna, Austria