

# EXACT DIAGONALIZATION METHODS FOR QUANTUM SYSTEMS

H.Q. Lin and J.E. Gubernatis

Department Editors: Harvey Gould

*hgould@vax.clark.edu*

Jan Tobochnik

*jant@kzoo.edu*

Exact diagonalization methods are important tools for studying the physical properties of quantum many-body systems. These methods typically are used to determine a few of the lowest eigenvalues and eigenvectors of models of many-body systems on a finite lattice. From these eigenvalues and eigenfunctions, various ground state expectation values and correlation functions are easily computed. Although the methods are limited to small lattice sizes, they have become increasingly popular in the past several years. In addition to providing useful benchmarks for approximate theoretical calculations and quantum Monte Carlo simulations they help to provide insight into the often subtle properties of unsolvable many-body problems in the thermodynamic limit.

In this column, we introduce the Lanczos method<sup>1-3</sup> and a related method, the recursion method.<sup>4,5</sup> To make our discussion concrete, we will use the one-dimensional (1D) Hubbard-Hamiltonian as a working example and provide sufficient detail so that the reader can develop a workable computer code. The ideas and concepts are simple. As we will discuss, the main programming effort is efficiently performing a matrix-vector multiply without storing the matrix. Extensions of the basic ideas and concepts to other many-electron models and to quantum spin models is straightforward.

The 1D Hubbard-Hamiltonian is

$$H = -t \sum_{i,\sigma} (c_{i,\sigma}^\dagger c_{i+1,\sigma} + c_{i+1,\sigma}^\dagger c_{i,\sigma}) + \frac{1}{2} U \sum_{i,\sigma} n_{i,\sigma} n_{i,-\sigma}, \quad (1)$$

where  $c_{i,\sigma}^\dagger$  and  $c_{i,\sigma}$  are the creation and destruction operators for a fermion at site  $i$  with spin  $\sigma$  (up or down) and  $n_{i,\sigma} = c_{i,\sigma}^\dagger c_{i,\sigma}$  is the fermion number operator at site  $i$  for spin  $\sigma$ . The first term in Eq. (1) is the kinetic energy of the electrons and describes their hopping, without spin flip, from site to site. The second term is the potential energy (Coulomb interaction) that exists only if two electrons occupy the same site.

A natural orthonormal basis for the model is the occupation number basis that includes states describing all possible distributions of  $N$  electrons over the  $M$  lattice sites. There are 4 possible electron occupancies at each site (unoccupied, singly occupied for spin up or down, and doubly occupied with one spin up and one spin down). We can represent the up and down spin configurations separately by assigning each lattice site a 1 if the site is occupied or a zero if it is not. An example of a state is this basis that has 5 electrons on an 8 site lattice with 3 up and 2 down spins is

$$|00101010\rangle |00100100\rangle. \quad (2)$$

The up spin electrons are at sites 3, 5, and 7, and the down spin electrons are at sites 3 and 6. The remaining sites are unoccupied.

For  $M$  lattice sites, the number of states in the basis is  $4^M$ , which for  $M = 16$  equals 4 294 967 296. Thus in this basis, the Hamiltonian matrix has  $(4\,294\,967\,296)^2$  elements, i.e., over  $10^{19}$ . Although this matrix is very sparse, the number of nonzero elements is still a very large number, and their storage in a computer's memory is well beyond what is possible. For this reason, the direct numerical solution of eigenvalue problems for quantum many-body systems by conventional dense matrix routines, such as those found in the LAPACK software package,<sup>6</sup> is possible only for lattice sizes up to about 8 sites. Clearly, other methods are needed.

The first step is making the problem more tractable is the use of symmetries to block-diagonalize the Hamiltonian. By similarity transformations, this step produces sequences of smaller matrices along the diagonal. If we want the lowest eigenvalue of the Hamiltonian matrix, i.e., the ground state energy, we simply find the smallest eigenvalue from each of these smaller matrices.

For many-electron models, one of the simplest symmetries is associated with the conservation of the total number of electrons. Using this symmetry, we need only consider the subspace that corresponds to a specified number of electrons for a given number of lattice sites. For a lattice with  $M$  sites and  $N$  electrons, the number of such states is  $(2M)!/N!(2M-N)!$ . For  $M = 16$  and  $N = 16$ , the largest block has 601 080 390 states, and hence the number of elements in this block of the Hamiltonian matrix is  $(601\,080\,390)^2$ , i.e., over  $10^{17}$ .

Hai Qing Lin is a Visiting Assistant Professor at the Department of Physics, University of Illinois, Urbana, IL 61801. James E. Gubernatis is a Staff Member in the Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545.

From the Hubbard-Hamiltonian, Eq. (1), we see that besides the total number of electrons, the number of electrons with a particular spin  $N_\sigma$  also is conserved, because neither term in Eq. (1) changes an up spin to a down spin or vice versa. In condensed matter physics, we typically are interested in systems where  $N_\sigma$  is fixed, so this symmetry is useful and important. Using this symmetry, we produce Hamiltonian blocks of size  $\Pi_\sigma M! / N_\sigma!(M - N_\sigma)!$ . For  $M=16$  with 8 up and 8 down electrons, the largest block size has  $(12\ 870)^2 = 165\ 636\ 900$  states. Thus, the Hamiltonian matrix has approximately  $2 \times 10^{16}$  elements, a number that is still too large for most computers.

Another common symmetry is translational invariance. In one dimension with periodic boundary conditions, two states connected by translational symmetry to the one given in Eq. (2) are

$$|00010101\rangle|00010010\rangle \text{ and } |10001010\rangle|00001001\rangle. \quad (3)$$

When translational symmetry is present, the states can be grouped by wave number  $k$ , since  $k$  is a good quantum number. For the 1D Hubbard model, translational symmetry reduces the block sizes by a factor of  $M$ , i.e., by the number of lattice sites, which happens to equal the number of  $k$  values.

As an example, if translational symmetry, conservation of  $N_\sigma$ , and the symmetry group of a square are used on a  $4 \times 4$  lattice with 16 electrons (8 up and 8 down), the dimension of the biggest block of the Hamiltonian matrix is found to be 1 310 242. This number is somewhat larger than what we might expect from simple considerations. *A priori*, we expect translational symmetry to reduce the size of the matrix by a factor of 16, and the point group of the square to reduce the size by a factor of 8, that is, we expect that these two symmetries together to reduce the matrix by a factor of  $16 \times 8 = 128$ . However,  $165\ 636\ 900 / 128 \approx 1\ 294\ 038$  is smaller than what we actually obtain because not all states are affected by both symmetries. Although the actual size of 1 310 242 is much smaller than  $4^{16}$ ,  $(1\ 310\ 242)^2$  matrix elements still are too many to store. We note, however, that storing a few vectors of length 1 310 242 is well within the capability of today's largest computers.

Other useful symmetries include particle-hole symmetry and charge conjugation. The use of symmetries, however, can be overdone as a point is easily reached where little is gained in terms of computer memory reduction and much is lost by now having a computer code "hard-wired" for a particular problem. Often, the conservation of  $N_\sigma$  is all that is used.

One might ask, "Why not store the matrix on a disk and read parts of it back as needed?" The difficulty is that even on computers with fast disks, the I/O speed still is too slow compared to the computational speed. To avoid this bottleneck, very fast algorithms for computing matrix elements "on-the-fly" have been developed. These algorithms are highly suitable for vector and parallel computers<sup>7</sup> and make it unnecessary to store the elements.

To see how these algorithms work, we first address the question of how do we represent all the possible configurations on a computer. We observe that each of the possible  $4^M$  states maps uniquely onto an integer  $I$  defined by

$$I = \sum_{i=1}^M [n_u(i)2^{i-1} + n_d(i)2^{M+i-1}], \quad (4)$$

where  $n_u(i)$  and  $n_d(i)$  are the occupancies of site  $i$  for the up and the down spins. The bits of this integer represent a specific state

$$|n_u(1), n_u(2), \dots, n_u(M)\rangle |n_d(1), n_d(2), \dots, n_d(M)\rangle. \quad (5)$$

Because we are interested in only those states that correspond to  $N$  electrons, we need to set up a one-to-one correspondence between  $I$  and a label  $J$  that runs from 1 to the number of states with  $N$  electrons. With such a label, a table  $T$ , defined by  $T(J) = I$ , compactly expresses the correspondence.

The two-table method of Lin<sup>8</sup> is an efficient and convenient way of establishing this correspondence. In this method, the states of the system are split into two pieces, which, for example, may represent the left- and right-halves of a 1D chain or the "red" and "black" sites of a two-dimensional (2D) checkerboard square lattice. For the Hubbard model, choosing the two pieces to correspond to the up and down spins is especially convenient. For a given value of  $I$ , we can write

$$I = I_1 + 2^M I_2, \quad (6)$$

so that the bits of the integers  $I_1$  and  $I_2$  represent the occupancy of the sites for up and down spins. We next define two arrays (tables)  $J_1$  and  $J_2$ , so that

$$J = J_1(I_1) + J_2(I_2). \quad (7)$$

Now, if we know  $J$  and properly define  $J_1$ , then  $J_2$  is fixed, and we have established a one-to-one correspondence between  $I$  and  $J$ . Before we define the array  $J_1$ , we remark that although each state  $I$  is occupied by  $N$  electrons, the occupancy of the states  $I_1$  and  $I_2$  varies from 0 to  $N$  as  $I$  assumes all possible values. For a given  $I_1$ , a number of different  $I_2$ 's will exist. We label each different states represented by these values of  $I_2$  serially by a number  $k$  ( $k = 1, 2, \dots$ ), and define the array  $J_1$  by  $J_1(I_1) = k$ . This value of  $J_1$ , along with the value of  $J$ , fixes  $J_2$ . The scheme is illustrated in Table I. Thus, if we know  $I$ , we determine  $I_1$  and  $I_2$  by looking at the bits of  $I$ . Then, from Eq. (7) we find  $J$ . With  $J$ , we find  $I$  from  $T(J)$ .

As we discuss below, the Lanczos and recursion methods require the efficient computation of a matrix-vector multiplication. The matrix is the Hamiltonian matrix  $H$  and the vectors are linear combinations of the complete set of states  $|I\rangle$  described by the bits of the integer  $I$ . If  $|\psi\rangle = \sum_I a(I)|I\rangle$  and  $H|\psi\rangle = \sum_{I'} b(I')|I'\rangle$  (or equivalently  $|\psi\rangle = \sum_J a(J)|J\rangle$  and  $H|\psi\rangle$ )

## COMPUTER SIMULATIONS

Table I. Electron configurations, their representations  $I_i$  and  $I_1$ , and position vectors  $J_i(I_i)$  and  $J_1(I_1)$  for a system of 3 electrons and 3 sites.

| Up configuration | $I_i$ | $J_i(I_i)$ | Down configuration | $I_i$ | $J_i(I_i)$ | $J = J_i + J_1$ |
|------------------|-------|------------|--------------------|-------|------------|-----------------|
| 111              | 7     | 1          | 000                | 0     | 0          | 1               |
| 011              | 3     | 1          | 001                | 1     | 1          | 2               |
| 101              | 5     | 2          | 001                | 1     | 1          | 3               |
| 110              | 6     | 3          | 001                | 1     | 1          | 4               |
| 011              | 3     | 1          | 010                | 2     | 4          | 5               |
| 101              | 5     | 2          | 010                | 2     | 4          | 6               |
| 110              | 6     | 3          | 010                | 2     | 4          | 7               |
| 011              | 3     | 1          | 100                | 4     | 7          | 8               |
| 101              | 5     | 2          | 100                | 4     | 7          | 9               |
| 110              | 6     | 3          | 100                | 4     | 7          | 10              |
| 001              | 1     | 1          | 011                | 3     | 10         | 11              |
| 010              | 2     | 2          | 011                | 3     | 10         | 12              |
| 100              | 4     | 3          | 011                | 3     | 10         | 13              |
| 001              | 1     | 1          | 101                | 5     | 13         | 14              |
| 010              | 2     | 2          | 101                | 5     | 13         | 15              |
| 100              | 4     | 3          | 101                | 5     | 13         | 16              |
| 001              | 1     | 1          | 110                | 6     | 16         | 17              |
| 010              | 2     | 2          | 110                | 6     | 16         | 18              |
| 100              | 4     | 3          | 110                | 6     | 16         | 19              |
| 000              | 0     | 1          | 111                | 7     | 19         | 20              |

$= \sum_j b(J')|J'\rangle$ , then by the orthonormality of the states, we have

$$b(J') = \sum_j \langle J'|H|J\rangle a(J). \quad (8)$$

Thus, the matrix-vector multiplication problem reduces to the problem of computing the nonzero matrix elements of  $\langle J'|H|J\rangle$ , or equivalently, the nonzero elements of  $\langle I'|H|I\rangle$ .

The matrix elements for the Hubbard model separate into matrix elements  $\langle I'|K|I\rangle$  for the kinetic energy and matrix elements  $\langle I'|V|I\rangle$  for the potential energy. For the kinetic energy, we write

$$\begin{aligned} \langle I'|K|I\rangle &= -t\Delta(I'_i, I_i) \sum_{i=1}^M \langle I'_i | (c_{i,i}^\dagger c_{i+1,i} + c_{i+1,i}^\dagger c_{i,i}) | I_i \rangle \\ &\quad - t\Delta(I'_i, I_i) \sum_{i=1}^M \langle I'_i | (c_{i,i}^\dagger c_{i+1,i} + c_{i+1,i}^\dagger c_{i,i}) | I_i \rangle, \end{aligned} \quad (9)$$

where

$$\Delta(I', I) = \begin{cases} 1, & \text{if } I' = I \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

The effect of the hopping terms  $c_{i,i}^\dagger c_{i+1,i} + c_{i+1,i}^\dagger c_{i,i}$  and  $c_{i,i}^\dagger c_{i+1,i} + c_{i+1,i}^\dagger c_{i,i}$  is to move an electron from site  $i$  to  $i+1$  or from  $i+1$  to  $i$ . This action changes the states  $|I_i\rangle$  and  $|I_1\rangle$  to  $|I_{i'}\rangle$  and  $|I_{1'}\rangle$ . Thus, if  $I_i$  has its  $i$  and  $i+1$  bits equal to 10, then  $I_{i'}$  will have these same bits changed to 01. If the bits are 01, they change to 10. (For the bit

combinations 00 and 11, hopping is not allowed.) We now rewrite Eq. (10) as

$$\begin{aligned} \langle I'|K|I\rangle &= -t \sum_i [\Delta(I'_i, I_{i'}) \Delta(I'_i, I_i) \\ &\quad + \Delta(I'_i, I_{i'}) \Delta(I'_i, I_i)]. \end{aligned} \quad (11)$$

From this expression, we see that for each value of  $i$ , there are two values of  $I'$  that give nonzero matrix elements. The first value is  $I' = I_i + 2^M I_1$  for which the first term in Eq. (12) is nonzero, and the second value is  $I' = I_i + 2^M I_{i'}$  for which the second term is nonzero.

To find  $I_{i'}$  and  $I_{1'}$ , we start by defining a mask equal to  $2^i + 2^{i+1}$  so that its only nonzero bits correspond to lattice sites  $i$  and  $i+1$ . To find  $I_{i'}$ , we first perform a bitwise and (AND) operation with  $I_i$  and the mask and call the result  $K$ . This integer records in its  $i$  and  $i+1$  bits the occupancies of the up spins. Next, we perform a bitwise exclusive or (XOR) operation with  $K$  and the mask and call the result  $L$ . The integer  $L$  is zero or equal to the mask if hopping between  $i$  and  $i+1$  is not allowed. If hopping is allowed, its  $i$  and  $i+1$  bits will be 10 or 01 depending on whether the  $i$  and  $i+1$  bits of  $I_i$  were 01 or 10. If hopping is allowed, then  $I_{i'}$  is simply  $I_i - K + L$ . The subtraction by  $K$  removes from  $I_i$  the original occupancy of the  $i$  and  $i+1$  bits, while the addition of  $L$  inserts the new bit configuration. The analogous sequence is performed on  $I_1$  to find  $I_{1'}$ , and both sequences are repeated for all values of  $i$ .

The evaluation of  $\langle I'|V|I\rangle$  is even simpler. We write

$$\langle I'|V|I\rangle = U\Delta(I', I) \sum_i \langle I | n_{i,i} n_{i,i} | I \rangle. \quad (12)$$

From the form of Eq. (12), we see that the state is unchanged, and the computational task is to count the number of doubly occupied sites in  $|I\rangle$ . To do so, we define a mask equal to  $2^i + 2^{M+i}$ . If the result of an AND operation on  $I$  with this mask equals the mask, then site  $i$  is doubly occupied and contributes 1 to the sum. This procedure is repeated for all values of  $i$ .

If conservation of  $N_\sigma$  is used, the two-level scheme is still used. All that changes is storing in the look-up table only those values of  $I$  that have the correct  $N_\sigma$ . The configurations between the horizontal lines in Table I correspond to the different fixed  $N_\sigma$  cases. To use any one of these blocks, all that is needed is a relabeling of the  $J_i$ 's and  $J$ 's. For example, if we have  $N_\uparrow = 2$  and  $N_\downarrow = 1$ , we resequence  $J$  from 2-10 to 1-9, and then change the  $J_i$  so that Eq. (7) is satisfied.

If translational symmetry is used, we can use the sublattice coding scheme of Lin.<sup>8</sup> This method is a bit more difficult to explain and program. The idea is take the two table storage scheme and use translational symmetry to reduce it to a smaller two table scheme. The problem is more complicated than the other cases because the Bloch state represents a linear combination of many-body basis states that is, in general, a complex number because of the  $e^{ikx}$  weights. This complication makes it difficult to explain the method succinctly. In many applications, however, the wave number of the ground state is often known beforehand, and we need only consider this one value. To give a flavor of using translational symmetry, we will assume we are interested only in the  $k=0$  state for which the phase factor becomes unity and the Bloch state is a simple linear combination of those states connected by translational symmetry.

We use Table I for illustrative purposes and focus on the configurations in the up configuration table. For a given value of  $N_\uparrow$ , we assign indices serially to all possible distributions of  $N_\uparrow$  electrons among  $M$  sites. For  $N_\uparrow = 2$  and  $M = 3$ , these distributions are 011, 101, and 110. For a given value of  $N_\uparrow$ , we determine from all the possible configurations of  $N_\uparrow$  among  $M$  sites the sets of configurations that are independent under translational symmetry.

Then, we choose from each set a representative state and serially label these states. The final step is to associate each of these states with each of the up spin states. For  $N_\uparrow = 1$ , the only distribution independent under translational symmetry is 001. The result is illustrated in Table II. An important point is that translational symmetry is applied simultaneously to both up and down spins. For  $N_\uparrow = 2$  and  $N_\downarrow = 1$ , translational symmetry generates from each entry in Table II three entries ( $J = 2, 7$ , and 9) in Table I. This degeneracy leads to an additional requirement that we record and use the weight  $W$  of each configuration.

The methods we described are efficient and minimize storage requirements. The specific details of their implementation depend greatly on the architecture of the computer, the bitwise operations in the compiler's library, and the degree of portability desired.<sup>7</sup> On a Cray-2 computer, one Lanczos iteration for a  $4 \times 4$  Hubbard model with 8 up and 8 down electrons requires only 90 s. For a 18 site Hubbard model, the largest size studied by this method, the computation time increases by an order of magnitude.<sup>9</sup>

We now describe the Lanczos method. We will assume in the following that the block of the Hamiltonian matrix  $H$  whose eigenvalues we want to calculate is called  $A$  and is of order  $n$ . For the labeling schemes presented above,  $n$  equals the maximum value of  $J$ . What is needed is a method for determining at least some of the eigenvalues and eigenstates of  $A$  without storing  $A$  and storing only a few vectors with  $n$  components ( $n$  vectors). The limiting factor is the amount of memory needed to store the vectors. We will describe a version of the Lanczos algorithm that needs to store only two  $n$  vectors, if just a few lowest lying eigenstates are being determined, and three such vectors, if the corresponding eigenvectors also are desired. Storing the matrix elements of  $A$  is unnecessary.

The concept of an invariant subspace is important for understanding the Lanczos method.<sup>2</sup> By a subspace, we mean the set of all  $n$  vectors that can be written as linear combinations of a set  $S = \{s_1, s_2, \dots, s_m\}$  of  $n$  vectors. If a matrix  $A = A^T$ , which is true in our case, the subspace is said to be invariant under  $A$  if for any vector  $x$  in the sub-

Table II. Electron configurations, their representations  $I_1$  and  $I_2$  and position vectors  $J_1(I_1)$  and  $J_2(I_2)$  for a system of 3 electrons and 3 sites after translational symmetry for  $k=0$  is applied to the configurations in Table I.

| Up configuration | $I_1$ | $J_1(I_1)$ | Down configuration | $I_2$ | $J_2(I_2)$ | $J = J_1 + J_2$ | $W$ |
|------------------|-------|------------|--------------------|-------|------------|-----------------|-----|
| 111              | 1     | 1          | 000                | 1     | 0          | 1               | 1   |
| 011              | 1     | 1          | 001                | 1     | 0          | 1               | 3   |
| 101              | 2     | 2          | 001                | 1     | 0          | 2               | 3   |
| 110              | 3     | 3          | 001                | 1     | 0          | 3               | 3   |
| 001              | 1     | 1          | 011                | 1     | 0          | 1               | 3   |
| 010              | 2     | 2          | 011                | 1     | 0          | 2               | 3   |
| 100              | 3     | 3          | 011                | 1     | 0          | 3               | 3   |
| 000              | 1     | 1          | 111                | 1     | 0          | 1               | 1   |

## COMPUTER SIMULATIONS

space, the vector  $Ax$  also is in the subspace. What does this invariance mean? Any eigenvector  $y$  of  $A$ , for example, determines an invariant subspace of dimension one because  $Ay = \lambda y$ , where  $\lambda$  is the eigenvalue, and  $\lambda y$  is an obvious linear combination of  $y$ . A set of  $m$  eigenvectors determines an invariant subspace of dimension  $m$ . Thus, any invariant subspace of  $A$  is spanned by a set of the eigenvectors of  $A$ .

If  $Q = \{q_1, q_2, \dots, q_m\}$  is a basis of an invariant subspace of  $A$ , arranged in the form of an  $n \times m$  matrix  $Q$  whose columns are  $q_i$ , then the matrix product  $AQ$  is a  $n \times m$  matrix whose columns are linear combinations of the columns of  $Q$ . Because of the invariance of the subspace, each vector  $Aq_j$  is in the subspace. These linear combinations can be expressed as the  $m \times n$  matrix  $QT$ , where  $T$  is an  $m \times m$  matrix. Thus, we have

$$AQ = QT. \quad (13)$$

If  $Q$  is an orthonormal basis in the subspace, i.e.,  $Q^T Q = I$ , then we can write

$$Q^T A Q = T. \quad (14)$$

What does the relation Eq. (14) do for us? If  $\lambda$  and  $y$  are an eigenpair of  $T$ ,

$$Ty = \lambda y, \quad (15)$$

then multiplying Eq. (15) by  $Q$  produces  $(QT)y = \lambda(Qy)$ . If we use Eq. (13), we find

$$A(Qy) = \lambda(Qy). \quad (16)$$

Equation (16) says that  $\lambda$  and  $Qy$  are an eigenvalue and eigenvector of  $A$ . Thus, the eigenvalues and eigenvectors of a large matrix  $A$  can be found from those of a smaller matrix  $T$ , if the space spanned by  $Q$  is invariant under  $A$ . The Lanczos method generates such an invariant subspace approximately. As a subspace, it uses the Krylov subspace defined by  $K = \{b, Ab, A^2 b, \dots, A^{m-1} b\}$ , where  $b$  is an arbitrary nonzero vector. The reasoning is roughly as follows: Under the action of  $A$ , the vectors  $Ab, A^2 b, \dots, A^m b$  are all in the Krylov space, except for the last vector. It can be shown that  $A^{m-1} b$  converges to an eigenvector of  $A$  if  $m$  is sufficiently large, so that  $A^m b$  is approximately proportional to  $A^{m-1} b$ , and hence is almost in the Krylov space. Thus, the Krylov space for  $m < n$  is almost an invariant subspace of  $A$ , and to a very good approximation, we can find eigenvalues and eigenvectors of the large matrix  $A$  from those of a smaller matrix. Several remarkable properties<sup>2</sup> follow from the selection of  $K$ :

- The matrix  $T = Q^T A Q$  is a symmetric tridiagonal matrix.
- A three term recursion relation exists for the calculation of the columns of  $Q$ .
- The matrix  $A$  is needed only to compute matrix-vector multiplications.
- Convergence is fast. (Typically, we need only  $m = 30 - 100$  even for a matrix as large as 16

million by 16 million. In some cases, e.g., if all off-diagonal matrix elements have the same sign, convergence is guaranteed.)

Let us see how these properties lead to a practical algorithm. We first want to compute the elements of the matrix,

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \dots & & 0 \\ \beta_1 & \alpha_2 & & & & & \vdots \\ & & \ddots & & & & \\ \vdots & & & \ddots & & & \\ 0 & \dots & & & \beta_{m-1} & & \alpha_m \end{bmatrix}. \quad (17)$$

If we equate columns of  $AQ = QT$ , we find the three-term recursion relation

$$Aq_j = \beta_{j-1} q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}, \quad (18)$$

where  $j = 1$  to  $m - 1$ , and  $q_j$  is the  $j$ th column of  $Q$ . The procedure begins by setting  $\beta_0 = 1$  and  $q_0 = 0$ . Then we let  $q_1 = b$ , where  $b$  is an arbitrary vector consistent with the symmetry of  $A$ . The orthonormality of  $q_j$  implies that  $\alpha_j = \langle q_j | A q_j \rangle$  and  $\beta_j = \langle q_j | A q_{j-1} \rangle = \langle q_{j-1} | A q_j \rangle$ . With these values of  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$  and  $\beta = \{\beta_1, \beta_2, \dots, \beta_{m-1}\}$ , the tridiagonal matrix  $T$  is formed, and its eigenvalues  $\lambda_i$  and eigenvectors  $y_i$  are found by a standard method.<sup>6</sup> Here, we can use these methods because  $T$  is a much smaller matrix than  $A$ . The process is repeated until the lowest eigenvalues converge to some desired accuracy. We remark that usually the eigenvalues are first determined; then computer storage is needed for only  $q_j$  and  $q_{j-1}$ .

If  $y_k$  is an eigenvector of  $T$ , then to compute the corresponding eigenvector  $\psi_k$  of  $A$  we see from Eq. (16) that we need to compute  $\psi_k = Q y_k$ . Because we did not store  $q_i$  while generating  $T$ , we need to regenerate the  $q_i$  by starting with the same  $b$  and repeating the Lanczos procedure. Now, in addition to storing  $q_{j-1}$  and  $q_j$ , we have to store a third  $n$  vector,  $\psi_k$ .

If this procedure sounds too good to be true, one might ask, "What is the catch?" The difficulty with the Lanczos method is that finite precision arithmetic causes the  $q_i$  to lose their orthogonality. One fix is to repeatedly reorthogonalize, another is to partially reorthogonalize, and a third is to ignore the problem.<sup>3,5</sup> We generally choose the latter to avoid the overhead associated with reading from and writing to the disk the long vectors that are needed to reorthogonalize. The cost is that only the extremal (lowest or highest few) eigenvalues can be determined accurately, but these are generally all that were wanted in the first place.

The following pseudocode<sup>3</sup> performs an effective Lanczos method. It assumes the existence of a function  $\text{mult}(A, q)$  that multiplies the matrix  $A$  times the vector  $q$  and returns the resulting vector. In this pseudocode,  $\alpha$  and  $\beta$  are vectors of the order of the maximum number of Lanczos steps planned, and  $b$  and  $q$  are  $n$  vectors. The

components of the vectors are indicated by a subscript  $i$  or  $j$ ;  $t$  is a scalar. The symbol  $\|q\|$  is a vector norm, usually taken to be the  $L_2$ -norm  $\sqrt{q^T q}$ . The vector  $b$  is an arbitrary nonzero vector consistent with the symmetry of  $A$ ,

```

q(1:n) = 0; beta_0 = 0; j = 0
while beta_j != 0
  if j != 0
    for i = 1:n
      t = b_i; b_i = q_i/beta_j; q_i = -beta_j/t
    end
  end
  q = q + mult(A,b)
  j = j + 1; alpha_j = b^T q; q = q - alpha_j b; beta_j = ||q||
end

```

(19)

How is this code used? After the  $m$ th pass through the while loop, the accrued components of  $\alpha$  and  $\beta$  define a tridiagonal matrix of order  $m$ . A conventional eigenvalue routine is used to find the eigenvalues of this matrix. If  $\lambda_0(m)$  is the lowest eigenvalue at step  $m$ , the procedure is repeated until  $|\lambda_0(m) - \lambda_0(m-1)|/|\lambda_0(m)| < \epsilon$ , where  $\epsilon$  is a small number, say  $10^{-10}$ .

Orthogonality breaks down in  $\beta_j$  becomes small. If  $\beta_j = 0$ , then we have reduced the tridiagonal matrix  $T$  to at least two tridiagonal parts, i.e., we have produced vectors  $q_i$  that define an invariant subspace of  $A$ . Since the method is supposed to produce at least approximately such a subspace, convergence of the method and the loss of orthonormality seem to go hand-in-hand. Fortunately, experience has shown that the convergence to the external eigenvalues (largest and smallest) is very rapid and the method can be used without reorthogonalization to determine them. Ignoring the loss of orthogonality is not as cavalier as it sounds.

Mathematicians have studied several versions of the Lanczos method and have developed its relation to variational principles and the method of moments. We have presented a version suggested by Paige,<sup>10</sup> who studied four versions and discovered that two of these versions, including the one described above, are numerically more stable and converge faster than the other two. There are other versions of the Lanczos method that require storing three vectors even for the determination of the eigenvalues. We are uncertain of their advantages.

Usually, the eigenpair corresponding to the lowest energy is all that is determined. Many properties of the ground state can easily be determined knowing this eigenpair. For example, if  $|\psi_0\rangle$  is the ground state wave function, then the spin and charge density wave correlations between sites of separation  $j$  are given by

$$\chi_{\pm}(j) = \frac{1}{N} \sum_i \langle \psi_0 | (n_{i,1} \pm n_{i,1}) \times (n_{i+j,1} \pm n_{i+j,1}) | \psi_0 \rangle, \quad (20)$$

where the plus sign refers to the charge density wave.

Similarly, superconducting pairing correlations between sites  $i$  and  $j$  are calculated by<sup>11</sup>

$$\chi_s(i-j) = \langle \psi_0 | \Delta_i \Delta_j^\dagger | \psi_0 \rangle, \quad (21)$$

where

$$\Delta_i = \frac{1}{N} \sum_j c_{j,1} c_{j+i,1}. \quad (22)$$

Perhaps one of the more interesting things that can be done with the ground-state energy  $\lambda_0$  and wave function  $|\psi_0\rangle$  is to compute dynamical properties of the system by the recursion method.<sup>4,5,12</sup> The type of quantity calculated is a spectral function defined as

$$I(\omega) = \sum_m |\langle \psi_m | B | \psi_0 \rangle|^2 \delta(\omega + \lambda_0 - \lambda_m), \quad (23)$$

where  $B$  represents some perturbation of the ground state. For example, if  $B$  is the current operator or the spin-lowering operator, then  $I(\omega)$  is the optical conductivity or the dynamical susceptibility. The action of  $B$  on  $|\psi_0\rangle$  is to create a new state from the ground state. The expression  $|\langle \psi_m | B | \psi_0 \rangle|^2$  is the fraction of this new state that is in an excited state  $|\psi_m\rangle$  of the Hamiltonian. The  $\delta$  function expresses conservation of energy. Using the standard relation

$$\delta(x-x') = -\frac{i}{\pi} \text{Im} \lim_{\epsilon \rightarrow 0} \frac{1}{x-x'+i\epsilon}, \quad (24)$$

and the resolution of unity

$$1 = \sum_m |\psi_m\rangle \langle \psi_m|, \quad (25)$$

we can rewrite Eq. (23) as

$$I(\omega) = -\frac{1}{\pi} \text{Im} \lim_{\epsilon \rightarrow 0} \langle \psi_0 | B^\dagger \frac{1}{zI-H} B | \psi_0 \rangle, \quad (26)$$

where  $z = (\omega + \lambda_0 + i\epsilon)$ . In this form, what must be computed is a specific element of the inverse of the matrix  $zI-H$ . What the recursion method does is to use the Lanczos method with  $B|\psi_0\rangle$  as the initial state to tridiagonalize  $H$ . In this simple form, the matrix inverse of  $zI-H$  for the specific element is readily obtained. Of course, the procedure is not used on the entire Hamiltonian matrix  $H$ . Usually, it is used only on the block that contains the ground state.

The spectral function  $I(\omega)$  also can be expressed as a continued fraction<sup>4,5,12</sup>

$$I(\omega) = -\frac{1}{\pi} \text{Im} \lim_{\epsilon \rightarrow 0} \frac{1}{z - \alpha_1 - \frac{\beta_1^2}{z - \alpha_2 - \frac{\beta_2^2}{\ddots}}}. \quad (27)$$

One way to evaluate the continued fraction is by "brute force." Another way is to observe the connection between a continued fraction and a "stair-case" Padé approximant<sup>13</sup> and to use a simple set of recursion equations to

## COMPUTER SIMULATIONS

evaluate the approximant. A third way is to use the eigenvalues and eigenvectors of  $T$  and evaluate Eq. (23) in this diagonal basis

$$I(\omega) = \lim_{\epsilon \rightarrow 0} \sum_m |\langle y_m | B | y_0 \rangle|^2 \frac{\epsilon}{\epsilon^2 + (\omega + \lambda_0 - \lambda_m)^2}. \quad (28)$$

One advantage of the recursion method is that we only collect states that are connected to the ground state by the operators whose spectrum we wish to calculate. This fact saves much computer time. The major disadvantage is that the finite size of the system causes the spectral functions to be a sum of  $\delta$  functions. Some smoothing of the spectrum is achieved by having  $\epsilon$  remain a small finite value in the range  $10^{-3}$ – $10^{-6}$ . Other methods also have been proposed.<sup>14</sup>

To conclude, we emphasize that the exact diagonalization method is an important, frequently used tool in theoretical studies of many-body problems. The method has been used, for example, to verify the existence of magnetic long-range order in the 2D Heisenberg model and to support the use of the 2D antiferromagnetic Heisenberg model to describe some properties of the newly discovered high- $T_c$  superconductors.<sup>15</sup> In contrast to quantum Monte Carlo methods, which frequently suffer from the "sign" problem,<sup>16</sup> the exact diagonalization method gives exact answers for many-body models on

finite lattices. Hence, the method is a means of studying the applicability of theoretical models to experiments, as well as the validity of approximations used in analytical methods. Often, the results of the exact diagonalization method point to parameter regimes where interesting physics may occur.

The main limitation of this method is its restriction to small lattices, and thus properties in the thermodynamic limit are difficult to obtain. Sometimes, however, this limitation is only an apparent one. The results for the small lattice might meaningfully represent those of the large systems because many-body interactions are short ranged and can lead to phenomena with short coherence lengths. If this length is smaller than the lattice size accessible by the exact diagonalization method, physically meaningful results are obtained. This situation frequently arises for 1D systems. In other cases, an extrapolation of results for finite sizes to infinite system sizes is possible. Such extrapolation processes were used, for example, in the exact diagonalization calculations of the staggered magnetization of two 2D Heisenberg models.<sup>15</sup> The exact diagonalization method is not a replacement for analytical analysis; rather, it is a complementary part of it.

### Suggestions for further study

1. Investigate the implementation of the Lanczos method to the spinless fermion<sup>17</sup> and Heisenberg models.<sup>8</sup> What symmetries are useful? Are there exploitable differences between the two models that makes one easier to implement than the other? Do the details of the two-table and sublattice coding schemes change? If so, how?

2. Quantum chemists often use the Davidson method<sup>18</sup> to find the lowest eigenvalues of a large matrix. Investigate its algorithmic structure. What are its advantages and disadvantages in comparison to the version of the Lanczos method described in this column?

3. Investigate more fully the implementation of the recursion method and derive Eq. (27). In deriving Eq. (27), consider the following approach: Replace the  $\alpha_i$  in the tridiagonal matrix  $T$  in Eq. (17) with  $z - \alpha_i$  and call the resulting matrix  $S_m$ . Then express  $S_m$  as the block matrix

$$S_m = \begin{bmatrix} a_1 & b_1^T \\ b_1 & S_{m-1} \end{bmatrix}, \quad (29)$$

where  $a_1 = z - \alpha_1$ ,  $b_1$  is a column vector of length  $m - 1$  with elements  $\{\beta_1, 0, \dots, 0\}$ , and  $S_{m-1}$  is the  $(m - 1) \times (m - 1)$  matrix formed from  $S_m$  by deleting its first row and column. Block invert  $S_m$  and show that the 11 element of the inverse is

$$(S_m^{-1})_{11} = \frac{1}{a_1 - \langle b_1 | \frac{1}{S_{m-1}} | b_1 \rangle}, \quad (30)$$

which equals  $[a_1 - \beta_1^2 (S_{m-1}^{-1})_{11}]^{-1}$ . To find the 11 element of  $S_{m-1}^{-1}$ , block  $S_{m-1}$ , and then block invert it. Repeating this procedure until only the inversion of  $S_1$  is

## Measurement Errors Theory and Practice

Semyon Rabinovich

This volume offers practical recommendations and procedures for problems related to the estimation of measurement errors. The author covers a wide range of subjects, including classical concepts of metrology, modern problems of instrument calibration, estimation of single and multiple measurement errors, and modern probability-based methods of error estimation. A valuable resource for graduate students, applied physicists, and engineers.

284 pages, cloth, ISBN 0-88318-866-X  
\$100.00 (Member price \$80.00)

Please indicate your AIP Member Society when ordering.

To order, call 1-800-488-BOOK

In Vermont: 1-802-878-0315. Fax: 1-802-878-1102  
Or mail check, MO, or PO (plus \$2.75 for shipping) to:

**AMERICAN  
INSTITUTE  
OF PHYSICS**

American Institute of Physics  
do AIPC, 64 Depot Road  
Colchester, VT 05446

left produces the continued fraction. Are the Padé and diagonal-space methods more stable or efficient than the brute force method for evaluating the continued fraction?

4. The recursion method also can be used to perform perturbation theory. For  $H = H_0 + V$ , where  $H_0|\phi_0\rangle = E_0|\phi_0\rangle$ , the idea is to compute the Green's function  $(zI - H)^{-1}$ , where  $z = E + i\epsilon$  and obtain the continued fraction. Because the poles of the Green's function are the eigenvalues of the Hamiltonian, the poles of the continued fraction are approximations to these eigenvalues. To develop these approximations, one takes unperturbed state  $|\phi_0\rangle$ , instead of  $B|\psi_0\rangle$ , as the initial state in the Lanczos step to tridiagonalize  $H$ . Work through this procedure analytically to second order and compare the result with second-order Brillouin-Wigner perturbation theory.

### Acknowledgments

This work was supported by the U.S. Department of Energy and the Department of Physics at the University of Illinois. We also thank the editors, Jan Tobochnik and Harvey Gould, for helpful comments and suggestions about the manuscript.

### References

1. C. Lanczos, *J. Res. Natl. Bur. Stand.* **45**, 225 (1950).
2. S. Pissanetsky, *Sparse Matrix Technology* (Academic Press, New York, 1984), Chap. 6.
3. G. H. Golub and C. F. van Horn, *Matrix Computations* (Johns Hopkins Press, Baltimore, 1989), Chap. 9.
4. R. Haydock, in *Solid State Physics* **35**, edited by H. Ehrenreich, F. Seitz, and D. Turnbull (Academic, New York, 1980), p. 215.
5. R. E. Wyatt in *Adv. Chem. Phys.* **LXXII**, 231 (1988).
6. LAPACK Users' Guide, edited by E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Cruz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostouchov, and D. Sorenson, SIAM, Philadelphia (1992).
7. P. W. Leung and P. E. Oppenheimer, *Comp. Phys.* **6**, 603 (1993).
8. H. Q. Lin, *Phys. Rev. B* **42**, 6561 (1990).
9. H. Q. Lin, *Phys. Rev. B* **44**, 7153 (1991).
10. C. C. Paige, *J. Inst. Math. Appl.* **10**, 373 (1972).
11. For example, see H. Q. Lin, J. E. Hirsch, and D. J. Scalapino, *Phys. Rev. B* **37**, 7359 (1988).
12. E. R. Gagliano and C. A. Balserio, *Phys. Rev. Lett.* **59**, 2999 (1987).
13. G. A. Baker, Jr., *Essentials of Padé Approximants* (Academic, New York, 1975), Chap. 4.
14. E. Y. Loh, Jr. and D. K. Campbell, *Synth. Metals* **27**, A499 (1988).
15. For a recent general survey, see E. Manousakis, *Rev. Mod. Phys.* **63**, 1 (1991).
16. For example, see E. Y. Loh, Jr., J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, *Phys. Rev. B* **41**, 9301 (1990), and references within.
17. W. R. Somsy and J. E. Gubernatis, *Comput. Phys.* **6**, 178 (1972).
18. E. R. Davidson, *J. Comp. Phys.* **17**, 87 (1975).

*From the editors.* We appreciate your feedback and encourage your contributions to this column. A copy of the guidelines are available from the editors. Please send comments and suggestions for future columns to hgould@vax.clarku.edu or jant@kzoo.edu.

# PHYS ADVANTAGE

## AN ONLINE BIBLIOGRAPHIC DATABASE FOR PHYSICISTS AND ASTRONOMERS

PHYS (Physics Briefs) is an online database—available exclusively on STN International—that enables you to search through physics literature worldwide to obtain English-language citations. Abstracts, bibliographic data, extensive index entries—PHYS gives you access to more than 1.4 million citations in all.

You will find citations from 1979 onward—information drawn from more than 2,800 scientific and technical journals, books, reports, conference proceedings, patents, and nonconventional literature (e.g., dissertations and corporate publications). The PHYS database is updated twice monthly with 120,000 new records added each year. Abstracts are often available simultaneously with publication of the articles.

### Major advantages of PHYS include:

**Immediacy**—30% of all journal citations are available within one month of publication

**Comprehensiveness**—significant coverage of monographs and reports; strong emphasis on Eastern European literature; indexing of all astronomical objects

**Searchability**—editors and authors are searchable, making citations easier to locate—especially helpful for conference proceedings literature

**Flexible Pricing**—choose either a discounted rate or an annual flat fee.

Act Now! For more information on the new academic pricing programs for PHYS contact the American Institute of Physics, Electronic Publishing Division.

**AMERICAN  
INSTITUTE  
OF PHYSICS**

Electronic Publishing  
500 Sunnyside Boulevard  
Woodbury, NY 11797-2999  
(516) 576-2262/2264  
E-mail: ellen@pinet.aip.org

PHYS is a cooperative effort of Fachinformationszentrum Energie, Physik, Mathematik and the American Institute of Physics.