

1993

Computation of Dominant Eigenvalues and Eigenvectors: A Comparative Study of Algorithms

M. P. Nightingale

University of Rhode Island, nigh@uri.edu

V. S. Viswanath

University of Rhode Island

Gerhard Müller

University of Rhode Island, gmuller@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/phys_facpubs

Citation/Publisher Attribution

Nightingale, M. P., Viswanath, V. S., & Muller, G. (1993). Computation of dominant eigenvalues and eigenvectors: A comparative study of algorithms. *Physical Review B*, 48(10), 7696-7699. doi: 10.1103/PhysRevB.48.7696

Available at: <http://dx.doi.org/10.1103/PhysRevB.48.7696>

This Article is brought to you by the University of Rhode Island. It has been accepted for inclusion in Physics Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu. For permission to reuse copyrighted content, contact the author directly.

Computation of Dominant Eigenvalues and Eigenvectors: A Comparative Study of Algorithms

Terms of Use

All rights reserved under copyright.

Computation of dominant eigenvalues and eigenvectors: A comparative study of algorithms

M. P. Nightingale, V. S. Viswanath, and Gerhard Müller

Department of Physics, The University of Rhode Island, Kingston, Rhode Island 02881-0817

(Received 15 June 1992)

We investigate two widely used recursive algorithms for the computation of eigenvectors with extreme eigenvalues of large symmetric matrices—the modified Lanczös method and the conjugate-gradient method. The goal is to establish a connection between their underlying principles and to evaluate their performance in applications to Hamiltonian and transfer matrices of selected model systems of interest in condensed matter physics and statistical mechanics. The conjugate-gradient method is found to converge more rapidly for understandable reasons, while storage requirements are the same for both methods.

I. INTRODUCTION

Many important problems in condensed-matter theory require for their solution the determination of extreme eigenvalues and the associated eigenvectors of large Hermitian matrices. In studies of quantum many-body systems, one is interested in the ground state of some model Hamiltonian. The dependence on model parameters (coupling constant, anisotropy, external field, etc.) of the finite-size ground-state wave function yields important clues on the degree and type of ordering in the infinite system at $T=0$ and valuable information on its dynamical properties. In statistical mechanics, the transfer-matrix method in conjunction with finite-size extrapolation techniques has proven to be a very powerful approach to studying the critical properties of two-dimensional lattice systems. In both kinds of applications, a successful analysis depends on the availability of precise data over a range of system sizes. In this paper we undertake a comparative study of two algorithms that have been widely used to produce such data.

The *modified Lanczös* (ML) method is derived, as its name says, from the well-known Lanczös algorithm¹ for matrix tridiagonalization. For the computation of eigenstates, the Lanczös tridiagonalization itself may be the first step of an algorithm. The second step would then be, for example, the bisection method for eigenvalues and inverse iteration for eigenvectors.² The modified version was developed in an effort to adapt the standard Lanczös algorithm for a more direct computation of the ground-state energy and wave function of a Hamiltonian system without the intermediate step of tridiagonalization.^{3,4} The ML method has since been popular in studies of 1D and 2D quantum spin models and models of strongly correlated electronic systems.

The *conjugate-gradient* (CG) method has long been known in the context of minimizing functions of several variables.⁵ It was designed such that for quadratic functions in n variables the algorithm is guaranteed to converge after n steps.⁶ In the context of an eigenvalue problem $H|x\rangle = E|x\rangle$, the CG method can be applied to the minimization of the Rayleigh quotient⁷ $R = \langle x|H|x\rangle / \langle x|x\rangle$. The fact is that the minimum (maximum) value

of R is equal to the lowest (highest) eigenvalue of H . The CG method has proven to be a reliable computational tool in statistical mechanics, notably in the context of the transfer-matrix approach.⁸

II. MODIFIED LANCZÖS METHOD

The idea behind the ML method is to embed the standard Lanczös recursive cycle within another recursive cycle. The outer cycle terminates the inner one after one iteration and resets the initial condition. In practice, the two cycles make up a single loop consisting of two iterative steps. The loop is started by an initial step and terminated by a user-supplied convergence criterion.

Initial step: Select a (normalized) trial vector $|\psi_0\rangle$ for the ground state of the system, which is specified by a Hamiltonian H . $|\psi_0\rangle$ must have a nonzero projection onto the true (but unknown) ground-state wave function $|\phi_0\rangle$.

Iterative step No. 1: Given the k th approximate vector $|\psi_k\rangle$, apply one cycle of the standard Lanczös algorithm to generate a vector $|\gamma_k\rangle$ which is orthonormal to $|\psi_k\rangle$:

$$|\gamma_k\rangle = (H - \langle H \rangle_k) |\psi_k\rangle / \sqrt{\langle H^2 \rangle_k - \langle H \rangle_k^2}, \quad (2.1)$$

where $\langle H^n \rangle_k \equiv \langle \psi_k | H^n | \psi_k \rangle$.

Iterative step No. 2: Construct a new vector $|\psi_{k+1}\rangle$ in the subspace spanned by $|\psi_k\rangle$ and $|\gamma_k\rangle$ such that it minimizes the energy. The lower eigenvalue of H in that subspace reads

$$\epsilon_{k+1} = \epsilon_k - v_k \sqrt{\langle H^2 \rangle_k - \langle H \rangle_k^2}, \quad (2.2)$$

where $\epsilon_k = \langle H \rangle_k$ and

$$v_k = \sqrt{1 + w_k^2} - w_k, \quad (2.3)$$

$$w_k = \frac{\langle H^3 \rangle_k - 3\langle H^2 \rangle_k \langle H \rangle_k + 2\langle H \rangle_k^3}{2(\langle H^2 \rangle_k - \langle H \rangle_k^2)^{3/2}}.$$

The associated eigenvector

$$|\psi_{k+1}\rangle = (|\psi_k\rangle - v_k |\gamma_k\rangle) / \sqrt{1 + v_k^2} \quad (2.4)$$

will be the starting vector of the next iteration.

Termination: Since v_k in (2.3) is non-negative, the se-

quence $\varepsilon_0, \varepsilon_1, \dots$ is monotonically decreasing. Under normal circumstances, it converges toward the exact value of the ground-state energy E_0 , and the sequence $|\psi_0\rangle, |\psi_1\rangle, \dots$ converges toward the exact ground-state wave function $|\phi_0\rangle$. We have used and can recommend the following convergence criterion:

$$\langle\langle H^2 \rangle_k - \langle H \rangle_k^2\rangle / \langle H \rangle_k^2 < \varepsilon, \quad (2.5)$$

where ε is comparable to machine precision.

In the ML algorithm, memory must be allocated for the simultaneous storage of three vectors, $|\psi_k\rangle, H|\psi_k\rangle, H^2|\psi_k\rangle$, in the k th iteration. Each iteration involves two matrix multiplications. We shall see that the rate of convergence in any typical application can be enhanced substantially by what amounts to a minor change in design.

III. CONJUGATE-GRADIENT METHOD

For easy comparison with the ML method, the conjugate-gradient (CG) method is also formulated in terms of an initial step and two iterative steps in a loop that is terminated by a user-supplied convergence criterion.

Initial step: Select a trial vector $|x_0\rangle$ (not necessarily normalized) with nonzero projection onto the ground-state wave function $|\phi_0\rangle$.

Iterative step No. 1: Given in the k th approximate vector $|x_k\rangle$, apply the gradient to the Rayleigh quotient

$$R_k = \langle x_k | H | x_k \rangle / \langle x_k | x_k \rangle \quad (3.1)$$

to generate a vector $|g_k\rangle$ which is orthogonal to $|x_k\rangle$:

$$|g_k\rangle \equiv \nabla R_k = (2/\langle x_k | x_k \rangle)(H|x_k\rangle - R_k|x_k\rangle). \quad (3.2)$$

Iterative step No. 2: Construct a new vector of the form

$$|x_{k+1}\rangle = |x_k\rangle + \alpha_k |p_k\rangle, \quad (3.3)$$

with $|p_k\rangle = -|g_k\rangle + u_{k-1}|p_{k-1}\rangle$ and $u_{k-1} = \langle g_k | g_k \rangle / \langle g_{k-1} | g_{k-1} \rangle$ for $k \geq 1$ ($u_{-1} = 0$), by minimizing the Rayleigh quotient R_{k+1} of (3.3) with respect to the real parameter α_k . The condition $\partial R_{k+1} / \partial \alpha_k = 0$ leads to the quadratic equation

$$A_k \alpha_k^2 + B_k \alpha_k + C_k = 0 \quad (3.4)$$

with coefficients

$$A_k = \langle p_k | H | p_k \rangle \langle x_k | p_k \rangle - \langle x_k | H | p_k \rangle \langle p_k | p_k \rangle, \quad (3.5a)$$

$$B_k = \langle p_k | H | p_k \rangle \langle x_k | x_k \rangle - \langle x_k | H | x_k \rangle \langle p_k | p_k \rangle, \quad (3.5b)$$

$$C_k = \langle x_k | H | p_k \rangle \langle x_k | x_k \rangle - \langle x_k | H | x_k \rangle \langle x_k | p_k \rangle. \quad (3.5c)$$

The larger one of the two solutions of (3.4) minimizes R_{k+1} .

Termination: The sequence R_1, R_2, \dots of minimized Rayleigh quotients converges toward the exact lowest eigenvalue E_0 of H , and the sequence $|x_1\rangle, |x_2\rangle, \dots$ of vectors converges (after normalization) toward the corresponding eigenvector $|\phi_0\rangle$. The convergence criterion corresponding to (2.5) reads

$$\langle g_k | g_k \rangle \langle x_k | x_k \rangle / R_k^2 < 4\varepsilon. \quad (3.6)$$

Note that the vector $|p_k\rangle$ (unlike $|g_k\rangle$) is, in general, not orthogonal to $|x_k\rangle$. The second term, $u_{k-1}|p_{k-1}\rangle$, has the effect of stabilizing the direction of the path in the Hilbert space toward the minimum of the Rayleigh quotient. This enhances the rate of convergence.

What is the most economical implementation of the CG method? The answer depends on whether the most valuable (or most limited) resource is (a) available CPU

TABLE I. Sequence of computations to be performed during the k th iteration of the CG method. Implementation (a) involves one matrix multiplication and requires memory for four vectors; implementation (b) involves one additional matrix multiplication but requires memory for only three vectors. The same sequence of computations, but with several simplifications, applies to the SD method discussed in Sec. IV.

0	Results of $(k-1)$ th iteration: $ x_k\rangle, H x_k\rangle, p_{k-1}\rangle, \langle x_k x_k \rangle, \langle x_k H x_k \rangle, \langle g_k g_k \rangle, R_k$
1(a)	Vector addition: $ g_k\rangle = (2/\langle x_k x_k \rangle)[H x_k\rangle - R_k x_k\rangle]$
1(b)	Vector addition: $ g_k\rangle = (2/\langle x_k x_k \rangle)[H x_k\rangle - R_k x_k\rangle]$, overwrites $H x_k\rangle$
2	Inner product: $\langle g_k g_k \rangle, u_{k-1}$
3	Vector addition: $ p_k\rangle = - g_k\rangle + u_{k-1} p_{k-1}\rangle$
4(a)	Inner products: $\langle x_k p_k \rangle, \langle p_k p_k \rangle, \langle p_k H x_k \rangle$
4(b)	Inner products: $\langle x_k p_k \rangle, \langle p_k p_k \rangle$
5	Matrix multiplication: $H p_k\rangle$, overwrites $ g_k\rangle$
6(a)	Inner product: $\langle p_k H p_k \rangle$
6(b)	Inner products: $\langle x_k H p_k \rangle, \langle p_k H p_k \rangle$
7	Quadratic equation: α_k
8	Vector addition: $ x_{k+1}\rangle = x_k\rangle + \alpha_k p_k\rangle$, overwrites $ x_k\rangle$
9(a)	Vector addition: $H x_{k+1}\rangle = H x_k\rangle + \alpha_k H p_k\rangle$, overwrites $H x_k\rangle$
9(b)	Matrix multiplication: $H x_{k+1}\rangle$, overwrites $H p_k\rangle$
10	Inner products: $\langle x_{k+1} x_{k+1} \rangle, \langle x_{k+1} H x_{k+1} \rangle, R_{k+1}$

time or (b) accessible memory. Implementation (a) requires only a single matrix multiplication per iteration (one less than ML) but needs memory for the simultaneous storage of four vectors (one more than ML). Implementation (b), by contrast, requires two matrix multiplications per iteration and storage space for three vectors, just as the ML method does. The sequence of computations in one iteration of the two implementations of the CG method as dictated by the above-mentioned minimum requirements are summarized in Table I.

The *steepest-descent* (SD) method is based on the same principle as the CG method—the minimization of the Rayleigh quotient for a one-parameter vector. But, unlike the CG method, the SD algorithm constructs the new vector $|x_{k+1}\rangle$ from the previous one and the gradient of its Rayleigh quotient R_k alone,

$$|x_{k+1}\rangle = |x_k\rangle - \alpha_k |g_k\rangle, \quad (3.7)$$

and thus ignores the directional information from the previous iterations that is contained in the vector $|p_k\rangle$ used in the CG method. Nevertheless, the general structure of the algorithm remains the same as in the CG case. Implementations (a) and (b) detailed in Table I are still applicable, but with some obvious simplifications. None of them reduces the number of matrix multiplications per iteration or the number of vectors to be stored simultaneously.

It is readily seen that the ML and SD methods are equivalent. For $|p_k\rangle = -|g_k\rangle$, the coefficients (3.5) of the quadratic equation that determines the parameter α_k in (3.7) can be expressed in terms of $\langle x_k|x_k\rangle$ and $\langle H^n\rangle_k \equiv \langle x_k|H^n|x_k\rangle/\langle x_k|x_k\rangle$. The results for a normalized vector (used here for mere convenience) read $A_k = 8(\langle H^2\rangle_k - \langle H\rangle_k^2)^2$, $B_k = 4(\langle H^3\rangle_k - 3\langle H^2\rangle_k\langle H\rangle_k + 2\langle H\rangle_k^3)$, $C_k = -2(\langle H^2\rangle_k - \langle H\rangle_k^2)$, yielding the parameter value

$$\alpha_k = v_k / \sqrt{\langle g_k|g_k\rangle}. \quad (3.8)$$

Hence the vectors (2.4) and (3.7) with (3.8) differ only in normalization. Likewise, the Rayleigh quotient R_{k+1} is equal to the eigenvalue ε_{k+1} given in (2.2).

IV. TWO ILLUSTRATIONS

For a comparison of the ML/SD and CG methods we have selected two systems at criticality. Their eigenvalue spectra are computationally challenging in that the gaps scale inversely proportionally to the system size. Consider first the 1D $s = \frac{1}{2}$ Heisenberg antiferromagnet $H = J \sum_l \mathbf{S}_l \cdot \mathbf{S}_{l+1}$, with N spins and periodic boundary condition. The ground state for even N is known to be a singlet ($S_T = 0$), with wave number $k = 0$ for even $N/2$ and $k = \pi$ for odd $N/2$. We employ a type (b) implementation of the CG and ML/SD algorithms and use translationally invariant basis vectors in the invariant subspace $S_T^z = 0$. The initial trial wave function used in both cases is a translationally invariant linear combination of Néel states

$$|\Psi_0\rangle = 2^{-1/2} [|\uparrow\downarrow\uparrow\downarrow\cdots\downarrow\rangle + (-1)^{N/2} |\downarrow\uparrow\downarrow\uparrow\cdots\uparrow\rangle].$$

For this comparative test we have run the programs for systems of four different sizes. The number of iterations it takes the two methods ML/SD (CG) to satisfy the convergence criterion (3.6) with $\varepsilon = 10^{-13}$ is 48 (21) for $N = 12$, 59 (24) for $N = 14$, 73 (27) for $N = 16$, and 96 (30) for $N = 18$.

The faster convergence of the CG algorithm is quite evident. Not only does it require less than half the number of iterations in comparison to the ML/SD method, that number increases considerably more slowly with system size too. The fact is that the CG algorithm makes the sequence of wave vectors $|x_1\rangle, |x_2\rangle, \dots$ progress on a more direct path toward the exact ground-state wave function than the ML/SD algorithm does. A strong indicator of the directness of that path is the sequence of angles θ_k between the vectors $|p_{k-1}\rangle$ and $|p_k\rangle$ of successive iterations. In the ML/SD method successive directions of the path toward the exact ground state are orthogonal to one another: $\langle \gamma_{k-1}|\gamma_k\rangle = 0$ and $\langle g_{k-1}|g_k\rangle = 0$, yielding $\theta_k = 90^\circ$ for all k . We have calculated the angles θ_k , $k = 1, \dots, 30$ obtained by the CG algorithm for the case $N = 18$ and found that all of them range between 45° and 60° , i.e., significantly below 90° . They tend to fluctuate within that range with no apparent trend toward lower or higher values.

Our statistical mechanics application pertains to the transfer matrix of the q -state Potts model on an $n \times \infty$ square lattice with periodic boundary conditions.⁹ The transfer-matrix multiplication was implemented in terms of a sparse matrix factorization.⁸ It was simplified by the assumption of symmetry under permutation of the Potts states. This reduces the order of the matrix by a factor q .

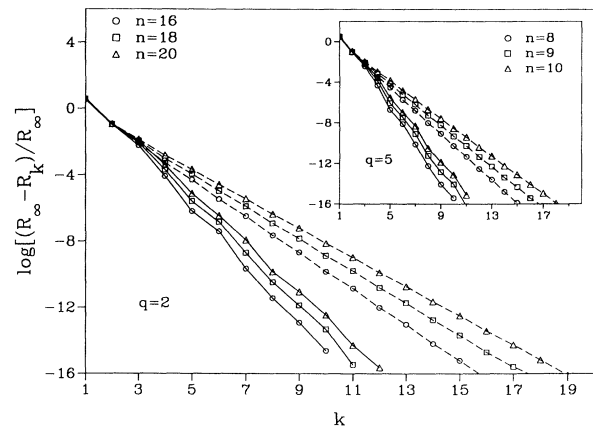


FIG. 1. Logarithm (base 10) of the relative deviation of the eigenvalue estimates R_k from the asymptotic value R_∞ , plotted vs k (the number of iterations). The value of R_∞ has been approximated by our best estimate, which satisfies the convergence criterion (3.6) to within machine precision. The main plot shows data for the Ising model ($q=2$) and the inset for the five-state Potts model. The data points connected by solid lines have been obtained from the CG method and those connected by dashed lines from the ML/SD method. In each case data for three different system sizes have been produced.

No other symmetries were used.

Figure 1 shows the logarithm of the relative deviation of the eigenvalue estimates R_k from the asymptotic value R_∞ , plotted versus the number of iterations. The main plot shows data for the Ising model ($q=2$) and the inset for the five-state Potts model. The data points connected by solid lines have been obtained with the CG method and those connected by dashed lines with the ML/SD method. The magnitude of the slope is a measure for the rate of convergence. In all cases shown, that rate is significantly larger for the CG method than for the ML/SD method. For the computation of these data we have chosen initial vectors with all elements set equal to unity. We have also experimented with random initial vectors, which is not a typical choice in a practical calculation. In most cases, we have found results that were

very similar to those shown in Fig. 1. However, the convergence of the CG results tends to be more erratic for random initial vectors, and in one exceptional case convergence was marginally slower than for ML/SD.

ACKNOWLEDGMENTS

This work was supported by the U.S. National Science Foundation under Grant Nos. DMR-92-14669 and DMR-90-07540. The computations were performed in part at the Cornell National Supercomputing Facility, a resource of the Center for Theory and Simulation in Science and Engineering at Cornell University, and in part at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

¹C. Lanczös, J. Res. Nat. Bur. Stand. **45**, 255 (1950).

²H. Nishimori and Y. Taguchi, Prog. Theor. Phys. Suppl. **87**, 247 (1986).

³E. Dagotto and A. Moreo, Phys. Rev. D **31**, 865 (1985).

⁴E. R. Gagliano, E. Dagotto, A. Moreo, and F. C. Alcaraz, Phys. Rev. B **34**, 1677 (1986).

⁵M. R. Hestenes and E. Stiefel, J. Res. Bur. Nat. Stand. **49**, 409 (1952).

⁶G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns

Hopkins University Press, Baltimore, 1983).

⁷W. W. Bradbury and R. Fletcher, Num. Math. **9**, 259 (1966).

⁸M. P. Nightingale, in *Finite-Size Scaling and Numerical Simulation of Statistical Systems*, edited by V. Privman (World Scientific, Singapore, 1990), p. 289; M. P. Nightingale, Proc. K. Ned. Wet. Ser. B **82**, 245 (1978).

⁹For a review on the Potts model, see F. Y. Wu, Rev. Mod. Phys. **54**, 235 (1982).