81-11-161

Ref.TH.3178-CERN

A FAST ALGORITHM FOR MONTE-CARLO SIMULATIONS
OF 4-d LATTICE GAUGE THEORIES WITH FINITE GROUPS

by

Gyan Bhanot

Institute for Advanced Study, Princeton, NJ 08540, USA

Christian B. Lang

CERN, Geneva, Switzerland

and

Claudio Rebbi

Physics Department, Brookhaven National Laboratory, Upton, NY 11973, USA

Ref.TH.3178-CERN
16 October 1981

PROGRAM SUMMARY

Title of program: LATGAUGEMC

Catalogue number:

Program obtainable from: CPC Program Library, Queen's University of Belfast,
N. Ireland (see application form in this issue)

Computer for which the program is designed and others on which it is operable:

Computer: CDC 7600; *Installation:* Brookhaven National Laboratory, Upton, NY, 11973
and CERN, Geneva, Switzerland

The program is operable on any computer which has a word structured memory with a wordlength of at least 28 bits, the possibility to implement mask operations (bit by bit logical product) and circular left-shift, and FORTRAN IV.

Operating system or monitor under which the program is executed: CDC Scope

Programming language used: FORTRAN IV

High speed storage required: 19634 D (46262₈) for the example

No. of bits in a word: 60

Overlay structure: none

No. of magnetic tapes required: none

Other peripherals used: Mass storage (amount depends on the application, 15741D computer words in the example)

No. of cards in combined program and test deck: 393

Card punching code: 029-code

CPC Library subprograms used: none

Keywords: Lattice gauge theory, discrete group, 4-dimensions, variable lattice size, periodic boundary conditions, Monte Carlo, Metropolis method.

Nature of the physical problem

The program LATGAUGEMC generates configurations of the link-gauge variables of a 4-dimensional hyper-cubic lattice of linear extent 2, 4 or 8 with periodic

boundary conditions. The link variables take values from any group where the number of group elements plus the number of elements equal to their own inverse is less than 128. The Monte Carlo method used to update the link variables is the Metropolis methods [1]. The configurations generated by the program are representative of the ensemble of configurations on a finite lattice that defines the partition function of a lattice gauge field theory. They may be used to evaluate expectation values of functions of the gauge variables. The program has been used by the authors [2] to study the 4-dimensional SU(2) lattice gauge theory. It could be used for Monte Carlo simulations of other problems in statistical mechanics and particle physics.

Typical running time

The actual running time depends on the number NUPG of times a single link variable is upgraded before proceeding to the next one. With NUPG = 6, the running time is ≈ 50 μ s/variable per MC iteration on a CDC 7600.

Unusual features of the program

- a) The group elements are labelled so that the operation of finding the inverse of any element is a single FORTRAN command.
- b) The periodic boundary conditions on the lattice are implemented using shift and mask operations to perform the necessary modulo arithmetic.

These features result in a considerable increase in the speed of the Monte Carlo over other conventional methods.

References

- [1] N. Metropolis, A.W. Rosenbluth, A.H. Teller and E. Teller, J. Chem. Phys. 21 (1953) 1087.
- [2] C. Rebbi, Phys. Rev. D 21 (1980) 3350;
D. Petcher and D.H. Weingarten, Phys. Rev. D 22 (1980) 2465;
G. Bhanot and C. Rebbi, Nucl. Phys. B180 [FS2] (1981) 469;
C.B. Lang, C. Rebbi, P. Salomonson and B.S. Skagerstam, Phys. Lett. 101B (1981) 173.

LONG WRITE-UP

1. INTRODUCTION

This paper describes a program to perform Monte Carlo simulations of 4-dimensional lattice gauge theories [1], [2] defined on discrete gauge groups [3]. The number of elements in the gauge group plus the number of elements that are equal to their inverse must not exceed 127. The program as written contains the parameter set for a lattice of 4^4 sites and the 120 element icosahedral subgroup Υ of $SU(2)$. The lattice size can be changed to 2^4 sites or 8^4 sites and the group Υ may be changed to any other group using information supplied in the form of comment statements in the program.

The properties of the gauge group that are used in the program have to be supplied as data in tables MT, ET and IG. Detailed instructions on how to create these tables for an arbitrary gauge group are given in Section 7 of this write-up.

2. THE PHYSICAL MODEL

The lattice gauge theory on a group G is defined by the action [1],

$$S(\{U\}, \beta) = \beta \sum_{(i\mu\nu)} \left[1 - \frac{1}{2N} \operatorname{Re} \{ \operatorname{tr} (V(i\mu\nu) + V^\dagger(i\mu\nu)) \} \right], \quad (2.1a)$$

$$V(i\mu\nu) = U_{i,\mu} U_{i+\mu,\nu} U_{i+\nu,\mu}^{-1} U_{i,\nu}^{-1}. \quad (2.1b)$$

The U 's are the link-gauge variables (represented by complex $N \times N$ matrices). They are elements of G and are defined on the links of a 4-dimensional hypercubic lattice, i labels the lattice sites and $\mu, \nu = 1, 2, 3, 4$ label the four directions. The sum in (2.1) is over all plaquettes (unit squares) of the lattice, each plaquette being considered once in the sum. Each term of the sum represents the product (2.1b) of the link variables around a plaquette $(i\mu\nu)$. The U variables are oriented in the sense that

$$U_{i,\mu}^{-1} = U_{i+\mu,-\mu}. \quad (2.1c)$$

The dynamics is defined by the partition function $Z(\beta)$ and the free energy $F(\beta)$.

$$Z(\beta) = e^{F(\beta)} = \sum_{\{U\}} \exp[-S(\{U\}, \beta)] \quad (2.2)$$

where the sum runs over all configurations of the U variables and the parameter β is the inverse temperature.

The expectation value $\bar{A}(\beta)$ of any function $A(\{U\})$ is given by,

$$\bar{A}(\beta) = \frac{1}{Z(\beta)} \sum_{\{U\}} A(\{U\}) \exp[-S(\{U\}, \beta)] \quad (2.3)$$

The formulation described above was used by Wilson [1] to study the confinement of quarks in strong-interaction particle physics. Since then, the lattice gauge theory has found applications in many diverse areas of particle and statistical physics [2,3].

3. THE METROPOLIS METHOD

Equations (2.1-3) define a problem of statistical mechanics. It is well known that statistical systems can be analysed by computer simulations. This is done by generating a sequence of configurations of the link-gauge variables such that the probability of having in the sequence a configuration with action S is $\exp(-S)$. A widely followed procedure to obtain such a sequence is based on an algorithm, originally introduced by Metropolis et al. [4] which we now describe.

Starting from an arbitrary configuration (with the link-variables assigned any set of values from the group G), one steps through the lattice link by link and updates the old value U of the field on the link to a new value U' using the following algorithm:

a) The contribution $S(U)$ of the link being updated to the total action $S(\{U\}, \beta)$ is calculated. This involves taking the product of U 's on the links bordering the plaquettes which are connected to the link being updated. A trial element U' is constructed by multiplying U with a randomly chosen group element (in our case a neighbour element of the identity). Then $S(U')$, the contribution with U replaced by U' is computed.

- b) If $S(U') \leq S(U)$ then U is replaced by U' .
- c) If $S(U') > S(U)$ a (pseudo) random number R with uniform distribution in $(0,1)$ is generated. If $R < \exp[S(U)-S(U')]$ then U is replaced by U' . Otherwise, the old value U on the link is retained.

While a particular link variable is being updated, all other variables are held fixed. Steps a), b) and c) may be repeated $n_u \geq 1$ times at each link before moving to another variable, provided that the number n_u is kept fixed until all the link variables on the lattice are updated.

One application of the above algorithm to every link of the lattice is called an iteration. It can be shown that the algorithm method described above asymptotically generates configurations with weight $\exp[-S(\{U\}, \beta)]$ and thus can be used for all approximate evaluation of the statistical averages.

4. LABELLING THE GROUP ELEMENTS

Since the group G has a finite number of elements, the possible values that the field U can take may be labelled by integers. It is convenient to set up the labelling as follows:

Suppose G has m elements, of which n are equal to their inverse. We enlarge G to a set G' of $m+n-1$ elements with a definite order. Thus

$$G': \{U_p; p = 1, 2 \dots m+n-1\} \text{ where}$$

- i) the element $U_{(m+n)/2}$ is the identity^{*};
- ii) the elements $U_1, U_2 \dots U_{n-1}$ are the remaining self-inverse elements;
- iii) the set $S_1 = \{U_n, U_{n+1}, U_{n+2} \dots U_{[(m+n)/2]-1}\}$ has elements which have their inverse elements in the set $S_2 = \{U_{[(m+n)/2]+1}, U_{[(m+n)/2]+2} \dots, U_m\}$. The ordering of elements in S_1 and S_2 is such that the inverse of $U_{[(m+n)/2]-i}$ in S_1 is $U_{[(m+n)/2]+i}$ in S_2 . ($i = 1, 2 \dots (m-n)/2$).
- iv) the element U_{m+i} is identical to the element U_{n-i} ($i = 1, 2 \dots (n-1)$).

This labelling assigns one or two integers p ($p = 1, 2 \dots m+n-1$) to each element U_p of the set G' . The advantage of this method of labelling the group elements

* Note that $m+n$ is always even.

is that the inverse of an element $U_i \in G'$ is the element $U_{m+n-i} \in G'$. Thus

$$U_i^{-1} = U_{m+n-i} \quad (i = 1, 2, \dots, (m+n-1)) , \quad (4.1a)$$

or, symbolically

$$(i)^{-1} \equiv (m+n-i) , \quad (4.1b)$$

5. THE PROGRAM

The updating of the lattice (iteration) is performed by calling a subroutine UPDATE (IDIR) which updates all links pointing along IDIR.

We now describe the most relevant features of LATGAUGEMC.

5.1 Locating a link on the lattice

The variables $U_{i,\mu}$ of the theory are defined on links. The location L of each variable is therefore given by five integers, four coordinates x, y, z, t of the site to which the link is connected and the direction μ along which it points.

x, y, z and t are assumed to be integers ranging from 1 to the lattice size. For a lattice of linear extent 2^n , define

$$L = \mu + (x-1) \cdot 2^2 + (y-1) \cdot 2^{n+2} + (z-1) \cdot 2^{2n+2} + (t-1) \cdot 2^{3n+2} . \quad (5.1)$$

Clearly, L ranges from 1 to $4 \cdot (2^n)^4$. This assigns one integer L to each link on the 4-dimensional lattice.

The gauge-variables are stored in an array LV of size $4 \cdot (2^n)^4$. $LV(L)$ contains the integer $p(L)$ labelling the value of the field on the link L (see Section 4 for the labelling convention). Thus,

$$LV(L) = p(L) . \quad (5.2)$$

5.2 Locating neighbouring links with periodic boundary conditions

Given a link L [Eq. (5.1)], the Metropolis method (Section 3) requires that we locate all other links on plaquettes connected to L . This is accomplished in the program by using forward shifts FX, FY, FZ, FT and backward shifts BX, BY, BZ, BT .

These, when appropriately added to L, give the coordinates of the neighbouring links. Periodic boundary conditions are imposed by the use of masks M1, M2 and M3. We shall illustrate the procedure by describing how it works on a lattice of 4^4 sites.

The number S representing the site to which L is connected is $S = L - \mu$.

On a 4^4 lattice, S is a number of up to 10 bits. Counting from right to left, the first two bits contain zero, the next two contain (x-1), the next two (y-1) and so on. Also,

$$S_{\max} = 1111111100 \text{ (binary)}. \quad (5.3)$$

To find the coordinates of a site neighbouring S, we perform the following steps:

Define

$$S' = \text{AND}(S + \text{SHIFT}(S, 15), M3) \quad , \quad (5.4)$$

where

$$M3 = 11001100000001100110000 \text{ (binary)}. \quad (5.5)$$

This operation has the effect of putting zeros in the original location of (x-1) and (z-1) and of shifting these to the bit positions 18, 19 and 22, 23 respectively. (y-1) and (t-1) are in bit positions 5, 6 and 9, 10 respectively. Now define the forward and backward shifts:

$$\begin{array}{ll} FX = 2^{17} & , \quad BX = 3 \cdot 2^{17} \\ FY = 2^4 & , \quad BY = 3 \cdot 2^4 \\ FZ = 2^{21} & , \quad BZ = 3 \cdot 2^{21} \\ FT = 2^8 & , \quad BT = 3 \cdot 2^8 \end{array} \quad (5.6)$$

To shift the site index S forward or backward along a given direction, we add the appropriate shift from (5.6) to S' . Suppose we want to move forward along the x direction and then want the index for the link pointing along μ' from this new site.

Then, define

$$L' = S' + FX + \mu. \quad (5.7)$$

This adds 1 to the location for $(x-1)$ in S' and puts μ' in the bit positions 1, 2,

3. Now we impose the periodic boundary condition using the mask M1:

$$M1 = 11001100000001100110111 \text{ (binary)}. \quad (5.8)$$

The logical operation

$$L'' = \text{AND}(L', M1) \quad (5.9)$$

removes any carry-over bit in the addition of 1 to $x-1$ and thus effectively makes the addition a sum modulo 4, enforcing the periodicity of the lattice.

Finally, we bring L'' into the standard form (Eq. 5.5) by the operation

$$L''' = \text{AND}(\text{SHIFT}(L'', -15) + L'', M2), \quad (5.10)$$

$$\text{with} \quad M2 = 1111111111 \text{ (binary)}. \quad (5.11)$$

L''' is then the index for the link pointing along μ' shifted from L by one site forward along x .

The appropriate shifts and masks for other lattice sites are defined in the program as comments. The subroutine UPDATE upgrades all the link variables in the x direction first, then the directions are permuted in the main program and the link variables with the other orientations are successively upgraded.

5.3 The Monte Carlo method

Once all the link labels for the link to be updated and all links on plaquettes connected to it are recovered, we read the values of the fields on these links from the array LV. Next, we use the multiplication table MT to find $S(U_L)$ and $S(U'_L)$ where U'_L is obtained by multiplying U_L by one of the elements nearest the identity (chosen randomly from the array IG of all such elements). Then, the Metropolis algorithm

(described in Section 3), is used to decide whether or not to change U_L to U'_L . Each link is updated $n_U = [(q+1)/2]$ times before moving to the next link on the lattice.

This updating procedure is performed once on each lattice link to complete one iteration of the lattice. As the lattice links are being updated, we measure the order parameter E defined by,

$$E = \left\langle 1 - \frac{1}{2N} \text{Re}(\text{tr}(U_\square + U_\square^{-1})) \right\rangle = - \frac{\partial F}{\partial \beta} \quad (5.12)$$

where $\square$ stands for a lattice plaquette; E is the average value of the action. E is measured by initializing it to zero before each iteration, incrementing it by $S(U)$ or $S(U')$ (depending on whether or not U was changed to U') as each link is updated, and finally, normalizing it by dividing by $6N_L$ at the end of the iteration.

E can be used as an indicator of the degree to which the lattice has approached equilibrium after a certain number of iterations. For instance, if we initialize the lattice links to the unit element, the initial lattice has $E = 0$. As we perform iterations at a given β , E will change until it stabilizes around some mean value $\bar{E}(\beta)$ characteristic of the temperature $1/\beta$. At this point, the lattice configuration is near thermal equilibrium, at least for the order parameter E . It should be noticed, however, that although the average value of E agrees with the correct thermal average when equilibrium is reached, the quantity E , as defined in the program, does not represent the mean value of the plaquette action of any definite configuration. Thus, in studies where the actual distribution of the action is relevant, such as analyses of fluctuations, E should be evaluated separately, stripping for instance the loop 2 in the subroutine UPDATE.

6. VARIABLES

6.1 List of Variables, their names in LATGAUGEMC and symbols used for them in this write-up

Variable	Definition	Symbol in write-up
NELE	Number of group elements	m
NSCE	Number of elements equal to their inverse	n
NEAR	Number of elements closest to the unit element	q
NLVA	Actual number of links (equal to four times the number of lattice sites) or higher	N_L
E	Average action per plaquette	E
FX,FY,FZ,FT	Forward shifts	FX,FY,FZ,FT
BX,BY,BZ,BT	Backward shifts	BX,BY,BZ,BT
M1,M2,M3	Masks	M1,M2,M3
NUPG	Number of Monte Carlo upgrades performed at a given link before moving to the next	n_u

6.2 Common blocks

The program contains four labelled common blocks.

Common block name and length	Variable list	Definition*)
COM1 Length = 23 words	IS11,IS12,IS13,IS21,IS22,IS23,IS31,IS32,IS33,IS41,IS42,IS43,IS51,IS52,IS53,IS61,IS62,IS63,M1,M2,M3,MI,NUPG	IS11 through IS63 are integer increments that find the coordinates of links on plaquettes connected to the link being updated. M1,M2 and M3 are masks, MI = NELE + NSCE.
COM2 Length = 2+NLVA words	LLEN1,LV(NLVA),E	LLEN1 is the number of lattice sites. LV(I) contains the value of the group element at link location I.
COM3 Length = (NMUL+NELE+NEAR) words	MT(NMUL),ET(NELE),IG(NEAR)	MT is the multiplication table which is used to find the group product of two elements. ET(I) contains the action of the I th group element. IG contains the identifiers of the elements closest to the identity.
COM4 Length = NELE	BFA(NELE)	BFA(I) contains the Boltzman weight $\exp(-\beta \cdot ET(I))$ of the I th group element

*) See Section a for definitions not given below.

7. PROCEDURE TO CONSTRUCT MT, ET AND IG

Storage requirements:

MT : $(2^7+1)(m+n-1)$ words

ET : m words

IG : q words

a) Constructing MT

- i) Construct the m, $N \times N$ matrices of the group G.
- ii) Construct the set G' of $(m+n-1)$ matrices $\{U_p; p = 1, 2 \dots (m+n-1)\}$ ordered as described in Section 5.
- iii) Consider a pair of matrices U_i and $U_j \in G'$. Let $U_i U_j = U_k$ ($k = 1, 2 \dots m$).

Now define,

$$l = i + \text{SHIFT}(j, 7) , \quad (7.1a)$$

and set

$$MT(l) = k . \quad (7.1b)$$

Repeat this for all possible values of i and j ($i, j = 1, 2 \dots m+n-1$) to create the multiplication table.

b) Constructing the energy table ET

For Wilson's definition [1] of the action (Eq. 2.1a) set

$$ET(p) = 1 - \frac{1}{2N} \text{Re}(\text{tr}(U_p + U_{m+p-1})) , \quad (7.2)$$

$$(p = 1, 2 \dots m) .$$

c) Constructing IG, the table of elements nearest the identity

Find the set of integers $\{p_i, i = 1, 2 \dots q\}$ for which $ET(p_i)$ has a minimum non-vanishing value.

$$\text{Set } IG(i) = p_i , \quad (i = 1, 2, \dots q) . \quad (7.3)$$

8. REMARKS

The program makes use of intrinsic functions such as AND or SHIFT which implement simple machine operations but are not standard FORTRAN functions. However, the availability of such functions in a wide range of compilers justifies in our opinion their usage. In any case, the corresponding operations can always be implemented through machine language subroutines or by equivalent arithmetic operations. The functions are defined as follows:

AND(I,J) is the bit by bit logical AND of I with J.

SHIFT(I,N) is the content of I circularly shifted to the left by N bits.

Wherever SHIFT(I,-N) is used, it may be either implemented by a right-shift (circular or with sign extension) or substituted by SHIFT(I,n-N) where n denotes the number of bits of a computer word.

The program utilizes the CDC-inline random number generator function RANF(DUMMY). Any fast generator of random numbers with uniform distribution in (0,1) may be used instead.

REFERENCES

- [1] K.G. Wilson, Phys. Rev. D 10 (1974) 2445.
- [2] R. Balian, J.M. Drouffe and C. Itzykson, Phys. Rev. D 10 (1974) 3376; Phys. Rev. D 11 (1975) 2098; *ibid.* 2104; Phys. Rev. D 19 (1979) 2514.
M. Creutz, L. Jacobs and C. Rebbi, Phys. Rev. Lett. 42 (1979) 1390; Phys. Rev. D 20 (1979) 1915.
M. Creutz, Phys. Rev. Lett. 43 (1979) 553; Phys. Rev. D 21 (1980) 2308; Phys. Rev. Lett. 45 (1980) 331.
G. Toulouse and P. Vannimenus, Phys. Reports 67, No. 1 (1980) 47.
For a more complete discussion and detailed listing of references, see e.g. M. Creutz, Lectures given at the Ettore Majorana School in Erice, BNL preprint 1981, or C. Rebbi, Lectures given at the GIFT School in San Felin de Guixols and at the ICPT, Trieste, ICTP preprint, 1981.
- [3] C. Rebbi, Phys. Rev. D 21 (1980) 3350.
D. Petcher and D.H. Weingarten, Phys. Rev. D 22 (1980) 2465.
G. Bhanot and C. Rebbi, Nucl. Phys. B180 [FS2] (1981) 469.
C.B. Lang, C. Rebbi, P. Salomonson and B.S. Skagerstam, Phys. Lett. 101B, (1981) 173.
- [4] N. Metropolis, A.W. Rosenbluth, A.H. Teller and E. Teller, J. Chem. Phys. 21 (1953) 1087.

APPENDIX A: LISTING OF THE PROGRAM CARD-DECK

```

PROGRAM MAIN(INPUT,OUTPUT,TAPE1)
-----
PROGRAM L A T G A U G E M C FOR MONTE CARLO SIMULATIONS
OF 4-DIMENSIONAL LATTICE GAUGE SYSTEMS WITH A FINITE
GAUGE GROUP.
-----
AUTHORS: G. BHANOT, C. B. LANG, C. REBBI
-----

THE PROGRAM GENERATES CONFIGURATIONS OF THE LINK-GAUGE
VARIABLES OF A 4-DIMENSIONAL HYPER-CUBIC LATTICE OF SIZE
2**4, 4**4, 8**4 TO BE SPECIFIED. THE NUMBER OF ELEMENTS
IN THE GAUGE GROUP PLUS THE NUMBER OF ELEMENTS EQUAL TO
THEIR INVERSE MUST NOT EXCEED 127. APART FROM THIS,
THE GAUGE GROUP MAY HAVE AN ARBITRARY STRUCTURE, WHICH
IS GIVEN IN A MULTIPLICATION TABLE MT. THIS TABLE
TOGETHER WITH AN ACTION TABLE (GIVING THE VALUE OF THE
ACTION FOR EACH GROUP ELEMENT) AND AN ARRAY GIVING THE
INDICES OF THE GROUP ELEMENTS CLOSEST TO THE IDENTITY
HAS TO BE CREATED BEFORE EXECUTION OF THE PROGRAM.
USUALLY THIS IS DONE ONLY ONCE AND THE TABLES ARE
PERMANENTLY KEPT ON MASS STORAGE (TAPE1).
THE MONTE-CARLO METHOD IS THE MODIFIED METROPOLIS METHOD.

THIS PROGRAM HAS THE PARAMETERS SET FOR A 4**4 LATTICE
AND THE 120-ELEMENT ICOSAHEDRAL SUBGROUP OF SU(2).
IN ORDER TO VARY THESE SPECIFICATIONS ALL NECESSARY
CHANGES ARE INDICATED IN COMMENTS. THE FOLLOWING
ABBREVIATIONS ARE USED: *)
    NELE = ACTUAL NUMBER OF GROUP ELEMENTS
    NSCE = NUMBER OF ELEMENTS EQUAL TO THEIR INVERSE
    NEAR = NUMBER OF GROUP ELEMENTS CLOSEST TO THE UNIT
          ELEMENT
    NMUL = MAXIMAL NUMBER OF NECESSARY ENTRIES IN THE
          MULTIPLICATION TABLE.(THIS NUMBER IS EASILY
          OBTAINED BY COMPUTING THE DECIMAL EQUIVALENT
          OF A 14-BIT BINARY NUMBER WHICH IS OBTAINED
          BY WRITING (NELE+NSCE-1) TWO TIMES IN ITS
          BINARY REPRESENTATION.)
    NLVA = ACTUAL NUMBER OF LINKS (FOUR TIMES THE
          NUMBER OF LATTICE POINTS) OR HIGHER.

THE PROGRAM USES THREE NON-STANDARD FUNCTIONS (CF. LONG
WRITEUP):
    RANF(DUMMY) PROVIDES A REAL RANDOM NUMBER UNIFORMLY
                  DISTRIBUTED IN THE INTERVAL (0.,1.),
                  DUMMY IS A DUMMY VARIABLE.
    AND(I,J)     GIVES THE BIT BY BIT LOGICAL PRODUCT OF I
                  WITH J.
    SHIFT(I,N)   GIVES THE CONTENT OF I CIRCULARLY SHIFTED
                  TO THE LEFT BY N BITS. IF N IS NEGATIVE
                  IT DENOTES A SHIFT TO THE RIGHT WHICH
                  MAY OR MAY NOT BE WITH SIGN EXTENSION.
-----
COMMON/COM1/IS11,IS12,IS13,IS21,IS22,IS23,IS31,IS32,IS33,
1      IS41,IS42,IS43,IS51,IS52,IS53,IS61,IS62,IS63,
2      M1,M2,M3,MI,NUPG

```


C		0068
C	FOR DIFFERENT LATTICE SIZE LV(1024) SHOULD BE	0069
C	CHANGED TO LV(NLVA). *	0070
C		0071
C	COMMON/COM2/LLEN1, LV(1024), E	0072
C		0073
C	FOR OTHER GROUPS MT(15609), ET(120), IG(12), AND BFA(120)	0074
C	SHOULD BE CHANGED TO MT(NMUL), ET(NELE), IG(NEAR) AND	0075
C	BFA(NELE). *	0076
C		0077
C	COMMON/COM3/MT(15609), ET(120), IG(12)	0078
C	COMMON/COM4/BFA(120)	0079
C	INTEGER FX, FY, FZ, FT, BX, BY, BZ, BT	0080
C	INTEGER FXA, FYA, FZA, FTA, BXA, BYA, BZA, BTA	0081
C		0082
C	THESE ARE THE INCREMENTS FOR A 4**4 LATTICE:	0083
C		0084
C	DATA FX, FXA, BX, BXA/2*400000B, 2*1400000B/	0085
C	DATA FY, FYA, BY, BYA/2*20B, 2*60B/	0086
C	DATA FZ, FZA, BZ, BZA/2*10000000B, 2*30000000B/	0087
C	DATA FT, FTA, BT, BTA/2*400B, 2*1400B/	0088
C	DATA M1, M2, M3/31401467B, 3777B, 31401460B/	0089
C		0090
C	FOR A 2**4 LATTICE USE INSTEAD THE FOLLOWING STATEMENTS:	0091
C		0092
C	DATA FX, FXA, BX, BXA/4*400000B/	0093
C	DATA FY, FYA, BY, BYA/4*10B/	0094
C	DATA FZ, FZA, BZ, BZA/4*2000000B/	0095
C	DATA FT, FTA, BT, BTA/4*40B/	0096
C	DATA M1, M2, M3/2400057B, 177B, 2400050B/	0097
C		0098
C	FOR A 8**4 LATTICE USE INSTEAD THE FOLLOWING STATEMENTS:	0099
C		0100
C	DATA FX, FXA, BX, BXA/2*400000B, 2*3400000B/	0101
C	DATA FY, FYA, BY, BYA/2*40B, 2*340B/	0102
C	DATA FZ, FZA, BZ, BZA/2*40000000B, 2*340000000B/	0103
C	DATA FT, FTA, BT, BTA/2*4000B, 2*34000B/	0104
C	DATA M1, M2, M3/343434347B, 77777B, 343434340B/	0105
C		0106
C	DATA MX, MXA, MY, MYA, MZ, MZA, MW, MWA/2*1, 2*2, 2*3, 2*4/	0107
C		0108
C	FOR 2**4 : LSIZE=1	0109
C	FOR 4**4 : LSIZE=2	0110
C	FOR 8**4 : LSIZE=3	0111
C		0112
C	LSIZE=2	0113
C		0114
C	FOR ANOTHER GROUP CHANGE 120, 2 AND 12 TO THE APPROPRIATE	0115
C	VALUES OF NELE, NSCE AND NEAR.	0116
C		0117
C	NELE=120	0118
C	NSCE=2	0119
C	NEAR=12	0120
C		0121
C	INITIALIZATION	0122
C		0123
C	ELE=NELE	0124
C	MI=NELE+NSCE	0125
C	MIH=MI/2	0126
C	NUPG=(NEAR+1)/2	0127

```

NSIZE=2*XL SIZE                                0128
LLEN1=NSIZE**4                                0129
LLEN2=4*LLEN1                                0130
PL4=24.*LLEN1                                0131
PRINT 100                                     0132
100  FORMAT(1H1,48HTEST RUN FOR A NON-ABELIAN GAUGE THEORY WITH THE, 0133
1      /,49H ICOSAHEDRAL SUBGROUP OF SU(2) AS GAUGE GROUP      , 0134
2      /,52H -----,/)                                0135
      PRINT 101,NSIZE,NELE                                0136
101  FORMAT(36H PROGRAM FOR LATTICE GAUGE THEORY      , 0137
1      /,52H -----,/,/, 0138
2      16H LATTICE SIZE : ,I1,3H**4,/ , 0139
3      28H DISCRETE GAUGE GROUP WITH ,I3,10H ELEMENTS ,/, 0140
4      38H B DENOTES THE COUPLING PARAMETER BETA,/ , 0141
5      46H E DENOTES THE AVERAGE ACTION PER PLAQUETTE      , 0142
6      //,52H -----,/)                                0143
C                                                    0144
C      READ THE MULTIPLICATION TABLE, ACTION TABLE, AND LIST OF 0145
C      ELEMENTS CLOSEST TO THE UNIT ELEMENTS.            0146
C                                                    0147
      REWIND 1                                           0148
      READ(1) MT,ET,IG                                  0149
C                                                    0150
C      INITIALIZE THE LATTICE:                            0151
C      ISTART=0      ALL ELEMENTS FROZEN TO UNITY        0152
C      ISTART=1      RANDOM START (HOT START)             0153
C                                                    0154
      ISTART=1                                           0155
      DO 1 I=1,LLEN2                                    0156
1      LV(I)=MIH*(1-ISTART)+ISTART*(INT(ELE*RANF(DUMMY))+1) 0157
C                                                    0158
C      START OF B LOOP                                    0159
C                                                    0160
      DO 2 IB=1,100                                     0161
      B=(IB-1)*0.05                                     0162
C                                                    0163
C      COMPUTE ALL BOLTZMANN FACTORS                      0164
C                                                    0165
      DO 3 I=1,NELE                                     0166
3      BFA(I)=EXP(-B*ET(I))                             0167
      E=0.                                               0168
C                                                    0169
C      UPDATE LINKS IN ALL FOUR DIRECTIONS.              0170
C      FOR EACH CALL OF THE SUBROUTINE UPDATE ALL LINKS 0171
C      IN THE CORRESPONDING DIRECTION ARE UPDATED.      0172
C      THE INCREMENTS ARE PERMUTED CORRESPONDINGLY.    0173
C                                                    0174
      DO 4 IDIR=1,4                                     0175
      IS11=MY                                           0176
      IS12=FY+MX                                         0177
      IS13=FX+MY                                         0178
      IS21=BY+MY                                         0179
      IS22=BY+MX                                         0180
      IS23=FX+BY+MY                                       0181
      IS31=MZ                                           0182
      IS32=FZ+MX                                         0183
      IS33=FX+MZ                                         0184
      IS41=BZ+MZ                                         0185
      IS42=BZ+MX                                         0186
      IS43=FX+BZ+MZ                                       0187

```

	IS51=MW	0188
	IS52=FT+MX	0189
	IS53=FX+MW	0190
	IS61=BT+MW	0191
	IS62=BT+MX	0192
	IS63=FX+BT+MW	0193
	CALL UPDATE(IDIR)	0194
	MX=MYA	0195
	FX=FYA	0196
	BX=BYA	0197
	MY=MZA	0198
	FY=FZA	0199
	BY=BZA	0200
	MZ=MWA	0201
	FZ=FTA	0202
	BZ=BTA	0203
	MW=MXA	0204
	FT=FXA	0205
	BT=BXA	0206
	MXA=MX	0207
	FXA=FX	0208
	BXA=BX	0209
	MYA=MY	0210
	FYA=FY	0211
	BYA=BY	0212
	MZA=MZ	0213
	FZA=FZ	0214
	BZA=BZ	0215
	MWA=MW	0216
	FTA=FT	0217
	BTA=BT	0218
4	CONTINUE	0219
C		0220
C		0221
C	DIVIDE THE TOTAL ACTION E BY THE NUMBER OF PLAQUETTES AND	0222
C	BY FOUR (DUE TO THE OVER COUNTING IN THE SUBROUTINE).	0223
C	ATTENTION: THIS E IS THEN THE AVERAGE OVER THE LATTICE	0224
C	WHILE IT IS UPDATED, THUS ALSO AN AVERAGE IN "TIME", AND	0225
C	NOT AN AVERAGE OVER THE ACTUAL LATTICE CONFIGURATION AT	0226
C	THIS MOMENT.	0227
C	CONSIDERING DETERMINATIONS OF FLUCTUATIONS THE USE OF	0228
C	THESE NUMBERS WOULD INDUCE AN ERROR. IN THIS CASE	0229
C	ONE HAS TO REALLY CALCULATE THE AVERAGE OVER THE ACTUAL	0230
C	CONFIGURATION BY E.G. SKIPPING THE UPDATE LOOP 2 IN THE	0231
C	SUBROUTINE UPDATE.	0232
C		0233
	E=E/PL4	0234
	PRINT 102,B,E	0235
102	FORMAT(10H B= ,F10.5,8H E= ,F10.5)	0236
2	CONTINUE	0237
	PRINT 103	0238
103	FORMAT(/,52H -----)	0239
	STOP	0240
	END	

	SUBROUTINE UPDATE(IDIR)	0241
C		0242
C	SUBROUTINE TO UPDATE ALL LINKS VARIABLE IN A DIRECTION.	0243
C		0244
C	FOR CHANGES OF THE COMMONS /COM2/ AND /COM3/ DUE TO OTHER	0245
C	LATTICE SIZE OR ANOTHER GROUP CONSULT THE MAIN PROGRAM.	0246
C		0247
	COMMON/COM1/IS11,IS12,IS13,IS21,IS22,IS23,IS31,IS32,IS33,	0248
1	IS41,IS42,IS43,IS51,IS52,IS53,IS61,IS62,IS63,	0249
2	M1,M2,M3,MI,NUPG	0250
	COMMON/COM2/LLEN1,LV(1024),E	0251
	COMMON/COM3/MT(15609),ET(120),IG(12)	0252
	COMMON/COM4/BFA(120)	0253
C		0254
C	GO THROUGH THE WHOLE LATTICE AND UPDATE THE	0255
C	LINK POINTING FROM THE LATTICE POINT IN THE	0256
C	DIRECTION IDIR.	0257
C		0258
	DO 1 IND1=1,LLEN1	0259
	IND=SHIFT(IND1-1,2)	0260
	N=IND+IDIR	0261
	NA=AND(IND+SHIFT(IND,15),M3)	0262
C		0263
C	PLAQUETTE NUMBER 1	0264
C	COMPUTE ADDRESSES OF THE OTHER THREE LINKS OF THE	0265
C	PLAQUETTE.	0266
C		0267
	N11=AND(NA+IS11,M1)	0268
	N11=AND(N11+SHIFT(N11,-15),M2)	0269
	N12=AND(NA+IS12,M1)	0270
	N12=AND(N12+SHIFT(N12,-15),M2)	0271
	N13=AND(NA+IS13,M1)	0272
	N13=AND(N13+SHIFT(N13,-15),M2)	0273
	L11=LV(N11)	0274
	L12=SHIFT(LV(N12),7)	0275
	L13=SHIFT(MI-LV(N13),7)	0276
	L1=MT(L13+MT(L11+L12))	0277
C		0278
C	NOW L1 CONTAINS THE ELEMENT NUMBER OF THE ORDERED PRODUCT	0279
C	OF THE GROUP ELEMENTS OF LINKS 1,2, AND 3 OF THIS FIRST	0280
C	PLAQUETTE.	0281
C	THE OTHER PLAQUETTES ARE TREATED CORRESPONDINGLY.	0282
C		0283
C	PLAQUETTE NUMBER 2	0284
C		0285
	N21=AND(NA+IS21,M1)	0286
	N21=AND(N21+SHIFT(N21,-15),M2)	0287
	N22=AND(NA+IS22,M1)	0288
	N22=AND(N22+SHIFT(N22,-15),M2)	0289
	N23=AND(NA+IS23,M1)	0290
	N23=AND(N23+SHIFT(N23,-15),M2)	0291
	L21=MI-LV(N21)	0292
	L22=SHIFT(LV(N22),7)	0293
	L23=SHIFT(LV(N23),7)	0294
	L2=MT(L23+MT(L21+L22))	0295
C		0296
C	PLAQUETTE NUMBER 3	0297
C		0298
	N31=AND(NA+IS31,M1)	0299
	N31=AND(N31+SHIFT(N31,-15),M2)	0300

	N32=AND(NA+IS32,M1)	0301
	N32=AND(N32+SHIFT(N32,-15),M2)	0302
	N33=AND(NA+IS33,M1)	0303
	N33=AND(N33+SHIFT(N33,-15),M2)	0304
	L31=LV(N31)	0305
	L32=SHIFT(LV(N32),7)	0306
	L33=SHIFT(MI-LV(N33),7)	0307
	L3=MT(L33+MT(L31+L32))	0308
C		0309
C	PLAQUETTE NUMBER 4	0310
C		0311
	N41=AND(NA+IS41,M1)	0312
	N41=AND(N41+SHIFT(N41,-15),M2)	0313
	N42=AND(NA+IS42,M1)	0314
	N42=AND(N42+SHIFT(N42,-15),M2)	0315
	N43=AND(NA+IS43,M1)	0316
	N43=AND(N43+SHIFT(N43,-15),M2)	0317
	L41=MI-LV(N41)	0318
	L42=SHIFT(LV(N42),7)	0319
	L43=SHIFT(LV(N43),7)	0320
	L4=MT(L43+MT(L41+L42))	0321
C		0322
C	PLAQUETTE NUMBER 5	0323
C		0324
	N51=AND(NA+IS51,M1)	0325
	N51=AND(N51+SHIFT(N51,-15),M2)	0326
	N52=AND(NA+IS52,M1)	0327
	N52=AND(N52+SHIFT(N52,-15),M2)	0328
	N53=AND(NA+IS53,M1)	0329
	N53=AND(N53+SHIFT(N53,-15),M2)	0330
	L51=LV(N51)	0331
	L52=SHIFT(LV(N52),7)	0332
	L53=SHIFT(MI-LV(N53),7)	0333
	L5=MT(L53+MT(L51+L52))	0334
C		0335
C	PLAQUETTE NUMBER 6	0336
C		0337
	N61=AND(NA+IS61,M1)	0338
	N61=AND(N61+SHIFT(N61,-15),M2)	0339
	N62=AND(NA+IS62,M1)	0340
	N62=AND(N62+SHIFT(N62,-15),M2)	0341
	N63=AND(NA+IS63,M1)	0342
	N63=AND(N63+SHIFT(N63,-15),M2)	0343
	L61=MI-LV(N61)	0344
	L62=SHIFT(LV(N62),7)	0345
	L63=SHIFT(LV(N63),7)	0346
	L6=MT(L63+MT(L61+L62))	0347
C		0348
C	OLD LINK ELEMENT:	0349
C		0350
	LO=SHIFT(MI-LV(N),7)	0351
C		0352
C	COMPUTE THE OLD BOLTZMANN FACTOR:	0353
C		0354
	BFO=BFA(MT(LO+L1))*BFA(MT(LO+L2))*BFA(MT(LO+L3))*	0355
1	BFA(MT(LO+L4))*BFA(MT(LO+L5))*BFA(MT(LO+L6))	0356
	DO 2 IU=1,NUPG	0357
C		0358
C	CHOOSE RANDOMLY ONE OF THE NEIGHBOURS OF THE UNIT ELEMENT.	0359
C	LN CONTAINS THE POSSIBLE NEW LINK VARIABLE.	0360

C	IR=IG(1+INT(12*RANF(DUMMY)))	0361
	LN=SHIFT(MT(L0+IR),7)	0362
C		0363
C	COMPUTE THE NEW BOLTZMANNFACTOR:	0364
C		0365
	BFN=BFA(MT(LN+L1))*BFA(MT(LN+L2))*BFA(MT(LN+L3))*	0366
1	BFA(MT(LN+L4))*BFA(MT(LN+L5))*BFA(MT(LN+L6))	0367
C		0368
C	TEST FOR ACCEPTANCE:	0369
C		0370
C	IF(BFO*RANF(DUMMY).GT.BFN)GOTO 2	0371
C		0372
C	ACCEPT THE CHANGE.	0373
C		0374
	L0=LN	0375
	BFO=BFN	0376
2	CONTINUE	0377
C		0378
C	FIX THE LINK VARIABLE.	0379
C		0380
C	LV(N)=MI-SHIFT(L0,-7)	0381
C		0382
C	COMPUTE THE CONTRIBUTION OF THE SIX BORDERING PLAQUETTES	0383
C	TO THE TOTAL ACTION	0384
C		0385
	E=E+ET(MT(L1+L0))+ET(MT(L2+L0))+ET(MT(L3+L0))+	0386
1	ET(MT(L4+L0))+ET(MT(L5+L0))+ET(MT(L6+L0))	0387
		0388
1	CONTINUE	0389
	RETURN	0390
	END	0391

APPENDIX B: OUTPUT OF A TEST RUN

TEST RUN FOR A NON-ABELIAN GAUGE THEORY WITH THE
ICOSAHERAL SUBGROUP OF SU(2) AS GAUGE GROUP

PROGRAM FOR LATTICE GAUGE THEORY

LATTICE SIZE : 4*4
DISCRETE GAUGE GROUP WITH 120 ELEMENTS

B DENOTES THE COUPLING PARAMETER BETA
E DENOTES THE AVERAGE ACTION PER PLAQUETTE

B=	0.00000	E=	1.00154
B=	.05000	E=	.99158
B=	.10000	E=	.97713
B=	.15000	E=	.96684
B=	.20000	E=	.95440
B=	.25000	E=	.93991
B=	.30000	E=	.92763
B=	.35000	E=	.90839
B=	.40000	E=	.89939
B=	.45000	E=	.90119
B=	.50000	E=	.88449
B=	.55000	E=	.86908
B=	.60000	E=	.86234
B=	.65000	E=	.82684
B=	.70000	E=	.81862
B=	.75000	E=	.81227
B=	.80000	E=	.79628
B=	.85000	E=	.79790
B=	.90000	E=	.77754
B=	.95000	E=	.77825
B=	1.00000	E=	.76095
B=	1.05000	E=	.74486
B=	1.10000	E=	.75206
B=	1.15000	E=	.73343
B=	1.20000	E=	.70469
B=	1.25000	E=	.70575
B=	1.30000	E=	.68589
B=	1.35000	E=	.67793
B=	1.40000	E=	.67553
B=	1.45000	E=	.66852
B=	1.50000	E=	.63665
B=	1.55000	E=	.62285
B=	1.60000	E=	.61611
B=	1.65000	E=	.60681
B=	1.70000	E=	.60638
B=	1.75000	E=	.59425
B=	1.80000	E=	.59513
B=	1.85000	E=	.55680
B=	1.90000	E=	.54129
B=	1.95000	E=	.54129
B=	2.00000	E=	.52153
B=	2.05000	E=	.49989
B=	2.10000	E=	.49692
B=	2.15000	E=	.48754
B=	2.20000	E=	.45910

B=	2.25000	E=	.46525
B=	2.30000	E=	.44965
B=	2.35000	E=	.43360
B=	2.40000	E=	.41680
B=	2.45000	E=	.41328
B=	2.50000	E=	.41739
B=	2.55000	E=	.39862
B=	2.60000	E=	.37693
B=	2.65000	E=	.37853
B=	2.70000	E=	.36458
B=	2.75000	E=	.36321
B=	2.80000	E=	.34357
B=	2.85000	E=	.33281
B=	2.90000	E=	.31477
B=	2.95000	E=	.31708
B=	3.00000	E=	.30388
B=	3.05000	E=	.29180
B=	3.10000	E=	.28286
B=	3.15000	E=	.27765
B=	3.20000	E=	.27507
B=	3.25000	E=	.26173
B=	3.30000	E=	.26001
B=	3.35000	E=	.25274
B=	3.40000	E=	.24571
B=	3.45000	E=	.25348
B=	3.50000	E=	.24970
B=	3.55000	E=	.24695
B=	3.60000	E=	.23071
B=	3.65000	E=	.23195
B=	3.70000	E=	.23223
B=	3.75000	E=	.21793
B=	3.80000	E=	.21806
B=	3.85000	E=	.21429
B=	3.90000	E=	.20256
B=	3.95000	E=	.20058
B=	4.00000	E=	.20842
B=	4.05000	E=	.19756
B=	4.10000	E=	.19714
B=	4.15000	E=	.19503
B=	4.20000	E=	.19310
B=	4.25000	E=	.19043
B=	4.30000	E=	.18324
B=	4.35000	E=	.18111
B=	4.40000	E=	.17758
B=	4.45000	E=	.17383
B=	4.50000	E=	.17586
B=	4.55000	E=	.17482
B=	4.60000	E=	.17769
B=	4.65000	E=	.17597
B=	4.70000	E=	.16569
B=	4.75000	E=	.16400
B=	4.80000	E=	.16537
B=	4.85000	E=	.16445
B=	4.90000	E=	.16584
B=	4.95000	E=	.16292