CMST 4(1) 75-78 (1998) DOI:10.12921/cmst.1998.04.01.75-78

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY 4,75-78 (1998)

THE ALGORITHM FOR GENERATION OF CONNECTED CLOSED LINEAR GRAPHS FOR LATTICE-SPIN SYSTEMS

GRZEGORZ MUSIAŁ

Computational Physics Division, Institute of Physics, A. Mickiewicz University
ul. Umultowska 85, 61-614 Poznań, Poland
e-mail: gmusial@pearl. amu. edu.pl

Abstract: The paper presents the algorithm and subroutine **GRGEN** for generation of connected closed linear graphs utilized in perturbation expansions. Starting with polygons and subsequently choosing pairs of not directly connected vertices in already generated graphs, the algorithm consequently projects one vertex onto the other one in the pair, generating only topologically different graphs. This algorithm is independent of the choice of a specific model and lattice geometry. The procedure is written in the FORTRAN 77 language, and is available at the Poznań Supercomputing and Networking Center on the SGI Power Challenge XL supercomputer.

1. INTRODUCTION

Perturbation expansions are widely used in physics and astronomy. For physical phenomena in which interactions between degrees of freedom of the system completely change the character of the solution, it is necessary to derive substantial numbers of terms for such expansions. Classification of the contributions of higher order terms naturally leads to a description in terms of linear graphs[1] (for the review see [1]), in which bonds correspond to interactions. In this work only linear graphs are considered.

One of such phenomena of basic importance is spontaneous magnetization, in modelling of which an important role play the lattice-spin systems. The perturbation expansions have attracted particular attention in statistical mechanics, and especially the low- and high-temperature expansions (L&HTE) have led to important progress in the theory of phase transitions and critical behaviour of these systems. Investigation of complex models of this phenomenon is usually carried out by computer simulation methods of the Monte Carlo type, whose results should be confirmed with the aid of another complementary method. The methods complementary to the Monte Carlo type ones are L&HTE (see e. g. [2] and the papers cited therein).

Application of L&HTE requires the use of all topologically different graphs of i-th order[2], for $i \leq n$, where n denotes the maximum order of expansion after which the series is truncated. Only the closed graphs bring non-zero contribution to the partition function of the lattice-spin systems, and then to the magnetization and to other thermodynamic functions (see e. g. [3]). In addition it is enough to generate only the connected graphs because the non-connected ones are combinations of the former.

[1] A linear graph is a collection of k points (called vertices) with i bonds between certain pairs of vertices and one pair of vertices is connected by one bond

[2] The i-th order graph is the one containing i bonds

In L&HTE graphs are "dressed", i. e. a specific spin-spin interaction together with a coupling constant is assigned to every bond in the graph, which is connected with the choice of the specific model. We will illustrate this "dressing" taking a simple Ising model as an example in which short-range interactions appear only between the nearest neighboring spin variables s, where $s = \pm 1$. Every interaction contributes to the partition function with the factor $e^{Ks_is_j}$ of the product which appears under the sum over all spin configurations (K is the coupling constant). Since $s_is_j = \pm 1$ and $e^{\pm A} = \cosh A(1 \pm \tanh A)$, thus $e^{Ks_is_j} = \cosh A\ (1 + s_i\, s_j \tanh K)$. In "dressing" a graph we assign the factor s_is_j tanh K to the bond connecting the i-th and j-th vertices (for details see [1, 3]). In the Ising model every bond is characterised by the same constant tanh K, because there appears only one kind of spin variables and one kind of interactions. But this problem is much more complex in a more complex model as e. g. Ashkin-Teller model.

After this stage of calculations, certain connected graphs are rejected, particularly in more complicated models. Therefore only after the graphs have been "dressed", all non-connected graphs can be constructed as combinations of connected graphs of lower orders whose sum is the order of the complex graph.

This paper describes the generation of closed connected linear graphs. It is worth noting that the algorithm proposed here is independent of the choice of the model and the lattice geometry, thus the results are of general character.

2. THE ALGORITHM OF CALCULATIONS

We propose here the algorithm **GRGEN,** which generates all closed connected linear graphs of i-th order, where $i \leq n$, and n denotes the maximum order of graphs generated. We perform the generation of graphs starting from polygons which are the simplest graphs and simultaneously they are the starting-point for further calculations. Next graphs are obtained by choosing in the starting graph of all possible pairs of not directly connected vertices and next by consequent projections of the vertex[3] of a higher number (which disappears) onto the other one in the chosen pairs.

An important matter is the way of labelling of graphs. We assign the label to each graph in such a way so that to generate only topologically different graphs. Two graphs which can be put into 1 to 1 correspondence, so that the vertices and bonds correspond, are isomorphic and should get the same label. For the graph composed of k vertices we form the (k x k)-dimensional matrix M of connections of vertices in a given graph: when i-th and j-th vertices are directly connected by the bond then the element $M_{ij}= M_{ji} = 1$ and 0 otherwise, whereas the diagonal elements are zero. The label of the graph is the vector composed of the overdiagonal elements of matrix M read over by rows starting from the left to right and from the top to bottom, but we number vertices in such a way that this vector forms the number of the maximum possible value. This is the unique way of labelling the graphs as a result of which all isomorphic graphs get the same label [4].

[3] Projection of one vertex onto another one physically denotes their overlap

The label of every graph obtained as a result of projection of vertices onto themselves is added at the end of the set of generated graphs provided it has not been generated earlier. The algorithm finishes calculations when all graphs from the set have undergone the projection procedure and from the last one no new graph arose.

It is worth noting that such labelling of graphs allows one to restore the matrix *M* of connections of a graph. Moreover, the labels allows a distinction of all topologically different graphs and they contain the full information about the graphs which they label.

3. DESCRIPTION OF THE GRGEN PROCEDURE

The **GRGEN** procedure generates closed connected linear graphs suitable for the L&HTE-like expansions.

The basic problem which arises at generation of graphs over 11-th order is a great amount of operational memory which is occupied by labels of generated graphs. That is why the subroutine **GRGEN** is written in the language FORTRAN 77, which allows the use of variables of non-standard length of 1 byte.

The call of the procedure:

CALL GRGEN (N, NINIT, FILEOUT, NGR, IERR)

The input parameters:

N - an INTEGER type variable which determines the maximum order of generated graphs.

NINIT - an INTEGER type variable which determines the maximum order of initial polygons, on the basis of which graphs are generated, **NINIT** ≥ **N** + 2 is required. The higher **N** the higher **NINIT** relative to **N.** For the sake of illustration, Table I shows the changes in the maximum order of completed graphs N_{compl} depending on a preselected value of **NINIT.** Taking **N** of above 12, it should be checked that an increase in **NINIT** by 1 does not imply an increase in the number of graphs generated **NGR** (*I*), *I* = **1,2,..., N.**

FILEOUT - a CHARACTER* 12 type variable which contains a file name where to write the labels of the generated graphs. The graphs are ordered by their orders from 3 up to **N.**

Table I. Maximum order of completed graphs N_{compl} in terms of the values of the parameter **NINIT**

NINIT	6	8	9	10	12	14	16	18
N_{compl}	5	6	7	8	9	10	11	12

The output parameters:

NGR - a vector declared as INTEGER NGR (30) where NGR(*I*) contains number of generated graphs of *I*-th order for *I*= 1,2,..., **N.**

IERR - an INTEGER type variable whose value:

0 indicates the successful ran of the procedure **GRGEN**,

1 indicates a data error: **NINIT** > 30 or **NINIT** ≤ **N** + 1,

2 indicates not sufficient memory to store graphs, i. e. **N** is too large.

In the implementation on SGI Power Challenge XL Supercomputer the accessible memory can host labels of all graphs up to **NINIT** $=17$. For **NINIT** >17, after the projection procedure the algorithm removes from the memory all graphs of the order $i >$ **N.** Since these graphs frequently appear once again the calculations, the time of the procedure completion is additionally prolonged.

Acknowledgements
The author wishes to thank Prof. G. Kamieniarz and Prof. J. Rogiers for valuable discussions as well as the Poznań Supercomputing and Networking Center for the access to SGI Power Challenge XL supercomputer. A financial support from the Committee for Scientific Research within the grant 8T1 IF 015 09 is also acknowledged.

References

[1] C. Domb, in *Phase Transitions and Critical Phenomena,* ed. by C. Domb and M. S. Green, Academic, New York 1974, Vol. 3, p. 1

[2] R. V. Ditzian et al., Phys. Rev., B22, 2542 (1980)

[3] J. J. Binney, N. J. Dowrick, A. J. Fisher, M. E. J. Newman, *The theory of critical phenomena,* Clarendon Press, Oxford 1995 (Polish edition by PWN, Warszawa 1998), chapt. 3

[4] J. Nagle, Math. Phys., 7, 1588 (1966)