Extrapolation and the Bulirsch-Stoer algorithm

James L. Monroe
Department of Physics, Penn State University, Beaver Campus, 100 University Drive, Monaca, Pennsylvania 15061-2799
(Received 19 December 2001; published 21 June 2002)

The Bulirsch-Stoer extrapolation algorithm was used in a statistical mechanics setting in 1984 by Henkel and Patkos. Since then it has been used numerous times in a large variety of settings to extrapolate from finite size systems to the infinite system in a large variety of situations in statistical mechanics. We investigate some of its characteristics by using it in situations where the behavior of the infinite system is known. One characteristic is the error involved with the algorithm. More importantly we investigate the dependence of the effectiveness of the algorithm on the size and number of systems used as input and find that a larger number of smaller systems results in better results than a few much larger systems.

DOI: 10.1103/PhysRevE.65.066116 PACS number(s): 05.50.+q, 75.10.−b, 64.60.−i

I. INTRODUCTION

In statistical mechanics one generally is interested in the behavior of a system in the thermodynamic limit, that is when the size of the system is allowed to go to infinity. In many cases of course one cannot find the quantities of interest in this limit so an approach is taken whereby finite systems are studied and then one attempts to extrapolate to the thermodynamic limit. A number of algorithms for doing this have been used and many are very thoroughly reviewed by Guttmann [1]. Just about the time of publication of Guttmann's review article Henkel and Patkos [2] introduced an algorithm, originally due to Bulirsch and Stoer [3] and hereafter to be referred to as the BST algorithm, into the area of critical phenomena. In Guttmann's review article this algorithm was mentioned and described ''as fair to middling'' in comparison to other methods but results using the BST algorithm were not presented in the same manner as the results of seven other algorithms were, because one would suppose, the BST algorithm's almost concurrent introduction with the review. Shortly after the introduction of the algorithm Henkel and Schütz [4] pointed out several characteristics of the algorithm and in particular stressed its being superior to an algorithm due to van den Broeck and Schwartz [5], hereafter to be referred to as the VBS algorithm, which was introduced into statistical mechanics by Hamer and Barber [6] and which was one of the approaches reviewed by Guttmann.

Since the introduction of the BST algorithm there have been a number of topics where use is made of it to take results for a series of finite size systems and extrapolate to the infinite system. Examples of such are quite varied and include the tricritical point and the phase diagram of a collapsing lattice animal [7], layered magnetic systems [8,9], polymers with crossing bonds [10], density profiles, Casimir amplitudes, critical exponents, corner exponents, and the location of the Lee-Yang zeros all for two-dimensional Potts models [11–14], critical temperature, critical exponent and correction to scaling estimates for two- and three-dimensional Ising models [15–18], magnetizations plateaus in antiferromagnetic Heisenberg spin-1/2 ladders [19], and interacting, oriented, self-avoiding walks [20]. While there has been much use of the algorithm there has been little analysis of the characteristics of the algorithm in situations where one knows the dependency of the infinite system on the finite systems and one can thereby check some of the characteristics of the algorithm.

We present such a study of the algorithm in the following. Specifically in the next section we introduce the algorithm and present a number of questions relevant to its use. Following that section we examine the BST algorithm in a setting similar to one of its most common uses listed above that of partition function zeros. However, we investigate its behavior in the extrapolation of results for finite one-dimensional systems to the infinite system. The obvious advantage is that we have numerous analytic results involving various expansions of quantities in terms of the size of the system available to us for the one-dimensional system but not for higher dimensional systems. In Sec. IV we see if some of our findings from Sec. III carry over to the two-dimensional Ising model. Since in some respects the one-dimensional model is rather pathological, e.g., having a phase transition only when the temperature is zero, it is worth seeing if some of our conclusions based on the one-dimensional model are also true for the two-dimensional case. But here again we are in a situation where at least some exact results are available such as the critical temperature which will allow us to see the accuracy and efficiency of the algorithm.

II. BASIC METHOD

In general we wish to determine the value of some quantity we will denote as T of an infinite statistical mechanical system by knowing the corresponding values of T for several finite systems. Since the value of T for the finite systems will depend on the size of the system we denote them as $T(L)$ where L is a measure of the system size. One supposes in general that $T(L)$ can be written as

$$T(L)=T_\infty+a_1L^{-\omega_1}+a_2L^{-\omega_2}+a_3L^{-\omega_3}+\cdots, \qquad (1)$$

where $0<\omega_1<\omega_2<\omega_3<\cdots$ and where T_∞ is the value for the infinite system. The BST algorithm allows

one to start with a finite sequence of values $T(L_1), T(L_2), T(L_3), \ldots, T(L_p)$ and estimate T_∞. In particular the algorithm allows one to construct a table of extrapolants of this sequence, e.g., for $p=5$ we have

$$\begin{array}{ccccccccc} T_0^1 & & & & & & & & \\ & T_1^1 & & & & & & & \\ T_0^2 & & T_2^1 & & & & & & \\ & T_1^2 & & T_3^1 & & & & & \\ T_0^3 & & T_2^3 & & T_4^1 & & & & \\ & T_1^3 & & T_3^2 & & & & & \\ T_0^4 & & T_2^3 & & & & & & \\ & T_1^4 & & & & & & & \\ T_0^5 & & & & & & & & \end{array} \tag{2}$$

where T_4^1 is the algorithm's best estimate for T_∞. The T_q^n are computed from

$$T_{-1}^n = 0, \tag{3}$$

$$T_0^n = T_c(n), \tag{4}$$

$$T_m^n = T_{m-1}^{n+1} + (T_{m-1}^{n+1} - T_{m-1}^n) \times \left[\left(\frac{L_n}{L_{n+m}} \right)^\omega \left(1 - \frac{T_{m-1}^{n+1} - T_{m-1}^n}{T_{m-1}^{n+1} - T_{m-2}^{n+1}} \right) - 1 \right]^{-1}, \tag{5}$$

where $m \geq 1$ and where ω is a free parameter.

Henkel and Schutz [4] in their investigation of the properties of the algorithm generally looked at the comparison between the VBS algorithm [6] and the results it produces compared to the BST algorithm and the results it produces. We concentrate solely on the BST algorithm which based on their results is superior to the VBS algorithm. The most important questions we consider in the following center on the very practical issue of how to get the best results from the algorithm taking into account that computing the input values is generally very difficult and time consuming. One typical question concerns how the accuracy of the estimates given by the BST algorithm depend on the size of the systems used to generate the input, i.e., the $T_0^{n\prime}s$? As an example one might ask if one could compute input data for 13 systems of size $L=3,5,7,\ldots,27$ or only 7 systems but of larger size, say $L=2,8,14,20,\ldots,38$, as we do in the next section, would the estimate based on the input values from the larger systems result in a more accurate estimate or vice versa? Generating data from 13 smaller systems is often easier in general than generating data using 7 larger systems. Second, since when using the algorithm one is constantly subtracting two quantities of near equal value such as $(T_{m-1}^{n+1} - T_{m-1}^n)$ how important is the accuracy of the original input? It is rather apparent that the accuracy is of importance and this is mentioned in Ref. [1] but we give some very clear and specific examples which can be used to give some guidance for the level of accuracy necessary. Finally we want to investigate the ways in which the error involved in a particular BST estimate may be gauged, in particular we investigate a procedure which has been used by several authors for estimating the error. This method relies upon the use of the absolute value of the difference between the two values at the next to last level of approximation, in particular we take as our estimate for error ϵ to be

$$\epsilon = |T_{m-2}^2 - T_{m-2}^1| \tag{6}$$

when there are m input values.

In Ref. [5] a good deal of the work involved test functions of the form of Eq. (1) but with only a term or two on the right-hand side. As they point out this is seldom the case in statistical mechanics applications. We consider Ising model systems which are governed by the Hamiltonian

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i, \tag{7}$$

where the variable σ_i denotes the spin variable on the ith site, each σ can take on the values ± 1, the first sum is over all nearest-neighbor pairs, and the later sum is over all spin variables comprising the system. J is the nearest-neighbor interaction and h is the external magnetic field. The partition function for such a system is

$$\mathcal{Z} = \sum_{\{\sigma\}} \exp[-\beta \mathcal{H}], \tag{8}$$

where $\beta = 1/kT$, and the sum is over all configurations of the system, a configuration being denoted by $\{\sigma\}$. The partition function can be written as a generalized polynomial, an expression where negative exponents are allowed, of z and u where $u \equiv \exp[-4\beta J]$ and $z \equiv \exp[2\beta h]$. One then can find the zeros of the polynomial considering either u or z as the variable. When we wish to consider u as the variable we will always take h to be zero. The zeros in this case are referred to as the Fisher zeros as these were first studied by Michael Fisher [21]. When considering the opposite case, i.e., taking z as the variable, the zeros will be referred to as the Lee-Yang zeros after the two authors whose work started the considerations of the zeros of the partition function [22].

III. ONE-DIMENSIONAL ISING MODEL CASE

In this section we consider all Ising spins to lie along a line and we consider the case where we have periodic boundary conditions. The partition function can be found using the two eigenvalues of the 2×2 transfer matrix for the system. The partition function for a system of N sites is the sum of the two eigenvalues each raised to the Nth power. Setting this sum equal to zero one obtains

$$z + \frac{1}{z} = 2(1-u)\cos\left(\frac{2\gamma - 1}{N} \pi \right) - 2u, \tag{9}$$

TABLE I. Absolute value of the difference between the exact value and the BST algorithm estimate for the argument of θ_1 and estimate of the error based on Eq. (9) when $u=(1/2)^4$, $\omega=2$, and system sizes used are $3,5,\ldots,2m+1$.

	8 figure accuracy input		12 figure accuracy input		16 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
7	2.4×10^{-7}	2.4×10^{-6}	1.6×10^{-7}	4.7×10^{-6}	1.6×10^{-7}	4.7×10^{-6}
8	9.9×10^{-8}	1.4×10^{-7}	5.9×10^{-9}	1.5×10^{-7}	6.0×10^{-9}	1.5×10^{-7}
9	1.9×10^{-7}	1.5×10^{-7}	2.9×10^{-10}	5.4×10^{-9}	1.2×10^{-10}	5.6×10^{-9}
10	7.8×10^{-8}	1.1×10^{-7}	4.1×10^{-10}	6.5×10^{-10}	3.0×10^{-12}	1.2×10^{-10}
11	2.2×10^{-7}	1.7×10^{-7}	9.4×10^{-11}	6.0×10^{-10}	4.5×10^{-14}	2.8×10^{-12}
12	2.0×10^{-6}	1.5×10^{-6}	7.8×10^{-11}	1.7×10^{-10}	3.9×10^{-14}	8.1×10^{-14}
13	2.0×10^{-7}	2.0×10^{-6}	2.6×10^{-11}	1.3×10^{-10}	6.3×10^{-15}	3.9×10^{-14}

where $\gamma=1,2,\ldots,N$. Knowing that the Lee-Yang zeros for the case where $J>0$, which is the only case we consider when dealing with the Lee-Yang zeros, lie on the unit circle in the complex z plane, a result of the Lee-Yang circle theorem [22], then z can be written as $z=\exp[i\theta_\gamma]$ and therefore their location is determined by the value of θ_γ. We will be interested only in the zero closest to the real, positive, z axis. This zero is designated as the leading zero. There are actually two such zeros since the zeros come in complex conjugate pairs. These are the zeros one obtains when $\gamma=1$ or $\gamma=N$. Using the above and taking $\gamma=1$ one can do a series expansion in the variable $(1/N)$ for θ_1 and one obtains

$$\theta_1=\arccos(1-2u)+\frac{\pi^2}{4}\sqrt{\frac{1}{u}-1}\left(\frac{1}{N}\right)^2+\frac{\pi^4}{192}\left(\frac{3}{u}-2\right)\sqrt{\frac{1}{u}-1}\left(\frac{1}{N}\right)^4+O\left(\frac{1}{N}\right)^6 \qquad (10)$$

which is precisely of the form of Eq. (1). Hence we have from a statistical mechanical system an example of exactly the situation that the BST algorithm was designed to handle. A further complication is the variable u which is present in all coefficients on the right-hand side of Eq. (10). As we will see in the following the effectiveness of the BST algorithm is dependent on the value of u. Finally we mention that given the expansion in Eq. (10) we know we want to set $\omega=2$ in the BST algorithm.

Obviously from Eq. (10) in the limit $N\to\infty$ then $\theta_1=\arccos(1-2u)$. We begin our study of the BST algorithm by taking as input a sequence of m values, these values being the value of the argument of θ_1 for systems of size $3,5,\ldots,2m+1$ sites. We then apply the BST algorithm to these m input values obtaining an approximation for the value of the argument of this zero in the infinite site limit. We do this for two different values of u. We are interested in the accuracy of the BST estimates and the easiest way to see this accuracy is to look at the absolute value of the difference between the exact value and the BST estimate. This for various values of m are given in Table I for $u=(1/2)^4$ and in Table II for $u=(1/4)^4$. In both cases we have presented results using 8 figure, 12 figure, and 16 figure accuracy input. Also in these tables we have presented estimates of the error, using Eq. (8), involved in our approximation for again the case of the three varying input accuracies.

First we address the issue of the accuracy of the estimate. In the $u=(1/2)^4$ case even for our smallest sequence of input values, $m=7$, eight figure accuracy is insufficient. Also Table I shows that while for $m=8$ there is little difference between the outcome of using 12 figure as opposed to 16 figure accuracy all longer sequences require 16 figure accuracy. With eight figure accuracy for the input the accuracy of the estimate is basically random. In fact the accuracy obtained with seven input values is virtually the same as that using 13 input values. Table I shows that the most accurate estimate with this input is obtained with $m=10$. However, from Table I one sees very clearly that with 16 figure accuracy for the input values one first of all gains significantly in overall accuracy of the output except when only seven input values are used and that the increase in the number of input

TABLE II. Absolute value of the difference between the exact value and the BST algorithm estimate for the argument of θ_1 and estimate of the error based on Eq. (9) when $u=(1/4)^4$, $\omega=2$, and system sizes used are $3,5,\ldots,2m+1$.

	8 figure accuracy input		12 figure accuracy input		16 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
7	1.9×10^{-3}	3.9×10^{-3}	1.9×10^{-3}	3.9×10^{-3}	1.9×10^{-3}	3.9×10^{-3}
8	5.8×10^{-4}	1.3×10^{-3}	5.8×10^{-4}	1.3×10^{-3}	5.8×10^{-4}	1.3×10^{-3}
9	1.4×10^{-4}	3.9×10^{-4}	1.3×10^{-4}	4.0×10^{-4}	1.3×10^{-4}	4.0×10^{-4}
10	7.7×10^{-6}	1.3×10^{-4}	3.1×10^{-5}	9.7×10^{-5}	3.1×10^{-5}	9.7×10^{-5}
11	2.3×10^{-5}	1.6×10^{-5}	5.7×10^{-6}	2.3×10^{-5}	5.7×10^{-6}	2.3×10^{-5}
12	2.0×10^{-5}	2.6×10^{-6}	1.0×10^{-6}	4.4×10^{-6}	1.0×10^{-6}	4.4×10^{-6}
13	2.5×10^{-5}	4.9×10^{-6}	1.2×10^{-7}	8.6×10^{-7}	1.5×10^{-7}	8.3×10^{-7}

TABLE III. Absolute value of the difference between the exact value and the BST algorithm estimate for the argument of θ_1 and estimate of the error based on Eq. (9) when $u=(1/4)^4$, $\omega=2$, and system sizes used are $3,9,\ldots,6m-3$.

	8 figure accuracy input		12 figure accuracy input		16 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
7	1.1×10^{-5}	1.2×10^{-4}	1.0×10^{-5}	1.1×10^{-4}	1.0×10^{-5}	1.2×10^{-4}
8	1.2×10^{-6}	8.8×10^{-6}	7.8×10^{-7}	9.5×10^{-6}	7.8×10^{-7}	9.5×10^{-6}
9	1.1×10^{-6}	2.2×10^{-6}	3.8×10^{-8}	7.2×10^{-7}	3.8×10^{-8}	7.2×10^{-7}
10	3.3×10^{-7}	1.5×10^{-6}	2.1×10^{-9}	3.6×10^{-8}	1.8×10^{-9}	3.6×10^{-8}
11	1.3×10^{-7}	1.9×10^{-7}	4.8×10^{-10}	2.5×10^{-9}	6.2×10^{-11}	1.8×10^{-9}
12	4.7×10^{-7}	4.1×10^{-7}	4.8×10^{-10}	9.9×10^{-10}	2.0×10^{-12}	6.0×10^{-11}
13	1.9×10^{-7}	6.4×10^{-7}	1.4×10^{-11}	4.5×10^{-10}	1.3×10^{-13}	1.9×10^{-12}

values corresponds to a very systematic increase in the accuracy of the final result of using the BST algorithm. To summarize for eight figure input accuracy one gets an estimate for the argument of the zero closest to the real, positive z axis of six figure accuracy regardless if one uses seven input values or 13 input values but if one goes to 16 figure accuracy input one sees a gradual increase in output accuracy from the BST algorithm from six figures to 14 figures, a pickup of eight orders of magnitude. In addition if one considers 22 figure accuracy of the input (not shown in Table I) then the BST algorithm for 13 input values will further increase the accuracy to 16 figures a pickup of two more orders of magnitude. Further increases in the input accuracy has no effect on the output accuracy as we are now limited by the number of input values, i.e., the size of the systems being used to approximate the infinite system.

The same basic characteristics are illustrated in Table II for the case where $u=(1/4)^4$. Here, however, our results do not have anywhere near the same level of accuracy as those of Table I due to the simple fact that the coefficients in the series expansion in Eq. (10) are larger for this value of u and hence the infinite system result is harder to approximate, i.e., larger systems must be included to obtain the same level of accuracy achieved in Table I. Because of the greater difficulty in the approximation for this value of u the impact of the accuracy of the input is not seen until ten or more input values are used and there is never any substantial difference between the results obtained by 16 figure accuracy input when compared to 12 figure accuracy input. Going to even larger accuracy for the input results in no gain as it is again the size of the systems which are the limiting factor.

As far as the estimation of the error from the results presented in Tables I and II we see with eight figure accuracy input that both for $u=(1/2)^4$ and $u=(1/4)^4$ that only about 1/2 of the time as m runs from 7 to 13 is the estimate of the error large enough. However, for 16 figure accuracy the estimate of the error is always large enough. Therefore given sufficiently accurate input the estimate $\pm$ the error is such that it produces an interval in which the exact value falls and the interval is generally not so large as to imply much less accuracy than is actually achieved.

We now look at the issue of the size of the systems being considered versus the number of systems being consider. Specifically the issue we wish to address can be phrased as: Is it better to obtain m input values from smaller systems or to obtain a smaller number, say $m-p$, input values but for generally larger systems? We can address this issue for our system because we can easily obtain the argument of θ_1 for any system size.

Specifically we will for $u=(1/4)^4$ consider a new series of finite lattice systems given by $3,9,\ldots,6m-3$ where m is then the number of input values for a given sequence of system sizes. We again will consider sequences of from 7 to 13 values hence we will consider systems with as many as 75 sites. The results for this is given in Table III. We again see very clearly the impact of the accuracy of the input values even at the level of eight input values. First for eight figure accuracy input the accuracy of the estimates is almost random as the length of the sequence is increased. For 12 figure accuracy input there is a gain in overall accuracy as well as at least up to $m=12$ sequence a systematic increase in the accuracy of the estimate as the value of m increases. But even 12 figure accuracy is not adequate for 13 input values where one see that going from 12 figure input accuracy to 16 figure input accuracy results in two orders of magnitude more accurate estimates. Furthermore an additional order of magnitude increase in the accuracy of the estimate is achieved for the $m=13$ case if one goes to 22 figure accuracy of the input. Also for the most accurate input the estimate of the error given by Eq. (9) is sufficient to guarantee the estimate $\pm$ the error estimate includes the exact value.

Obviously we also see an increase in the accuracy of the estimates given by the BST algorithm when comparing for any given value of m the estimate from Table II to that of Table III. But to answer the question raised in the earlier paragraph we want to compare estimates from Tables II and III between differing sequence lengths. This can best be done graphically and is done so in Fig. 1. Here we plot as a function of m, the number of input values, the logarithm of the absolute value of the difference between the BST estimate for the argument of θ_1 for the infinite system and the exact value. This is done both for the results from Table II and Table III. From Fig. 1 one sees that the result from Table II involving 13 input values is more accurate than either the results from the $3,9,\ldots,39$ and $3,9,\ldots,45$ site series despite the fact that the largest system involved with the 13 input value sequence of Table II consists of only 27 sites.

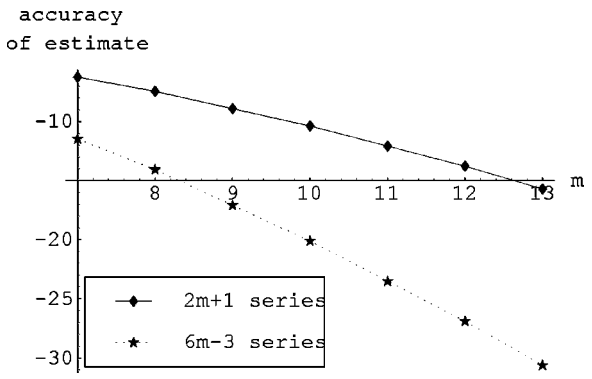

FIG. 1. Log (accuracy of estimate) vs *m* plot for estimates of leading Lee-Yang zero of one-dimensional Ising model.

Even the sequences involving 11 and 12 input values from Table II, which respectively have a largest system size of 23 and 25 sites, produce better estimates than the seven sequence input from Table III which involves three larger systems, i.e., systems of 27, 33, and 39 sites. While in the case of the one-dimensional Ising model one can directly calculate θ_1 for any size system in general applications going to a system with even one more lattice site may double the amount of work necessary to get an input value, e.g., in the direct calculation of a partition function by summing over the states adding a single Ising spin doubles the number of configurations one must sum over.

The above shows the important point that it can be better to consider more smaller system sizes than a few larger system sizes. In the following section we look at the two-dimensional square lattice Ising system to see if this result holds at larger dimensions.

IV. TWO-DIMENSIONAL ISING MODEL CASE

For the two-dimensional Ising model on the square lattice analytic results like those found for the one-dimensional system are unavailable but we do have Onsager's exact result locating the critical temperature and hence where the locus of zeros cross the positive real axis in any appropriate plane involving a complex temperature. Hence we will look only at the Fisher zeros in this case and consider only the case where $h=0$. Several authors, notably Alves *et al.* [15], Creswick [16], and Bhanot [17], have used the BST algorithm on the Fisher zeros of the two-dimensional, square lattice, Ising model to approximate the critical temperature, the critical exponent ν, and corrections to scaling. Specifically we will consider the partition function zeros in the complex *u* plane. We want to look at some of the same aspects of the BST algorithm used on this system as was done in the previous section for the one-dimensional case. In particular we want to contrast the outcome using fewer but larger system sizes with more but smaller systems.

Again the accuracy to which we know the imaginary and real parts of the leading zero will be crucial. The partition function can be written as a polynomial in *u*. Luckily for this model Kaufmann [23] has produced an exact closed form expression allowing one to calculate the partition function for a general $m\times n$ site system with periodic boundary conditions. Furthermore, Beale [24] using Kaufmann's expression has placed on the Internet a MATHEMATICA program which computes the partition function for such a system. Alves *et al.* [15] were aware of this but state that precise computation of the zeros for large systems, e.g., of size 64×64 sites, is nevertheless unfeasible. They point out various numerical problems in determining the zeros of the partition function especially when considering relatively large lattices. They also point out that prior to their paper a number of studies have been presented, two being Refs. [16] and [17], looking at the critical properties of the two-dimensional system using the behavior of the leading zeros but all of them were limited to lattices no larger than 13×13 sites. In order to consider larger systems they develop a method to approximate the leading zero using methods from lattice gauge theory and were able to consider systems as large as 64×64 sites. Their method is iterative in nature and could approximate the zeros to arbitrary accuracy in theory while in actual implementation the time required will certainly limit the computation. Specifically they obtained the real and imaginary parts of the leading zeros to only ten figure accuracy. By ''leading'' zero we mean again as in the one-dimensional case the zero whose argument is closest to zero in value.

For our input data into the BST algorithm we have used Beale's MATHEMATICA program to generate the partition function and then with our own MATHEMATICA programs computed the Fisher zeros for systems up to and including a 30×30 site system. We have used the arbitrary precision allowed by MATHEMATICA and for the computation of the zeros the 30×30 site system have used 400 figure accuracy. For completeness and to allow others to perhaps try other extrapolation algorithms on our data we present in Table IV the leading zeros to 34 figure accuracy for systems of size $4\times 4,\ 5\times 5,\ 6\times 6,\ldots,30\times 30$. It should be pointed out that all our calculations were performed on a personal computer running at 1.7 GHz and for the largest system where we obtained all zeros, the 28×28 system, it took approximately 12 h running MATHEMATICA programs for this systems at 260 figure accuracy. Actually we can go well beyond the 28×28 system size if we are only interested in the leading, Fisher zero and for systems of size 29×29 sites and 30×30 sites this is what was done. The time required to get the leading zero for the 30×30 system requires less than 5 min including generation of the partition function and the memory requirements are negligible. Therefore if needed the computations could certainly be extended to significantly larger systems, however, much of our interest here centers on showing the interplay between the system size and the number of systems used. Once again we will see by using the results of Alves *et al.* [15] who considered systems as large as 64×64 BST estimates based on using smaller systems but more of them are often more accurate than can be obtained using data from the larger systems of reference.

The estimation of u_c, the critical exponent ν, and corrections to scaling are all based on the finite size scaling results

TABLE IV. Real and imaginary part of u_0 system sizes $L\times L$ of the two-dimensional Ising model.

$L\times L$	$\mathrm{Re}(u_0)$	$\mathrm{Im}(u_0)$
6×6	0.17569136155737110163055488623085 96	0.10528348724565994787378078120456 78
7×7	0.17736278577614778737975695893128 78	0.08900625809878365207707381088270372
8×8	0.17808092748072174788641068239952 87	0.07710375571859840823196605520729804
9×9	0.17833701998174593727114246760890 29	0.06801661701229172187622691018925158
10×10	0.17835718536665830913931570255996 33	0.06084948477969045257452302555866051
11×11	0.17825413895935209373015175617117 66	0.05550963710060081067331718039590 75
12×12	0.17808732386962948596369811131205 88	0.05026266795690422238961891903733239
13×13	0.17788934509666346975699116293167 06	0.04624145356375805315485090249218081
14×14	0.17767854492491905486275002775429 51	0.04281649121622217969216131425049106
15×15	0.17746536713158380890298945542622 55	0.03986421637638012018294978025341565
16×16	0.17725574090009677856057278073404 87	0.03729302674155069833522141561729018
17×17	0.17705295348657444913283266115815 94	0.03503356799760033272244700559562696
18×18	0.17685872088459927764955348685674 71	0.03303236187478630182322752204173483
19×19	0.17667381640200396625397114161672 22	0.03124750680583837636622727552824986
20×20	0.17649844762800630038712260687264 72	0.02964570467921901895809107900555564
21×21	0.17633248627574325095258439001237 33	0.02820015958955516896307201692162145
22×22	0.17617561000838819165383887829141 37	0.02688906399403547465887719236387629
23×23	9.17602739059042621376795743406752 51	0.02569448914945364282960995168567808
24×24	0.17588734878335461370421369380136 63	0.02460155919288038906316834895358412
25×25	0.17575498837319274447822287221138 47	0.02359782770153706245275070090846425
26×26	0.17562981697627647208019232823112 23	0.02267280107627923764921341551298777
27×27	0.17551135841421790191370466597932 38	0.02181756992097064653375986537726636
28×28	0.17539915969781059297989600220638 34	0.02102452090076385688342966841851667
29×29	0.17529279456872497955137682578948 71	0.02028710929560394031194939636661839
30×30	0.17519186485865697302359551760490 97	0.01959967783597390122756411622592126

of Itzykson, Pearson, and Zuber [25]. They show for u_0, the leading zero, that

$$u_0(L)=u_c+AL^{-1/\nu}[1+O(L^{-\omega})], \tag{11}$$

where $\omega>0$ and is the correction to scaling exponent. According to Creswick [17] one can write separate expressions for the real and imaginary parts of $u_0(L)$ of the form

$$\mathrm{Re}[u_0(L)]=u_c+BL^{-1/\nu}[1+O(L^{-\omega})] \tag{12}$$

and

$$\mathrm{Im}[u_0(L)]=CL^{-1/\nu}[1+O(L^{-\omega})]. \tag{13}$$

Alternately only can use the absolute value of $u_0(L)$. We have used the real, imaginary, and absolute values of the $u_0(L)$ as input into the BST algorithm and they all give similar results. In the following the specific values were found using the absolute value.

Using as input into the BST algorithm the real part of the ten figure accuracy zeros of Alves *et al.* [15] (actually nine digit accuracy with rounded errors in the tenth digit), using as many as 13 input values based on systems of size 12×12, 15×15, 16×16, 18×18, 20×20, 24×24, 30×30, 32×32, 36×36, 40×40, 48×48, 60×60, and 64×64 sites (the 13 largest systems looked at in Ref. [15]), and with $\omega=1$, we have for $|u_c-u_c^*|$ values of 1.8×10^{-7}, 4.2×10^{-9}, 4.6×10^{-9}, 4.1×10^{-9}, and 8.8×10^{-9} for a sequence of the 9, 10, 11, 12, and 13 systems, respectively. Here u_c^* denotes the BST estimate. Because of the small accuracy in obtaining the values of the leading zero by Alves *et al.* the use of the BST algorithm results in estimates which do not increase in accuracy as the length of the input se-

TABLE V. Absolute value of the difference between the exact value and the BST algorithm estimate of u_c when $\omega=1$, and system sizes used are $12,15,\ldots,3m+9$.

	10 figure accuracy input		16 figure accuracy input		34 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
4	6.8×10^{-6}	6.8×10^{-5}	6.8×10^{-6}	6.8×10^{-5}	6.8×10^{-6}	6.8×10^{-5}
5	2.4×10^{-7}	3.7×10^{-6}	2.5×10^{-7}	3.7×10^{-6}	2.5×10^{-7}	3.7×10^{-6}
6	4.6×10^{-9}	1.4×10^{-7}	6.9×10^{-9}	1.5×10^{-7}	6.9×10^{-9}	1.5×10^{-7}
7	3.3×10^{-7}	8.8×10^{-8}	2.4×10^{-9}	5.7×10^{-9}	2.4×10^{-9}	5.7×10^{-9}

TABLE VI. Absolute value of the difference between the exact value and the BST algorithm estimate of u_c when $\omega=1$, and system sizes used are 12,14, . . . ,$2m+10$.

	10 figure accuracy input		16 figure accuracy input		34 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
5	5.0×10^{-7}	4.4×10^{-6}	4.4×10^{-7}	4.3×10^{-6}	4.4×10^{-7}	4.3×10^{-6}
6	2.4×10^{-7}	3.6×10^{-7}	2.2×10^{-8}	2.2×10^{-7}	2.2×10^{-8}	2.2×10^{-7}
7	1.8×10^{-7}	4.5×10^{-7}	7.6×10^{-9}	1.6×10^{-8}	7.6×10^{-9}	1.6×10^{-8}
8	1.4×10^{-8}	2.9×10^{-8}	6.4×10^{-10}	4.9×10^{-9}	6.4×10^{-10}	4.9×10^{-9}
9	4.3×10^{-8}	1.8×10^{-8}	4.7×10^{-11}	3.0×10^{-10}	4.4×10^{-11}	3.0×10^{-10}
10	1.2×10^{-8}	1.2×10^{-7}	3.3×10^{-11}	8.2×10^{-12}	5.1×10^{-11}	4.6×10^{-12}

quence increases just as we saw with the one-dimensional Lee-Yang zeros.

We now present our own attempts to estimate u_c using the leading Fisher zeros, Eq. (12), and the BST algorithm with $\omega=1$. We use three different sequences of system sizes and vary the accuracy of the input as was done in the one-dimensional case. The sequences of system sizes are $(3m+9)\times(3m+9)$, $(2m+10)\times(2m+10)$, and $(m+11)\times(m+11)$. Again, as in the previous section, m will denote the number of input values. Obviously with $m=1$ all our sequences will begin with the 12×12 site system (the same size system as started the 13 input sequence using the results of Alves *et al.* [15]) and then increase by one, two, or three columns and rows, respectively. Our results for these sequences are presented in Tables V, VI, and VII.

Results for the short sequence of up to seven input values are given in Table V and show that this sequence results in an accuracy comparable to that found using the above larger systems and that as to be expected while the ten figure input accuracy results are somewhat erratic the 16 figure input results in steadily increasing accuracy as the sequence is lengthened and that going beyond 16 figure input accuracy results in no gain. The mid-length sequence with sufficient accuracy for the input gives better results than the ten figure accuracy with system sizes to 64×64 sites with the results based on a nine and ten system sequence but with 16 or better 34 figure input are approximately two orders of magnitude better. Finally for the longer sequence using as many as 17 systems results in no appreciable gain over the mid-length sequence although obviously it is again in general two orders of magnitude better than the sequence using the larger systems of Alves *et al.* [15].

Again the errors as given by Eq. (6) are generally conservative estimates but not quite with the regularity as seen for the one-dimensional case. While with sufficient accuracy of the input we generally have a systematic approach to the correct value given by the BST algorithm we do note that the very systematic approach to the correct value found for the one-dimensional case and clearly illustrated in Fig. 1 is not present, e.g., for our mid-length sequence the value obtained using a ten system sequence is slightly less accurate than that given by a nine system sequence. We suspect that this is due to the added complication that the Fisher zeros for the two-dimensional system with periodic boundary conditions that we are using so as to be able to use the results of Kaufmann and Beale do not fall on the locus of the zeros for the infinite system. This is not true for the zeros in the one-dimensional case where both the Lee-Yang and the Fisher zeros are for any finite system on the same line they would be for the infinite system.

V. CONCLUSIONS

The above two examples show that not very surprisingly the accuracy of the input plays a major role in the accuracy of the BST algorithm and that while not very surprising it may not be as fully appreciated as it needs to be. More importantly the above two examples show that what one can

TABLE VII. Absolute value of the difference between the exact value and the BST algorithm estimate of u_c when $\omega=1$, and system sizes used are 12,13, . . . ,$m+11$.

	10 figure accuracy input		16 figure accuracy input		34 figure accuracy input	
m	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error	Accuracy of est.	Estimate of error
7	6.5×10^{-7}	4.7×10^{-7}	5.5×10^{-9}	2.8×10^{-9}	4.9×10^{-9}	2.6×10^{-9}
8	5.8×10^{-7}	1.9×10^{-7}	5.2×10^{-10}	4.9×10^{-9}	3.7×10^{-10}	4.1×10^{-9}
9	5.6×10^{-7}	9.2×10^{-7}	5.8×10^{-10}	2.3×10^{-11}	3.7×10^{-10}	9.8×10^{-13}
10	9.2×10^{-8}	8.4×10^{-7}	5.2×10^{-10}	8.5×10^{-10}	3.7×10^{-10}	2.0×10^{-10}
11	5.2×10^{-8}	7.7×10^{-8}	8.5×10^{-11}	3.0×10^{-10}	2.0×10^{-11}	5.1×10^{-9}
12	3.5×10^{-7}	1.1×10^{-7}	1.9×10^{-10}	8.7×10^{-11}	2.4×10^{-11}	2.3×10^{-11}
13	7.0×10^{-8}	7.7×10^{-7}	9.2×10^{-11}	3.5×10^{-11}	6.2×10^{-12}	1.5×10^{-11}
14	3.5×10^{-7}	1.0×10^{-7}	2.2×10^{-10}	1.3×10^{-10}	2.5×10^{-11}	3.9×10^{-12}
15	5.8×10^{-8}	3.5×10^{-7}	5.8×10^{-11}	3.9×10^{-12}	2.8×10^{-11}	2.0×10^{-9}

obtain with a slightly larger number of small systems may be greater than what one can get out of a smaller number of larger systems. This is of course of importance due to the fact that in all the cases mentioned in the introduction significantly greater effort may be needed to increase the size of the system even just slightly.

ACKNOWLEDGMENT

We wish to thank Professor Beale for placing on the Internet, so that anyone might use it, his MATHEMATICA program which allows one to calculate the partition function for finite, two-dimensional, Ising systems on the square lattice.

[1] A. J. Guttmann, *Phase Transitions and Critical Phenomena* edited by C. Domb and J. L. Lebowitz (Academic Press, New York, 1989), Vol. 13.
[2] M. Henkel and A. Patkos, J. Phys. A **20**, 2199 (1987).
[3] R. Bulirsch and J. Stoer, Numer. Math. **6**, 413 (1964).
[4] M. Henkel and G. Schutz, J. Phys. A **21**, 2617 (1988).
[5] J. M. van den Broeck and L. W. Schwartz, SIAM (Soc. Ind. Appl. Math.) J. Math. Anal. **10**, 658 (1979).
[6] C. J. Hamer and M. N. Barber, J. Phys. A **14**, 2009 (1981).
[7] R. N. Onody and U. P. C. Neves, J. Stat. Phys. **12**, 91 (1996).
[8] D. Karevski and M. Henkel, Phys. Rev. B **55**, 6429 (1997).
[9] P. E. Berche and B. Berche, Phys. Rev. B **56**, 5276 (1997).
[10] K. D. Machado and J. F. Stilck, J. Phys. A **30**, 1445 (1997).
[11] E. Carlon and F. Igloi, Phys. Rev. B **57**, 7877 (1998).
[12] D. Karevski, P. Lajko, and L. Turban, J. Stat. Phys. **86**, 1153 (1997).
[13] S-Y. Kim and R. J. Creswick, Phys. Rev. Lett. **81**, 2000 (1998).
[14] S-Y. Kim and R. J. Creswick, Phys. Rev. E **58**, 7006 (1998).
[15] N. A. Alves, J. R. Drugowich de Felicio, and U. H. E. Hansmann, Int. J. Mod. Phys. C **8**, 1063 (1997).
[16] N. A. Alves, J. R. Drugowich de Felicio, and U. H. E. Hansmann, J. Phys. A **33**, 7489 (2000).
[17] R. J. Creswick, Phys. Rev. E **52**, R5735 (1995).
[18] G. Bhanot, J. Stat. Phys. **60**, 55 (1990).
[19] A. Trovato and F. Seno, Phys. Rev. E **56**, 131 (1997).
[20] D. C. Cabra, A. Honecker, and P. Pujol, Phys. Rev. B **58**, 6241 (1998).
[21] M. E. Fisher, *Lectures in Theoretical Physics* (University of Colorado Press, Boulder, 1965), Vol. 7C.
[22] C. N. Yang and T. D. Lee, Phys. Rev. **87**, 404 (1952); **82**, 410 (1952).
[23] B. Kaufman, Phys. Rev. **76**, 1232 (1949).
[24] P. D. Beale, Phys. Rev. Lett. **76**, 78 (1996); URL: http://physics.colorado.edu/~beale/IsingExactMathematica.html
[25] C. Itzykson, R. B. Pearson, and J. B. Zuber, Nucl. Phys. **B220**, 415 (1983).