The Ice-type model of Linus Pauling and the three-color problem: an interesting puzzle between Physics, Graph theory, and Computers

Roberto da Silva, Osvaldo S. Nakao, J. R. Drugowich de Felício

1 - Instituto de Física, Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, Brazil
2 - Departamento de Engenharia de Estruturas e Geotécnica, Escola Politécnica da Universidade de São Paulo, São Paulo, São Paulo, Brazil
3 - Departamento de Física, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, São Paulo, Brazil

The ice-type model proposed by Linus Pauling to explain its entropy at low temperatures is here approached in a didactic way. The first theoretical estimate from the model is presented and compared with some results numerically obtained. As follows, we consider the mapping between this model and the three color problem and making use of the transfer-matrix method, all allowed configurations were exactly enumerated for two-dimensional square lattices, with linear size $L$ changing from 2 to 15, where $N = L^2$ is the number of oxygen atoms. Finally, from a linear regression of the numerical results, we obtain an estimate for the case $N \to \infty$ which is compared with the exact solution obtained by Lieb. Moreover we present a brief study of graph theory to support the main results that are here explored.
I. INTRODUCTION

Counting problems can be illustrated by several situations that are part of our daily lives. In general, they appear disguised in the format of probability calculation but it is known that there is no way to estimate probabilities (at least when one considers the Laplace definition) without enumerating all possibilities (the sample space). It is exactly in this stage of calculation that we can understand why the lottery apportionment is frequently transferred for the further weeks. What is never mentioned is that such types of games, instead of challenging gamblers, sometimes resist the cleverness and intelligence of mathematicians, statisticians, and physicists that try to solve them.

This is exactly what happened, for example, with the so-called ice-type model, the subject of this article. It can be resumed as a simple question: How many different ways can hydrogen bridges be arranged if the water is frozen to $T = 0$? More precisely, what is the residual entropy of the ice, i.e. $S = k_B \ln \Omega$, where $k_B$ is the Boltzmann constant and $\Omega = W^N$ is the number of accessible configurations to the system? This is not just a curiosity. This problem appeared in 1933, when Giauque & Ashley [1] measured the entropy of ice at low temperatures and found for the molar entropy the result $s = 0.82 \pm 0.05 \text{ cal/(mol K)}$. It must be observed that molar entropy corresponds to the product of the Avogrado number ($N_0$) by the entropy per site which is equal to $S/N = k_B \frac{1}{N} \ln \Omega = k_B \ln \Omega^{1/N}$ or even $k_B \ln W$.

The first theoretical estimate for the residual entropy of the ice was performed right after, by Linus Pauling, resulting in $s = 0.805 \text{ cal/(mol K)}$. It was published in 1935 [2] and is in good agreement with the experimental value, despite the several approximations performed by the author. In the own words of Lieb [3], this calculation must be considered as one of the more fortunate applications of the Statistical mechanics to real substances. Only in 60’s, Nagle [4] performed more precise numerical estimates for the entropy, obtaining $s = 0.8580 \pm 0.0013 \text{ cal/(mol K)}$ [5], and finally, Lieb [3] obtained an exact solution for the problem in two dimensions ($s = 0.856 \text{ cal/(mol K)}$). We must observe that the searched answer depends on the spatial dimension in which the $\text{H}_2\text{O}$ molecules are inserted, and of the kind of lattice (square, hexagonal, simple cubic) that such molecules are composing. For the sake of simplicity, the calculations were always performed in two dimensions (except by the numerical work done by Nagle) adopting the so called “ice rules” first introduced by Bernal and Fowler [6] in 1933 and enlarged/improved by Pauling.

Following Pauling, let us suppose that the lattice composed by oxygen atoms be square, with hydrogen atoms occupying the bonds between the oxygen atoms. Since in the square lattice there are four bonds between each oxygen atom and its nearest neighbors and how each water molecule
has only two hydrogen atoms, two of them must be in the nearest equilibrium position \((d = 0.95 \, \text{Å})\) and the two other at larger distance \((d = 1.81 \, \text{Å})\), which belong to the neighboring oxygen atoms. In sequence, we come back to this subject to show this corresponds to consider neutral molecules and gave rise to six-vertex model very well known in Statistical Mechanics.

As Lennard [3] observed, this problem is equivalent to discover how many ways there are to properly paint a square map using only 3 colors. A proper coloring of the graph means that two neighbor countries cannot have the same color, i.e., two vertices (or countries) that have one edge in common (If this were to happen, the border between countries would disappear). The two problems are isomorphic, except by a factor 3, as we will show in this paper which is organized as follows: In the next section we introduce the ice-type model and reproduce the calculation presented by Linus Pauling. In section III we show the equivalence between this problem and the colouring map. Previously we performed a revision about graph theory which we consider important to the physicists. Thus, we add more context to the problem and give a brief description of why considering the coloring of large lattices is more complicated. In this same section, we enumerate the acceptable configurations for the small systems and we give directions of the numerical solution of the problem in “brute force” motivating our next section.

In section V we present a original numerical calculation, developed for the colouring version of the problem which allows to proceed to the systems with \(N = 144\) atoms. Moreover, after an extrapolation to the thermodynamic limit \((N \rightarrow \infty)\), our numerical estimate is compared with the results obtained by Lieb. Finally, we present some summaries and conclusions in section VI.

**II. THE ESTIMATE OF LINUS PAULING**

Let us remember that each hydrogen atom can be in two distinct positions, Pauling used an arrow to indicate if it is near (arrow comes in) or far (arrow exits of) each oxygen atom. Moreover the percentage of ions \(H_3O^+\) and \(OH^-\) are considered zero what means that each oxygen atom (site in the lattice) must necessarily have 2 and only 2 hydrogen atoms next to it (two arrows arriving and two arrows departing from site). The consequence is that from 16 possible kind of vertices presented in Fig. 1 only 6 vertices (the first six vertices in this same plot, outlined in blue) satisfy the “ice rules”.

This fact gave the rise to the so called six-vertex model which would be exactly solved during the 60s. Returning to the Pauling’s calculation, once only a fraction \((6/16)\) of the vertices should exist in each site and the number of possible configurations (when one randomly chooses the direction
FIG. 1. The sixteen possibilities of the vertex-types with only the six first vertices satisfying the ice rules, i.e., satisfying the conservation law where two and only two arrows coming in (which means two and only two arrows coming out) are possible (see the allowed vertices outlined in blue).

of the $2N$ arrows starting from the $N$ oxygen atoms) is $2^{2N}$. Thus, considering that the sites are statistically independents (which is not really true) and supposing that in all $N$ vertices the ice rules are satisfied, we have

$$\Omega = W^N$$

$$\cong 2^{2N}(\frac{6}{10})^N$$

(1)
or

\[ W \cong 1.5, \]

which leads to an entropy per mol equal to

\[ s = N_0 k_B \ln W \cong R \ln(1.5) = 0.805 \text{ cal/(mol} \text{ K)}, \]

where \( N_0 \) is the Avogadro number, \( k_B \) is the Boltzmann constant and \( R = N_0 k_B = 1.985 \text{ cal/(mol} \text{ K}) \) is the universal or ideal gas constant.

III. THE LANGUAGE OF COLORS: GRAPHS, CHROMATIC POLYNOMIALS, AND RELATED TOPICS

In this section we will show how the six-vertex model can be mapped on the three color problem. For that, and by the beautiful of the topic, first, we slightly left our focus to show some peculiarities about the coloring of graphs, in order to motivate Physics readers on this topic. After, we will pick up the part of the problem that pointedly deserves our attention and following we definitely show the mapping reported.

Let us obtain some intimacy with the problem of the coloring maps by analysing small “worlds”. If our world was composed for only 4 countries, there would be 81 ways of painting with three colors. This is the result from the operation \( 3^4 \) due to the product \((3.3.3.3)\) since each one of the four countries can be painted (in principle) with any of the three colors\(^1\). However, many ways among these 81 ways to paint our world do not satisfy the condition of the problem: two adjacent countries cannot be painted with the same color. That is the problem. Such condition destroys the independence between the events (This prohibits the multiplication \(3.3.3.3 = 3^4\)) which drastically reduces the number of acceptable paintings. But, in order to obtain the correct number of possibilities, it is necessary to perform a jeweler’s work. There is not a simple logical reasoning that leads to the correct answer.

The difficulty of our problem exactly lies in the fact of events are not independent. As we can see, when one paints a country with a specific color, this color is eliminated of the possibilities to

\(^1\) You should think that the correct was \(4^3\) but it is not true. Think in what would happen at the roll of two dice. The total number of different outcomes in that case is \(6^2 = 36\) and not \(2^6 = 64\). Each of the six faces of the first die can appear together with any of the six faces of the second die. Therefore, one has \(6.6 = 6^2\) possibilities. With three dice, one has \(6.6.6 = 6^3\) possibilities. Thus, the correct is to take as basis the number of possible states for each entity: (dice face, country color, etc.) and as exponent the number of entities (number of dice, number of countries, etc.)
paint a neighbor country. However, that point should not discourage us. At least, in the case of 4 countries, this counting can be performed of a simple way. Before continuing, let us denote colors by numbers. Thus for example, define the colors as yellow (Y), green (G), and red (R). The Fig. 2 shows two ways of painting a world with 4 countries.

The countries are rotulated as \( P_1, P_2, P_3, \) and \( P_4 \), starting from upper left corner and rotating clockwise. In Fig. 2(a) we can see a situation that do not satisfy the constraints of the problem, since it has two neighbor countries painted with same color \((P_3 = P_4)\). However, Fig. 2(b) shows a situation where all constraints are satisfied and all neighbors are painted with distinct colors.

Now we can correctly enumerate the number of distinct paintings for this particular case. Remember that for the fact of working with only three colors, painting one of countries, its neighbor can be colored with one of the two remaining colors. If the yellow is chosen to paint the country \( P_1 \), thus the country \( P_2 \) will only be painted with colors red or green. Moreover, the country \( P_4 \) has to satisfy the same constraint which means that choosing the color of the country \( P_1 \) one has \( 2 + 2 = 4 \) ways to paint the countries \( P_1, P_2, \) and \( P_4 \). Sure, there is still the country \( P_3 \) to paint. Well, we have the following alternatives: Either its two neighbors \((P_2 \) and \( P_4)\) are painted with the same color (both with the color red or both with the color red) or they are painted with different colors. In the first case the country \( P_3 \) can be painted with color Yellow, the same color of \( P_1 \) or with a
remaining color (G or R) and therefore the two first possibilities are multiplied by 2, leading to 4 possible configurations. In the second case ($P_2$ and $P_4$ with different colors) there is no other way to $P_3$: it must be painted with the color Yellow (the same color of $P_1$). Thus, our score results in 6 possible colorings of the world of 4 countries (it is exactly the same number that Pauling found for the number of vertices that satisfies the iced-type rules). However this is not yet the final result to the case of colors. We obtained 6 times by choosing the color Yellow to paint $P_1$. We have more 6 possible colorings starting with Red in the $P_1$ and other 6 more colorings by painting the first country with color Green. Therefore, in a total we have $3 \times 6 = 18$ different ways to coloring a world with 4 countries with 3 colors, satisfying the desired restriction.

But, why this problem is interesting to us? How does it work to larger worlds? Well, here we offer two options to the reader:

1. We recommend the reading of the following subsection if one desires understanding a little bit more about the map coloring and the context of the 3-coloring problem in a more general scenario: the graph theory. Important connections were explored and some little results to be used in the numerical exploration of the problem are developed in this same subsection.

2. We suggest to momentarily avoid the reading of the subsection III A and its possible more technical details and go directly to the subsection III B to the readers that desire a faster reading of this manuscript, since in this current subsection they will find that the 3-coloring problem is exactly the six-vertex problem except by a multiplicative factor, the most important concept to be used until the end of the manuscript

We will use some results of the subsection III A but the reader is able to overcome the missing information or even reading this missing subsection after having a general understanding of the manuscript since this subsection concerns about generality of the problem and places the 3-coloring problem in the context of the graph theory

A. Graph theory and the problem of map coloring

Looking from a graph theory point of view, we can observe maps as a set of countries as vertices, while the borders between the countries representing edges linking these vertices. We can observe a map with four countries and its respective graph in the Figs. 3 (a) and 3 (b).

It is important to mention that the graph represented in Fig. 3 (c) does not correspond to the map observed in Fig. 3 (a) since the country $P_1$ is not a neighbor of $P_4$, and $P_2$ is not a
FIG. 3. (a) Map with four countries. (b) Corresponding graph of map in (a). (c) This graph does not correspond to situation (a) since the country $P_1$ is not a neighbor of $P_4$, and $P_2$ is not a neighbor of $P_3$.

neighbor of $P_4$. Since we studied as properly coloring with three colors a graph/map with four vertices/countries, the Graph theory is much more general and we can extend this for $x$ colors and even for more general graphs, which consists in a interesting and important illustration before to study the desired mapping and continue our main results. In graph theory the number of ways to properly color a graph with $x$ colors is so called the chromatic polynomial of this graph which is here denoted by $\phi(x)$, and you will see that for the graph (b) it is a easy step since you have understood the case of 3 colors previously performed.

First there are only two possibilities: $P_1$ and $P_4$ have the same color, or they have different colors. In the first case, one has $x$ ways to put the same color simultaneously in $P_1$ and $P_4$ in this case you can color $P_2$ with $x - 1$ colors while $P_3$ one also has $x - 1$ ways since they are not neighbors, so one has in this first case
\[ \phi_1(x) = x(x - 1)(x - 1) = x(x - 1)^2 \]

On the other hand (second case) one has \( P_1 \) and \( P_4 \) painted with different colors. The number of ways to perform this task is \( x(x - 1) \). For each way among them one can paint \( P_2 \) with \( x - 2 \) colors and \( P_3 \) also with \( x - 2 \) colors, resulting in this second case a total number of ways calculated by

\[
\phi_2(x) = x(x - 1)(x - 2)(x - 2) = x(x - 1)(x - 2)^2
\]

Thus the total number of coloring the graph \( G \) (b) is

\[
\phi(x) = \phi_1(x) + \phi_2(x)
\]

\[
= x(x - 1)^2 + x(x - 1)(x - 2)^2
\]

\[
= x(x - 1)(x^2 - 3x + 3) \quad (2)
\]

A fast test of this formulae is to perform the particular case \( x = 3 \), and according to our previous calculations we must obtain \( \phi(3) = 18 \), which is exactly the result previously obtained. Please see the graph \( G \) (c). In this case we have all vertices connected to all vertices, graphs that satisfy this condition are known as complete graphs \( K_n \). This particular case is the \( K_4 \) (complete graph with \( n = 4 \) vertices) and it must be observed that only the number of vertices define the graph since they have all possible edges (a total of \( \binom{n}{2} \) edges). In this graph all nodes are connected to all other nodes and a proper coloring of this graph demands \( x \geq n \) colors. Thus, the chromatic polynomial of the graph \( K_4 \) is easily calculated:

\[
\phi_{K_4}(x) = x(x - 1)(x - 2)(x - 3)
\]

and in general case (\( n \) vertices) the fundamental counting principle similarly gives:

\[
\phi_{K_n}(x) = x(x - 1)(x - 2)...(x - n + 1)
\]

Actually, these graphs are much more “sui generis” than we can imagine, since for example for \( n \geq 5 \) they have not a planar representation (or in simple words, a map representation), i.e., we cannot draw a planar representation of these graphs without necessarily having two or more edges
intersecting. Let us better explain this point. For example, one observes that $K_4$ has a notorious planar representation (see Fig. 4 (a)), we are able to draw this graph as a map (in this case with 3 regions) or yet, without the edges intersecting (of course, unless the vertices themselves).

![Fig. 4](image)

FIG. 4. (a) $K_4$ is a planar graph (b) $K_5$ is not a planar graph. (c) $K_{3,3}$ is not also a planar graph. Both $K_5$ and $K_{3,3}$ are the small graphs non planar and represent fundamental structures for the graph theory.

On the other hand, we are not able to draw $K_5$ in a planar way. For example in Fig. 4 (b) we observe two attempts (two isomorphic graphs of $K_5$—i.e., roughly speaking are the same graph $K_5$ drawn in a different way). No one of them leads to a planar representation. The graph $K_5$ is a kind of “minimal non planar graph”. Other similar structure is the graph $K_{3,3}$ (complete bipartite graph on six vertices, three of which connect only to each of the other three). Actually, any non-planar graph cannot have a subgraph which is a subdivision of the $K_5$ or $K_{3,3}$. But why the planarity is an important concept if we are talking about graph colouring? Because a fundamental theorem says that any planar graph can be properly colored with a maximum of 4 colors. The theorem was demonstrated for the first time by 1976 por Kenneth Appel e Wolfgang Haken (see for example [7]), by using an IBM 360, the first accepted proof by using a computer. For example $\phi_{K_5}(x) = x(x - 1)(x - 2)(x - 3)(x - 4)$. If we make $x = 4$, $\phi_{K_5}(4) = 0$ which corroborates the theorem, since $K_5$ is a non-planar graph. On the other hand $\phi_{K_4}(3) = 0$, but $\phi_{K_4}(4) = 24$ ways, which also corroborates the theorem since $K_4$ is a planar graph.
Let us go back to our world with four countries. As we saw, it is more complicated to color this map than the graph $K_4$. We also can observe that a world with four countries is a particular case of coloring a disk of $n$ sectors/countries (Fig. 5(a)), where the Fig. 5(b) is a graph representation of the this world where the countries are placed as disk sectors. This graph is known as a circular disk.

![Fig. 5](image_url)

**FIG. 5.** (a) Circular sector, (b) graph representation of a circular sector (c) A generalization of map with $N = L^2$ countries (a two-dimensional lattice) (d) Graph corresponding a world with 6 countries (e) A particular configuration of colors Y (yellow), G (green), and R (red) to the vertex of the lattice represented in (c) where the blue balls are changed by square cells (countries).

The idea is the same, for example, fixing two countries $P_1$ and $P_3$, or any couple of countries (non-adjacent sectors) separated by only a sector or neighbor a common sector (in this particular choice, $P_2$). In this situation we have two options, that these countries ($P_1$ and $P_3$) can be colored with the same color (situation I) or with two different colors (situation II). Denoting $\phi_n(x)$ the
number of ways to properly color this sector can be described by the recurrence relation

\[ \phi_n(x) = (x - 2)\phi_{n-1}(x) + (x - 1)\phi_{n-2}(x) \]  

(3)

which demands some explanation. In situation I, i.e., \( P_1 \) and \( P_3 \) have the same color works as if these two sectors were merged in a same sector. Thus, for each coloring of the disk of \( n - 2 \) sectors composed by the sector originated from the fusion of the sector \( P_1 \) with sector \( P_3 \) and by other all sectors \( n - 3 \) sectors (except by sector \( P_2 \)), one has \( x - 1 \) ways to paint the sector \( P_2 \) which cannot have the same color of \( P_1 \) neither \( P_3 \). On the other hand (situation II), we have that for each coloring of a disc with \( n - 1 \) sectors composed by all sectors except by the sector \( P_2 \), and for each painting of this disk we have \( x - 2 \) options to the sector \( P_2 \) which necessarily has a color different of the colors attributed to neighboring sectors \( P_1 \) and \( P_3 \), which justify the recurrence relation (3).

An interesting answer to this recurrence relation is \( \varphi_n(x) = \alpha^n \), by direct substitution one has:

\[ \alpha^2 - (x - 2)\alpha - (x - 1) = 0 \]

that has two distinct roots: \( \alpha_1 = p - 1 \) and \( \alpha_2 = -1 \), and a general solution is given by the linear combination: \( \phi_n(x) = A(x - 1)^n + B(-1)^n \). Such equation requires two initial conditions which we know. First a disk with two sectors has \( \phi_2(x) = x(x - 1) \) ways to be colored, since the color attributed to \( P_1 \) necessarily have a color different of \( P_2 \). In a disk with 3 sectors, all of them are neighbors, thus similarly \( \phi_3(x) = x(x - 1)(x - 2) \). So with these two initial conditions we can conclude that \( A = 1 \) and \( B = x - 1 \), which results in

\[ \phi_n(x) = (x - 1)^n + (-1)^n(x - 1) \]  

(4)

It is important to mention that such equation recovers the map with four countries (disk with four sectors) since \( \phi_4(x) = (x - 1)^4 + (-1)^4(x - 1) = x(x - 1)(x^2 - 3x + 3) \), exactly as we obtained in Eq. 2. After this tour across the graph theory and its connection with the colouring of the graphs, let us come back to the colouring of worlds with many countries. We already study the simple case of world with four countries described by Fig. 3(a) and represented by Fig. 3(b). In the case of many countries which is represented by the two-dimensional lattice (Fig. 5(c)), the coloring is not easy. We can start extending the world of four countries to six countries (Fig. 5(d)). An important theorem in graph theory is the deletion-contraction theorem. This theorem says that for example choosing the edge between the countries \( P_1 \) and \( P_6 \) in the original graph \( (G) \), the chromatic polynomial of \( G \) is the polynomial of the graph obtained by deletion of this edge \( (G_1) \) minus the polynomial of the graph obtained by contraction of this edge \( (G_2) \). To calculate
\( \phi_{G_1}(x) \), we can observe that it is obtained multiplying \( \phi_4(x) \) times the ways of properly colouring the vertex \( P_3 \), which occurs in \( x - 1 \) possible ways, times the ways of properly coloring the vertex \( P_6 \) which also occurs in \( x - 1 \) possible ways, so:

\[
\phi_{G_1}(x) = \phi_4(x)(x - 1)(x - 1)
\]

\(
= x(x - 1)^3(x^2 - 3x + 3).
\)

On the other hand, we must observe that the vertex \( P_{3,6} \) has a stronger restriction, it can be colored with colors different of \( P_2 \) and \( P_4 \) that always have different colors, thus

\[
\phi_{G_2}(x) = \phi_4(x)(x - 2)
\]

\(
= x(x - 1)(x - 2)(x^2 - 3x + 3)
\)

So, one has

\[
\phi_G(x) = \phi_{G_1}(x) - \phi_{G_2}(x)
\]

\(
= x(x - 1)^3(x^2 - 3x + 3) - x(x - 1)(x - 2)(x^2 - 3x + 3)
\)

\[
= x(x - 1)(x^2 - 3x + 3)^2
\]

With \( x = 3 \) colors, we obtain \( \phi_G(3) = 54 \) ways. We should naively imagine that recursively an expression for the lattice with \( N = L^2 \) countries should be obtained. But is is not true! Actually we have no an analytical expression for an arbitrary \( N \). In [13] for example it is shown that for \( N \rightarrow \infty \) upper and lower bounds are obtained:

\[
\frac{1}{2}(x - 2 + \sqrt{x^2 - 4x + 8}) \geq \phi(x) \geq \frac{x^2 - 3x + 3}{x - 1}
\]

However for \( x = 3 \), both bounds are the golden ratio \( \frac{1}{2}(1 + \sqrt{5}) \) and \( 3/2 \). However Lieb in a brilliant work has obtained a exact result for \( x = 3 \) at limit \( N \rightarrow \infty \): \( \phi_\infty(3) = (\frac{4}{3})^{3/2} \). But is \( x = 3 \) an important case to us? Absolutely, since we can show that three-coloring problem is exactly the six-vertex problem (ice-type model) except by a multiplicative factor, which is exactly our original problem as we show in the the next subsection.

**B. Mapping the six-vertex problem in the three-coloring problem**

Let us attribute three colors (the same previous colors: Yellow, Green, and Red) to the vertices in the lattice (Fig. 5 (c) ) such that neighboring vertices cannot have the same color, exactly as
countries in a map, by following our convention where vertices (little balls) correspond to cells (countries), as for example we can observe in Fig. 5 (e). Thus, let us establish the cyclical convention of the colors: Y follows G, G follows R, R follows Y (YGRYGRYGR...). From here, every time that, rotating clockwise in relation to a perpendicular axis to the lattice plane, passing by the common point to the neighbor 4 countries (the black points in 5 (e)), and considering all worlds of the 4 countries in a configuration as in Fig. 5 (e), starting for example by convention from $P_1$ in each of this small worlds, if we change from Yellow to Green (or from Green to Red or from Red to Yellow) the arrow in the boundary will be directed to the common point (blue arrow), while the changing from Red to Green (or from Green to Yellow or from Yellow to Red) it will be exiting from the common point (orange arrow), and this for both situations: horizontal and vertical boundaries. This common point will always have two arrows in and two out. It is necessary to clear such point! The Fig. 6 shows the six arrow configuration that can be appear.

![Fig. 6](image)

FIG. 6. Examples of colorings of worlds with four countries translated in the possible 6 configurations of arrows. We start from $P_1$ and we rotate clockwise until $P_1$ again following the cyclical convention established, considering blue (arrow in) and orange (arrow out). Any other configuration of colors is translated in one of these possible 6 configurations of arrows.

Let us consider the first color configuration in this same figure. We start in $P_1$ with Y and when we go to $P_2$ we see G and therefore, following the clockwise orientation, one has a blue vertical
arrow and thus it points out to the black point. However from \( P_2 \) to \( P_3 \), we have G to Y. In this case it follows the counter-clockwise direction (orange horizontal arrow out from black point). Finally from \( P_3 \) to \( P_4 \) (orange vertical arrow out from black point) and from \( P_4 \) to \( P_1 \) (blue horizontal arrow pointing out to the black point). The other colorings in this same figure, translate one of possible arrow configurations. But we cannot forget that each configuration of arrows correspond to three possible colorings since we can choose one of the three possible to start in \( P_1 \).

Thus we can write

\[
\Omega_{\text{colors}}(N) = 3W^N
\]

and therefore, \( W^N = \frac{1}{3}\Omega_{\text{colors}}(N) \), i.e., the factor 1/3, link the number of configurations of a problem in the other problem and must be used to arrive to correct result for the entropy by correcting the calculus of Pauling.

However solving this problem using a computer should be exciting which we can perform this calculation for finite \( N \). In the next section we start this task first using an algorithm which simply calculates \( \Omega_{\text{colors}}(N) \) using brute force (BF).

IV. COMPUTERS AND BRUTE FORCE: PRELIMINARY NUMERICAL RESULTS

A computer can help us to build a table that shows how the number of possibilities \( \Omega_{\text{colors}}(N) \) evolutes when the number of countries enlarges. What we have to do is to generate all possibilities of painting discarding that ones that differentiate the neighbors. We can observe that a program (using fortran 77, since you can compile this program in any Fortran compiler) with a few lines can be written to paint a world with four countries (see Table: I), by using “brute force”, i.e., checking all possible neighbors in a map.

It is important to observe that there is no increase of the difficulty passing from a world of 4 to other with 9. The program will enumerate \( 3^9 = 1.9683 \times 10^4 \) configurations in the case of 9 countries. In this case we can consider two alternatives: Free boundary conditions (FBC) or Periodic boundary conditions (PBC) in one of the directions to perform the counting which is not important when we consider 4 countries, since there is no difference in such case. However for \( N = 9 \) countries this makes difference since, for example \( P_1 \) is neighbor to \( P_3 \), or \( P_2 \) is neighbor to \( P_7 \) with PBC but such neighborhood relations are not considered for FBC, i.e., the red links in Fig. 7 are removed.

See for example how the brute force algorithm for \( N = 9 \) with PBC demands a lot of ”brute
Program: Coloring “Brute Force $N = 4$”

0  Integer $P_1, P_2, P_3, P_4, Icount$
1  $Icount = 0$
2    Do $P_1 = 0, 2$
3      Do $P_2 = 0, 2$
4        Do $P_3 = 0, 2$
5          Do $P_4 = 0, 2$
6            If $(P_1.ne.P_2.$ and. $P_1.ne.P_3)$ then
7              If $(P_4.ne.P_2.$ and. $P_4.ne.P_3)$ then
8                $Icount = Icount + 1$
9              Endif
10            Endif
11          Enddo
12        Enddo
13      Enddo
14    Enddo
15 Write(*,*) Number of configurations $= Icount$
16 Stop
17 End

TABLE I. Algorithm Brute Force

“Brute Force” than for the case $N = 4$.

For $N = 16$, wow! We have $3^{16} = 4.3046721 \times 10^7$ possibilities to select the acceptable colorings and the brute force algorithm becomes a lot quantity of comparisons must be done. How about a world with 25 countries? There’s nothing special either, at least from the point of view of programming logic, but there is a notorious problem of computer/machine limitation. Do you know how many configurations have to be generated? Sure: $3^{25}$, or $8.47288609443 \times 10^{11}$. And how long would it take a personal computer to generate all these configurations? Well, it depends on the spent time to generate each configuration. Actually, we have personal computers very fast and they can execute small operations in thousandths of billionths of a second. Technically, the computational performance is measured in FLOPS (floating-point operations per second) and so actually the computers operates in the scale of GIGAFLOPS and beyond.

Even thinking in GIGAFLOPS, the task is not easy, since it will be necessary at least $\Delta t \simeq 10^{12}.10^{-9} = 10^3$ seconds, or $10^3/(60) \simeq 17$ minutes which is only an rough estimate. So the task gets more and more complicated even in faster computers. And there’s no point in arguing that
**Program:** Coloring “Brute Force N = 9”

0 Integer $P_1, P_2, P_2, P_3, Icount$

1 $Icount = 0$

2 Do $P_1 = 0, 2$

3 Do $P_2 = 0, 2$

4 Do $P_3 = 0, 2$

5 Do $P_4 = 0, 2$

6 Do $P_5 = 0, 2$

7 Do $P_6 = 0, 2$

8 Do $P_7 = 0, 2$

9 Do $P_8 = 0, 2$

10 Do $P_9 = 0, 2$

11 If $(P_1.ne.P_2.and.P_1.ne.P_4)$ then

12 If $(P_2.ne.P_3.and.P_2.ne.P_5)$ then

13 If $(P_4.ne.P_5.and.P_4.ne.P_7)$ then

14 If $(P_5.ne.P_6.and.P_5.ne.P_8)$ then

15 If $(P_7.ne.P_8.and.P_7.ne.P_9)$ then

16 **Inclusion of the conditions for the PBC in one direction:**

17 If $(P_1.ne.P_3.and.P_4.ne.P_6.and.P_7.ne.P_9)$ then

18 $Icount = Icount + 1$

19 Endif

20 Endif

21 Endif

22 Endif

23 Endif

24 Endif

25 Enddo

26 Enddo

27 Enddo

28 Enddo

29 Enddo

30 Enddo

31 Enddo

32 Enddo

33 Enddo

34 Write(*,*) "Number of configurations =", Icount

35 Stop

36 End

---

**TABLE II.** Algorithm Brutal Force for $N=9$
FIG. 7. Configuration with \( N = 9 \) countries. The red lines correspond to neighborhood relations created for periodic boundary conditions. Such connections must be removed for free boundary conditions.

| \( N \) | \( \Omega_{\text{colors}}^{(PBC)} \) | Time | \( \Omega_{\text{colors}}^{(PBC)} \) | Time |
|-------|-----------------|------|-----------------|------|
| 4     | 18              | \(< O(10^{-2})\) | 18              | \(< O(10^{-2})\) |
| 9     | 246             | \(< O(10^{-2})\) | 24              | \(< O(10^{-2})\) |
| 16    | 7812            | \(\approx 0.17 \text{ sec}\) | 4626           | \(\approx 0.18 \text{ sec}\) |
| 25    | 580986          | \(\approx 41 \text{ min}\) | 38880          | \(\approx 39 \text{ min}\) |

TABLE III. Counting by the "brute force" method in faster computers will be possible to advance much more. For example, in order to enumerate the configurations of a map with \( 6 \times 6 = 36 \) countries in the same time that we today perform the computation of the configurations for a map \( 5 \times 5 \) will be necessary to build a computer 170000 faster than these which are considering. Well, but the things are no so bad!

We already have in hands, some exact results from small systems using the “brute force” method (see Table III). In this table, we show the execution of the brute force algorithms using FBC and PBC. We also present the time required for the execution by using a processor Intel i7-8565 U for both situations. Sure, the time depends on a lot of situations and we present the results of one execution only for an idea of the order of magnitude. One can observe that such times are
impracticable since from $N = 16$ to $N = 25$, which are very small systems, the time changes from fraction of seconds approximately to something on the order of 1 hour. Thus, it is interesting to find a numerical alternative for larger systems which permits to perform an extrapolation to the thermodynamic limit: $N \to \infty$. This will be performed in the next section.

V. ELEGANT NUMERICAL RESULTS: THE TRANSFER-MATRIX METHOD

To work with a large number of countries is need to resort an idea very employed in many problems from Statistical Physics: the reduction from a two-dimensional problem to a succession of one-dimensional problems. This is an extremely useful approach when working with problems that have a cylindrical geometry [8] or toroidal, infinite in one of directions.

The first attempts to work with finite systems in the two directions (which is our case) was performed by Binder [9]. However, it was Creswick [10], in 1995, who obtained the best way to apply the techniques to this geometry. In this work, we present an alternative even simpler than the procedure used by these authors, by using intrinsic functions of the programming languages as Fortran and C to pass from one line (of countries) to the following. Moreover, to ensure a faster convergence at limit $N \to \infty$ using periodic boundary conditions one of the directions. Creswick has considered a impossible work to consider both directions. After a hard work, we observed that loops made in pairs take the programming faster, but this procedure leads to a “loss of memory” in one of the directions since the algorithm is greedy discarding the analysed things.

The calculation begins choosing the possible colorings (configurations) for any line of countries. A country can be colored with one of the three colors (0, 1 or 2) and two adjacent countries cannot have the same color (the last country cannot have the color of the first one due to the PBC in the horizontal direction). Remembering that $L$ is the width of the map you can observe that there are

$$\text{possible different colorings} = (3 - 1)^L + (-1)^L(3 - 1) = 2^L + 2(-1)^L$$

possible different colorings to color a line in these conditions, since it corresponds to color a circuit graph with 3 colors as we deduced in the Eq. [4]

We have $L$ loops and $(L/2 + 1)$ decision commands (if’s) to discover the configurations that can be used during the evolution. Following, we move to the second line that also can present only one of these acceptable configurations in the initial step. Among them, we need to discover which are the compatible colorings with each configuration of the first line, i.e., that have no two countries in the same position, colored with the same color. This operation can be performed associating
| $N$ | $\Omega_{\text{colors}}^{(PBC)}$ | Time (secs) |
|-----|-------------------------------|-------------|
| 4   | 18                            | $< O(10^{-2})$ |
| 9   | 24                            | $< O(10^{-2})$ |
| 16  | 4626                          | $< O(10^{-2})$ |
| 25  | 38880                         | $< O(10^{-2})$ |
| 36  | 37284186                      | $< O(10^{-2})$ |
| 49  | 1886476032                    | $< O(10^{-2})$ |
| 64  | 9527634436194                 | $\approx 0.016$ |
| 81  | 28252600002442752             | $\approx 0.047$ |
| 100 | 77048019386428981200          | $\approx 0.14$ |
| 121 | 132046297983569476000000      | $\approx 0.66$ |
| 144 | 196988209730969736000000000000 | $\approx 2.06$ |
| 169 | 19355435196552348800000000000000 | $\approx 12$ |
| 196 | 159147870862109172000000000000000 | $\approx 64$ |
| 225 | 89200916957093512100000000000000000000 | $\approx 333$ |

TABLE IV. Results obtained from transfer matrix method. The advantage of times using this method in comparison with the BF method is notorious and we can run in few minutes colorings with 225 countries. From $N = 100$ the zeros are placed only to complete the power obtained in the numerical result since one has 18 significant digits in double precision.

Each configuration of a line to an integer number, using binary language. This integer has $2L$ bits since each country needs two bits to store its color (00 corresponds to the color 0; 01 corresponds to color 1, and 10 to color 2). Following, we apply the operation exclusive OR or simply XOR (IEOR for Fortran compilers) to the two integers that represent the configurations of the first and of the second line. Since XOR (exclusive OR) works on all bits (see Fig. 8), resulting in 0 if the bits were equal in the same position, and 1 if they are different. A double zero occurs only if one has the same bits occupying the same position in the lines, corresponding to two countries that have the same color.

Thus, $L$ decision commands (if’s) between bits of the integer resulting from operation $\text{IEOR}(\text{Line1},\text{Line2})$ are enough to detect the existence of the adjacent countries colored with the same color. In this case, a configuration of the second line will not be accepted as compatible with the first line. Once it was performed this selection, we go to the comparison between the second and third lines and so on until the last line since the PBC condition is not considered in this direction. This is a important point to make the algorithm faster, since in this way it is greedy, i.e., we discard the
The obtained results with this algorithm can be observed in Table IV. They are exactly equal to those obtained by direct counting (BF) in the simpler cases \(N = 9, 16\) and 25) but extended to \(N = 225\). With these results in hand it is possible to perform an extrapolation to \(N \to \infty\), shown in Fig. 9. According to Eq. 5 it is expect that \(\ln \Omega_{\text{colors}} = \ln 3 + N \ln W\), and a plot of \(\ln \Omega_{\text{colors}}\) as a function of \(N\) suggests a linear behavior with a slope numerically equal to \(\ln W\) which is exactly observed in Fig. 9.

A linear fitting obtained leads to \(W_{\text{num}} = 1.5421 \pm 0.0054\) which must be compared with the result obtained by Lieb in 1967 [3]:

\[
W = (4/3)^{3/2} = 1.5396007.
\]

After this problem explored by Lieb that actually were based in the work of Lee and Yang [11], other works involving six-vertex models were explored in the literature. The novelty about these problems is that vertices are not equally probable (they have no the same weight) since...
FIG. 9. A plot of $\ln \Omega$ obtained by the transfer matrix method as function of $N$. It is expected of the slope of this curve gives and estimate to $\ln W$.

they represent energetically distinct situations. These models present phase transition even in one dimension when one changes the temperature since they obey Mermin-Wagner theorem. But this is a another history for other opportunity!

VI. CONCLUSIONS AND SUMMARIES

The problem of residual entropy of ice was revisited by using a procedure very useful in Statistical mechanics. To employ this method, it was necessary to use a mapping between the ice-type model in the problem of three colors. A brief review about graph theory was revised and we describe important and necessary points to bring context to problem. The results obtained with transfer-matrix method indicates that even working with small systems is possible to obtain a good estimate to thermodynamic limit. The convergence is due to the Periodic boundary conditions in one of directions, according to the ideas of Creswick that were studied, applied in this work, and
even not considering both directions bring a good estimate according to the error bars. The employment of the binary language in the representation of the configurations and the possibility of comparing two configurations, by using only simple operations with bits, is the other point that must be highlighted.

Finally, we call the attention of the reader, mainly that one which is interested in the Combinatorics that the exact result obtained by Lieb, composed by a fraction of integers (4/3) raised to (3/2) is the only exact case and for more colors there is an upper and a lower bound at limit.

**ACKNOWLEDGMENTS**

R. da Silva thanks CNPq for financial support under grant numbers 311236/2018-9, and 424052/2018-0.

[1] W. Giauque, M. Ashley, Phys. Rev. **43**, 81 (1933)
[2] L. Pauling, J. Am. Chem. Soc. **57**, 2680 (1935)
[3] E. H. Lieb, Phys. Rev. Lett. **18**, 692 (1967), Phys. Rev. **162**, 162 (1967)
[4] J. F. Nagle, Journal of Mathematical Physics **7**, 1484 (1966)
[5] It is important to mention that Nagle presented the estimate \( W = 1.540 \pm 0.001 \) in his original work and not exactly an estimate for the entropy of the two-dimensional version of the ice-type model model. We take the liberty to present the estimate for \( s \) based on this value considering that the uncertainty in \( s \) was performed via error propagation: \( \sigma_s = \sqrt{\ln^2 W \cdot \sigma_{k_B}^2 + \frac{k_B^2}{W^2} \sigma_W^2} \approx \frac{k_B}{W} \sigma_W \), once \( \sigma_W >> \sigma_{k_B} \), where \( \sigma_{k_B} \) and \( \sigma_W \) are respectively the uncertainties in \( k_B \) and \( W \), since \( k_B = (1.9872041 \pm 0.0000018) \) cal/mol K.
[6] J.D. Bernal, R. H. Fowler, J. Chem. Phys. **1**, 515 (1933)
[7] K. Appel, W. Haken, The Four-Color Problem. In: Steen L.A. (eds) Mathematics Today Twelve Informal Essays. Springer, New York, NY (1978)
[8] M. N. Barber in Phase Transitions and Critical Phenomena, vol.9, edited by Domb & Lebowitz, Academic Press, New York (1984)
[9] K. Binder, Physica **62**, 508 (1972)
[10] R. J. Creswick, Phys. Rev. E **52**, R5735 (1995)
[11] C. N. Yang, T. D. Lee, Phys. Rev. **150**, 321 (1966)
[12] J. A. Plascak, S. R. A. Salinas, Rev. Bras. de Ensino de Fis. **10**, 173 (1980)
[13] N. Biggs, Bull. London Math. Soc. **9**, 54 (1977)
APPENDIX: TRANSFER MATRIX ALGORITHM FOR L=15 USING PBC IN ONE OF THE DIRECTIONS

1. Main algorithm

```fortran
transfer_matrix_method_L=15_with_PBC.f

1                    2                    3                    4                    5
6: c***************************************************************c
7: c***************************************************************c
8: c***************************************************************c
9: c***************************************************************c
10: c***************************************************************c
11: implicit integer (i-m)
12: implicit real (a-h,o-p)
13: Parameter(L=15,nmax=**L**2*(L-1)**1)
14: dimension itraduz(nmax),ric(nmax),ril(nmax),riv(nmax)
15: c***************************************************************c
16: CALL CPU_TIME(time1)
17: icount = 0
18: Do 12 i1 = 0, 2
19:   Do 14 i2 = 0, 2
20:     Do 16 i3 = 0, 2
21:       Do 18 i4 = 0, 2
22:         Do 20 i5 = 0, 2
23:           Do 22 i6 = 0, 2
24:             Do 24 i7 = 0, 2
25:               Do 26 i8 = 0, 2
26:                 Do 28 i9 = 0, 2
27:                   Do 30 i10 = 0, 2
28:                     Do 32 i11 = 0, 2
29:                       Do 34 i12 = 0, 2
30:                         Do 36 i13 = 0, 2
31:                           Do 38 i14 = 0, 2
32:                             Do 40 i15 = 0, 2
33:                               Do 42 i16 = 0, 2
34:                                 Do 44 i17 = 0, 2
35:                                   Do 46 i18 = 0, 2
36:                                     Do 48 i19 = 0, 2
37:                                       Do 50 i20 = 0, 2
38:                                         Do 52 i21 = 0, 2
39:                                          Do 54 i22 = 0, 2
40:                                            Do 56 i23 = 0, 2
41:                                              Do 58 i24 = 0, 2
42:                                               Do 60 i25 = 0, 2
43:                                                 Do 62 i26 = 0, 2
44:                                                   Do 64 i27 = 0, 2
45:                                                     Do 66 i28 = 0, 2
46:                                                       Do 68 i29 = 0, 2
47:                                                         Do 70 i30 = 0, 2
48:                                                             Do 72 i31 = 0, 2
49:                                                               Do 74 i32 = 0, 2
50:                                                                Do 76 i33 = 0, 2
51: 24                    25                    26                    27                    28
52: 23                    24                    25                    26                    27
53: 22                    23                    24                    25                    26
54: 21                    22                    23                    24                    25
55: 20                    21                    22                    23                    24
56: 19                    20                    21                    22                    23
57: 18                    19                    20                    21                    22
58: 17                    18                    19                    20                    21
59: 16                    17                    18                    19                    20
60: 15                    16                    17                    18                    19
61: 14                    15                    16                    17                    18
62: 13                    14                    15                    16                    17
63: 12                    13                    14                    15                    16
64: 11                    12                    13                    14                    15
65: 10                    11                    12                    13                    14
66:               do itrans=1,L-2
67:                call Transfere(L,RIC,RIL,RIV,nmax,itrados)
68:                do index = 1,nmax
69:                  RIC(index) = RIV(index)
70:                enddo
71:               enddo
72:               call Transfere(L,RIC,RIL,RIV,nmax,itrados)
73:               do iq = 1,nmax
74:                 dsum = dsum + RIV(iq)
75:               enddo
76:               do index = 1,nmax
77:                 write(*,'(1x,13,7x,a,1x,13,7x,a)')index,nmax
78:               enddo
79:               write('*','(1x,13,7x,a,1x,13,7x,a)')time2-time1
80:               pause
81:               STOP
82:               END
```
2. Subroutine

```fortran
transfer_matrix_method_L=15_with_PBC.f
20/05/2020 00:26:58
83: c*************************************************************************
84: c Subroutine Transfer_matrix
85: c*************************************************************************
86: Subroutine Transfer_matrix(L, RIC, RIL, RIV, NMAX, ITRADUZ)
88: implicit integer (i-n)
89: implicit real*8 (a-h,o-z)
90: dimension ITRADUZ(NMAX), RIC(NMAX), RIL(NMAX), RIV(NMAX)
92: do j = 1, NMAX
93:     RIV(j) = 0.0d0
94:     enddo
96: do ind = 1, NMAX
97:     if (RIC(ind).eq.0) goto 250
98:     if (RIL(index).eq.0) goto 240
100:     ires = ITRADUZ(index)
102:     do ibit = 0, 2*L-2, 2
104:         ia = 0
106:         if (btest(ires,ibit)) ia=1
108:         ib=0
110:         if (btest(ires,ib+1)) ib=1
112:         enddo
114:     RIV(ind) = RIV(ind)+RIC(index)
116:     continue
118:     end
```

C*************************************************************************