Volume 1 | Number 1 | Jan 2013 | Pages 1–100
PCCP
Physical Chemistry Chemical Physics
www.rsc.org/pccp
ROYAL SOCIETY OF CHEMISTRY

Neural network iterative diagonalization method to solve eigenvalue problems in quantum mechanics

Hua-Gen Yu[1]

Department of Chemistry, Brookhaven National Laboratory,
Upton, NY 11973-5000, USA

(May 3, 2015)

Abstract

We propose a multi-layer feed-forward neural network iterative diagonalization method (NNiDM) to compute some eigenvalues and eigenvectors of large sparse complex symmetric or Hermitian matrices. The NNiDM algorithm is developed by using the complex (or real) guided spectral transform Lanczos (cGSTL) method, thick restart technique, and multi-layered basis contraction scheme. Artificial neurons (or nodes) are defined by a set of formally orthogonal Lanczos polynomials, where the biases and weights are dynamically determined through a series of cGSTL iterations and small matrix diagonalizations. The algorithm starts with one random vector. The last output layer produces wanted eigenvalues and eigenvectors near a given reference value via a linear transform diagonalization approach. Since the algorithm uses the spectral transform technique, it is capable of computing interior eigenstates in dense spectrum regions. The general NNiDM algorithm is applied for calculating energies, widths, and wavefunctions of two typical molecules HO_2 and CH_4 as examples.

[1] E-mail:hgy@bnl.gov

1 Introduction

The eigenvalue problem is one of the most important topics in quantum mechanics. It often results from numerically solving Schrödinger equations in electronic structure, spectroscopy, and quantum scattering studies.[1–6] In those fields, secular equations are formed by either a Hermitian Hamiltonian $\hat{H}$ or its extended one $\tilde{H} = \hat{H} - iW$ in a basis set. Here $-iW$ is a negative imaginary potential (NIP)[7] used to impose proper scattering boundary conditions. Although $\tilde{H}$ is non-Hermitian, its matrix representation is complex symmetric. It gives complex eigenvalues[4,7–9] as $z_n = E_n - i\Gamma_n/2$. Such an eigenstate is called a resonance with an energy of E_n and a width of Γ_n (or lifetime $\tau_n = \hbar/\Gamma_n$). Resonance wavefunctions are defined by eigenvectors. Of course, a Hermitian Hamiltonian always gives real eigenvalues, i.e., energies E_n.

For many-body systems, it is challenging to solve the eigenvalue problem of the system owing to the huge dimension of the matrix.[10–14] In practice, one often uses an iterative diagonalization method,[2,15–26] in which only the Hamiltonian-vector products are required without explicitly constructing the Hamiltonian matrix. The high efficiency of an iterative diagonalization algorithm is attributed by the sparseness of matrix. Currently, there are many highly efficient iterative algorithms to calculate eigenvalues. Among them the two most powerful approaches are the Lanczos- and Chebyshev-based algorithms.[10,16,21,22,27–29,29–33] Those low storage algorithms[21,29–32] are very powerful to calculate eigenvalues in just using a single long iteration. Nevertheless, if both eigenvalues and eigenvectors are required, those methods have to restart the iteration to extract eigenvectors as a long iteration excludes saving large intermediate vectors. In order to improve the efficiency, the iterative algorithms are often combined with other techniques such as the filter diagonalization method,[34,35] the restart approach,[36,37] and the spectral transform method.[38] The latter two techniques aim to do a short iteration to converge the results so that the working units for vectors are limited. There are several Lanczos-based spectral transform algorithms, e.g., see Refs. [2, 10, 39–45]. Recently, we also developed an efficient complex guided spectral transform Lanczos method for studying molecular bound and resonance states.[46]

On the other hand, the neural network method[47–51] has recently become a powerful technique in many fields, especially, in biosciences since the neural network was introduced by McCulloch and Pitts[47] in 1943. It has many attractive merits such as the extreme flexibility and multi-dimensional representation. The flexibility is attributed by the artificial multi-layer neural network structure[48,50] with arbitrary hidden layers[49] and the use of transform functions.[52] The multi-dimensional representations are also very compact so that neural network functions can interpolate high dimensional targets efficiently and accurately. The applications of neural network in chemical physics have grown quickly since early 1990s, e.g., see the recent excellent review.[51] For instance, one popular application is to fit high dimensional potential energy surfaces (PES) and electronic densities in electronic structure calculations.[51,53–55] In order to persist the permutation symmetry among identical atoms, Jiang and Guo[56] recently developed a permutationally invariant (PI) neural network PES method by employing a set of PI basis functions.[57–59] In addition, a few neural network-based methods have been developed for computing eigenvalues and eigenvectors of matrices.[60–64] For the eigenvalue problems in quantum mechanics, neural network functions are commonly used as a basis set for solving Schrödinger equations.[63,64] It is not surprising because of the merits of the flexibility and high dimensional representation of neural network functions. Manzhos and co-workers[64] showed that a small set of neural network functions are able to well converge the vibrational states of H_2O. However, one has to optimize the non-linear parameters appearing in the neural network functions.

So far, neural network based eigensolvers are still rather limited.[60–62] In this work, we will explore the combination of iterative diagonalization approach with a neural network structure for developing a new eigensolver, named the neural network iterative diagonalization method (NNiDM). The NNiDM algorithm aims to calculate both the eigenvalues and eigenvectors of complex symmetric or Hermitian matrices. It will be presented in Sec. 2. Some numerical examples will be illustrated in Sec. 3. Finally a short conclusion is summarized in Sec. 4.

2 Theory

In this section, we describe the neural network iterative diagonalization method. It is mainly developed from our recently developed complex guided spectral transform Lanczos (cGSTL) method[46] with the help of the thick restart technique[37] and multi-layered basis contraction approach.[13,24,45,65] The presentation will begin with a concise review on an artificial feed-forward neural network method that connects to this work. Without explicitly saying, the algorithm is illustrated with a complex symmetric matrix or an extended Hamiltonian $\tilde{H}$. For Hermitian matrices or Hamiltonian $\hat{H}$, some necessary but straightforward modifications are discussed in Sec. 2.2.2. Generally, we use two types of vectors $|\psi_i)$ and $|\psi_j>$. The vectors $|\psi_j>$ are used for a Hermitian matrix or Hamiltonian $\hat{H}$. They have the normal normalization condition $<\psi_j|\psi_j>=1$. On the other hand, the vectors $|\psi_i)$ are employed for a complex symmetric matrix or extended Hamiltonian $\tilde{H}$. They obey the c-product normalization condition[7,8] $(\psi_i|\psi_i) =<\psi_i^*|\psi_i>=1$, namely without the complex conjugation $(*)$ in integrals.

2.1 Artificial neural network

There are many types of neural networks in the literature.[50,51,66] A typical feed-forward neural network[51] is shown in Fig. 1. It is formed by four layers of artificial neurons (also called nodes).[48] The first and last layers are the input and output layers respectively. The middle ones are the hidden layers.[49] Each layer consists of one or a few nodes. They are represented by a node function y_i^l, where the superscript l and subscript i refer to the layer and node indices. In the first input layer the y_i^0 functions describe the information about input raw data (e.g. geometries) whereas the node function in the last layer gives the wanted properties such as potential energy surfaces, electronic densities, and wavefunctions.

In the feed-forward neural network, the l^{th} nodes are only determined from the previous $(l-1)^{th}$ nodes by transforming a shifted linear combination of node functions y_k^{l-1}, namely,

$$y_i^l = f_i^l(x_i^l) \tag{1}$$

with

$$x_i^l = b_i^{l,l-1} + \sum_{k=1}^{N^{l-1}} w_{i,k}^{l,l-1} y_k^{l-1} \tag{2}$$

where the expansion coefficients $w_{i,k}^{l,l-1}$ are the weights from the k^{th} node in layer $l-1$ to the i^{th} node in layer l. The shifting parameter $b_i^{l,l-1}$ is a bias for the i^{th} node in layer l from layer $l-1$. N^{l-1} is the node number in layer $l-1$. In Eq. (1) f_i^l is a transform function for the i^{th} node in layer l. Currently, the most widely used functions are linear (x), Gaussian $(exp(-x^2))$, and sigmoidal functions, e.g., $f(x) = [1+exp(-x)]^{-1}$ and hyperbolic tangent $(tanh(x))$. The transform functions with biases are used to confine the range of node functions so that they are well behavior.

Indeed a feed-forward neural network structure defines just a function linking the input data and target. Such a function may have no physical meaning but a mathematical form. This function can be evaluated by using a layer-by-layer propagation. In particular, it associates with many (often non-linear) parameters including the biases, weights, layer number, and node number in each layer. Therefore, a neural network function is ultra-flexible. It is also a multi-dimensional representation according to the nested neural network structure. For instance, the neural network function in Fig. 1 is written as

$$\mathcal{R} = f_1^3(x_1^3) = f_1^3\{b_1^{3,2} + \sum_{k=1}^{4} w_{1,k}^{3,2} f_k^2[b_k^{2,1} + \sum_{j=1}^{3} w_{k,j}^{2,1} f_j^1(b_j^{1,0} + \sum_{i=1}^{2} w_{j,i}^{1,0} y_i^0)]\}. \tag{3}$$

Finally the parameters in a neural network function can be determined by using an optimization method.[50] But one should pay more attention to the flexibility that may cause a overcomplete representation of data. Therefore, a testing and learning procedure is often adapted in optimization calculations.[50,51,66]

2.2 Neural network iterative diagonalization method

Our proposed neural network iterative diagonalization algorithm is illustrated in the architecture in Fig. 2, where each node in the first input layer and (m) hidden layers is described by a complex formally orthogonal Lanczos polynomial, i.e., $\mathcal{L}_i^l$ at node i in layer l. N^l is the number of nodes of the l^{th} hidden layer. k^l refers to the size of thick restart block in layer l (see below for details). In addition, the output M nodes are the calculated eigenstates including both eigenvalues z_n and eigenvectors (or wavefunctions) $|\psi_n(z_n))$. All parameters involved in the neural network will be determined on-the-fly during a series of recurrences. No non-linear optimization is required similar to the molecular dynamics optimization approach.

2.2.1 Network structure and node functions

Before discussing how to construct the nodes in Fig. 2, let's first define a complex guided spectral transform function $F(\tilde{H})$ of $\tilde{H}$. It is expressed in a series of complex formally orthogonal Lanczos (cFOL) polynomials $(\tilde{\mathcal{L}}_i(\tilde{H}))$,[46,67]

$$F(\tilde{H}) = \sum_{j=1}^{L_C} A_j(E_{ref})\tilde{\mathcal{L}}_j(\tilde{H}), \tag{4}$$

where the cFOL polynomials are obtained using the Lanczos-type recurrence

$$\tilde{\beta}_{j+1}\tilde{\mathcal{L}}_{j+1} = \tilde{H}\tilde{\mathcal{L}}_j - \tilde{\alpha}_j\tilde{\mathcal{L}}_j - \tilde{\beta}_j\tilde{\mathcal{L}}_{j-1}, \tag{5}$$

with

$$\tilde{\mathcal{L}}_0(\tilde{H}) = 0 \text{ and } \tilde{\mathcal{L}}_1(\tilde{H}) = 1.$$

Here $(\tilde{\alpha}_j, \tilde{\beta}_j)$ are the pre-defined Lanczos coefficients calculated with $\tilde{H}$ and a random initial vector $|\tilde{\psi}_1)$. They are calculated using a complex-symmetric Lanczos recursion,[4,8,9,32,42,68–70]

$$\tilde{\beta}_{j+1}|\tilde{\psi}_{j+1}) = \tilde{H}|\tilde{\psi}_j) - \tilde{\alpha}_j|\tilde{\psi}_j) - \tilde{\beta}_j|\tilde{\psi}_{j-1}), \tag{6}$$

with

$$\tilde{\beta}_1 = 0,\ \ |\tilde{\psi}_0) = 0. \tag{7}$$

Therefore, $\tilde{\alpha}_j = (\tilde{\psi}_j|\tilde{H}|\tilde{\psi}_j)$ and $\tilde{\beta}_{j+1} = (\tilde{\psi}_{j+1}|\tilde{H}|\tilde{\psi}_j)$ are the mean value and recursion coefficient of vector $|\tilde{\psi}_j)$, respectively. Those Lanczos coefficients need be calculated only once.

In Eq.(4) $A_j(E_{ref})$ are the expansion coefficients for a reference value E_{ref}. This value specifies the pivot of desired eigenstates. Normally, the expansion order L_C is small about tens. The expansion coefficients are determined by the guidance of an analytical function such as the retarded Green function[38,44,46] $(G^+(E_{ref}) = (\tilde{H} - E_{ref})^{-1})$, exponential[10,39,41] and hyperbolic tangent[71] functions. For more details how to obtain the coefficients, the reader can refer to Refs. [2,46] and references therein since they are well documented in the literature. Using the guided spectral transform function is mainly to (1) avoid the difficulty of analytical function-vector products; (2) dilate the spectrum near the reference value so that the wanted eigenstates can be quickly converged; and (3) transform the eigenvalues of interest into the largest or smallest ones in the transformed spectrum. For convenience, in this work, a single spectral transform function $F(\tilde{H})$ is used in all calculations.

Now, the neural network iterative algorithm to construct node functions can be written in a matrix form as

$$F(\tilde{H})\mathbf{L}^l_{N^l-1} = \mathbf{L}^l_{N^l}\mathbf{T}^l_{N^l-1} = \mathbf{L}^l_{N^l-1}\mathbf{T}^l_{N^l-1} + \beta^l_{N^l}\mathcal{L}^l_{N^l}e^T_{N^l-1} \tag{8}$$

starting with

$$\mathcal{L}^1_1 = \mathcal{L}^0_1 = 1 \text{ and } \mathcal{L}^1_0 = 0, \tag{9}$$

where $e^T_K = [0, 0, \cdots, 1]$ is a unit vector in the K^{th} dimension. The superscript "T" refers to the matrix transpose. $\mathbf{L}^l_K = [\mathcal{L}^l_1, \mathcal{L}^l_2, \cdots, \mathcal{L}^l_K]$ is a matrix with a size of K $(K \leq N^l)$ for the l^{th} layer node functions. The $\mathbf{T}^l_K$ matrices are given by a complex symmetric tridiagonal matrix

$$\mathbf{T}^l_K = \left[\begin{array}{ccccc} \alpha^l_1 & \beta^l_2 & & & \\ \beta^l_2 & \alpha^l_2 & \beta^l_3 & 0 & \\ & \beta^l_3 & \ddots & \ddots & \\ & 0 & \ddots & \ddots & \beta^l_K \\ & & & \beta^l_K & \alpha^l_K \end{array}\right], \text{ for } l = 1 \tag{10}$$

or a symmetric stripped tridiagonal one

$$\mathbf{T}^l_K = \left[\begin{array}{ccccccccc} \alpha^l_1 & & & & \beta^l_2 & & & & \\ & \alpha^l_2 & 0 & & \beta^l_3 & & & & \\ & 0 & \ddots & & \vdots & & 0 & & \\ & & & \alpha^l_{k^l} & \beta^l_{k^l+1} & & & & \\ \beta^l_2 & \beta^l_3 & \cdots & \beta^l_{k^l+1} & \alpha^l_{k^l+1} & \beta^l_{k+2} & & & \\ & & & & \beta^l_{k^l+2} & \alpha^l_{k^l+2} & \beta^l_{k^l+3} & & \\ & & 0 & & & \beta^l_{k^l+3} & \ddots & \ddots & \\ & & & & & & \ddots & \ddots & \beta^l_K \\ & & & & & & & \beta^l_K & \alpha^l_K \end{array}\right], \text{ for } l = 2, \cdots, m \tag{11}$$

where k^l is the size of restarted block for layer l. All β^l_1 are zero.

For the first hidden layer ($l = 1$), the elements in the $\mathbf{T}^l_K$ matrix in Eq. (10) are indeed Lanczos coefficients. They are produced using the complex guided spectral transform Lanczos method[46]

$$\beta^l_{j+1}|\psi^l_{j+1}) = F(\tilde{H})|\psi^l_j) - \alpha^l_j|\psi^l_j) - \beta^l_j|\psi^l_{j-1}), \tag{12}$$

where α^l_j and β^l_j are the complex mean value and residual of the j^{th} vector with respect to the spectral transform function $F(\tilde{H})$ respectively. They are written as

$$\alpha^l_j = (\psi^l_j|F(\tilde{H})|\psi^l_j),\ \beta^l_{j+1} = (\psi^l_{j+1}|F(\tilde{H})|\psi^l_j). \tag{13}$$

The recursion in Eq. (12) starts with an random initial vector, i.e., $|\psi^{l=1}_1) = |\phi_0)$.

However, for other hidden layers ($l = 2, 3, \cdots, m$), the iterations depend on the outcomes of proceeding layer ($l - 1$) as usual. The first k^l sets of (α^l_j, β^l_j) are determined by (see Eq. (24) below for details)

$$\begin{aligned} \alpha^l_j &= \lambda^{l-1}_j, & (14) \\ \beta^l_{j+1} &= \beta^{l-1}_{N^{l-1}}\mathbf{P}_{K'j},\ j = 1, 2, \cdots, k^l\ , & (15) \end{aligned}$$

where λ_j^{l-1} and $\mathbf{P}_j$ are the largest (or smallest, which depends on the spectral transform function $F(\tilde{H})$) k^l eigenvalues and eigenvectors of the $\mathbf{T}^{l-1}_{K'=N^{l-1}-1}$ matrix from the $(l-1)$ layer, that is,

$$\mathbf{T}^{l-1}_{K'}\mathbf{P}^{l-1} = \mathbf{P}^{l-1}\Lambda^{l-1} \tag{16}$$

with $diag(\Lambda^{l-1}) = [\lambda_1^{l-1}, \lambda_2^{l-1}, \cdots, \lambda_{k^l}^{l-1}]$. Here one should notice that $\mathbf{P}^{l-1}$ is a $K' \times k^l$ matrix with $(\mathbf{P}^{l-1})^T\mathbf{P}^{l-1} = \mathbf{I}$. By using the thick restart technique,[37] the other elements in the $\mathbf{T}^l_{K'}$ matrix in Eq. (11) are again determined by the complex spectral transform Lanczos method but starting with a given vector $|\psi^l_{k^l+1}) = |\psi^{l-1}_{N^{l-1}})$ at step k^l+1. They can be expressed as

$$\beta^l_{k^l+2}|\psi^l_{k^l+2}) = F(\tilde{H})|\psi^l_{k^l+1}) - \alpha^l_{k^l+1}|\psi^l_{k^l+1}) - |\varphi^l_{k^l}), \tag{17}$$

$$\beta^l_{j+1}|\psi^l_{j+1}) = F(\tilde{H})|\psi^l_j) - \alpha^l_j|\psi^l_j) - \beta^l_j|\psi^l_{j-1}),\ j \geq k^l+2, \tag{18}$$

with

$$\alpha^l_{k^l+1} = (\psi^l_{k^l+1}|F(\tilde{H})|\psi^l_{k^l+1}) \tag{19}$$

and

$$|\varphi^l_{k^l}) = \sum_{i=1}^{k^l} \beta^l_{i+1}|\psi^l_i), \tag{20}$$

where $|\psi^l_i)$ are the rotated Lanczos eigenvectors,

$$|\psi^l_i) = \sum_{j=1}^{N^{l-1}-1} \mathbf{P}^{l-1}_{ji}|\psi^{l-1}_j). \tag{21}$$

Accordingly, we have the node functions for layer l $(= 2, \cdots, m)$

$$\mathcal{L}^l_i = \sum_{j=1}^{N^{l-1}-1} \mathbf{P}^{l-1}_{ji}\mathcal{L}^{l-1}_j,\ i = 1, \cdots, k^l, \tag{22}$$

$$\mathcal{L}^l_{k^l+1} = \mathcal{L}^{l-1}_{N^{l-1}}. \tag{23}$$

The polynomial $\mathcal{L}^l_{k^l+1}$ adapts the last node function in the proceeding layer. Such a node is shown in red in Fig. 2. Also displayed in blue are the nodes obtained by Eq. (22) as they are obtained by a linear combination of a subset of the functions in the proceeding layer via a diagonalization approach. All other node functions in hidden layers are obtained via a Lanczos-type recursion as shown in Eq. (8).

To better understand the algorithm, let us multiply Eq. (8) by the $\mathbf{P}^l$ matrix from the right side. Then we have

$$\begin{aligned} F(\tilde{H})\mathbf{L}^l_{N^l-1}\mathbf{P}^l &= \mathbf{L}^l_{N^l-1}\mathbf{T}^l_{N^l-1}\mathbf{P}^l + \beta^l_{N^l}\mathcal{L}^l_{N^l}e^T_{N^l-1}\mathbf{P}^l \\ &= \mathbf{L}^l_{N^l-1}\mathbf{P}^l\Lambda^l + \mathcal{L}^l_{N^l}[\beta^l_{N^l}e^T_{N^l-1}\mathbf{P}^l] \end{aligned} \tag{24}$$

that exactly defines the first k^l iterations for the $(l+1)$ layer together with the node functions $\mathbf{L}^{l+1}_{k^l+1} = [\mathbf{L}^l_{N^l-1}\mathbf{P}^l, \mathcal{L}^l_{N^l}]$ and the leading $(k^l+1)\times(k^l+1)$ block of $\mathbf{T}^{l+1}_K$ in Eq. (11). Importantly, the operation above works as a filter projection because $\mathbf{P}$ is a rectangular matrix with $|\mathbf{P}^l(\mathbf{P}^l)^T| \leq \mathbf{I}$. As a result, the components outside the range of interest (approximately estimated by those selected eigenstates in Eq. (16)) are largely damped. Therefore, the layered iteration structure can enhance the efficiency of eigenstate calculations.

Finally, the last output layer nodes (eigen-pairs) are obtained by diagonalizing the perturbed matrix[46]

$$\tilde{H}'_\lambda = \tilde{H} + \lambda_s(-iW), \ \lambda_s \geq -1 \tag{25}$$

in a set of well converged basis set $\{|\psi'_1), |\psi'_2), \cdots, |\psi'_k)\}$ with $k \geq M$. They are eigenvectors of the unperturbed Hamiltonian, namely

$$\tilde{H}|\psi'_n) = z'_n|\psi'_n), \text{ and } |\psi'_n) = \mathcal{L}^m_n|\phi_0) \ . \tag{26}$$

Then the matrix representation of $\tilde{H}'_\lambda$ is written as

$$\tilde{\mathbf{H}}'_\lambda = \mathbf{D}_z - i\lambda_s\mathbf{W} \tag{27}$$

with

$$\mathbf{W}_{ij} = (\psi'_i|W|\psi'_j) \tag{28}$$

and the diagonal matrix $diag(\mathbf{D}_z) = (z_1^{'}, z_2^{'}, \cdots, z_k^{'})$. Therefore, the eigenvalue trajectories of desired states can be calculated by adjusting the NIP strength parameter λ_s. An optimal true eigenstate is defined by a cusp point.[7,8] Since the basis size is small, those diagonalizations can be done quickly. Here we have used the orthonormal property of Lanczos vectors or polynomials. The relationship is expressed as

$$|\psi_j^l) = \mathcal{L}_j^l|\phi_0) \text{ and } Tr(\mathcal{L}_i^l\mathcal{L}_j^l)_{\phi_0} = \delta_{ij}. \tag{29}$$

As $\mathcal{L}_j^l$ are polynomial functions of Hamiltonian, they commute, i.e. $[\mathcal{L}_j^l, \tilde{H}] = 0$. Furthermore, no trajectory searching is required for a Hermitian matrix or Hamiltonian as its eigenstates are independent of any NIP.

In short, the neural network iterative diagonalization algorithm is constructed by using some small matrix diagonalizations (having the k^l dimensions of restarted blocks) and two layered Lanczos iterations. The nested Lanczos recursion is used to carry out the spectral transform of the original matrix or Hamiltonian. The algorithm starts with a random vector, and the parameters in the neural network are determined dynamically without any non-linear optimization. The hidden layer propagation will stop at the node N^m of layer m if all the state residuals satisfy with the condition,

$$|\beta_{N^m}^m \mathbf{P}_{N^m-1,j}| \leq \epsilon, \text{ for } j = 1, 2, \cdots, M+q \ , \tag{30}$$

where ϵ is a small criterion (saying 10^{-8}) and q (usually less than four) is a number of additional states to be converged. Those states are employed to secure the convergence of wanted states in the trajectory optimization procedure. In particular, the NNiDM only requires the Hamiltonian-vector products so that it is very efficient for large sparse matrices. Therefore, the neural network iterative diagonalization method is general and problem-independent. As the core memory can be shared among the hidden layers, roughly, the requirement of core memory is determined by the storage of $(N_{max}^l + 2)$ vectors. N_{max}^l is the maximum value of N^l, and the number two refers to the two Lanczos vectors used to perform the guided spectral transform Lanczos in the inner layer iterations. That is, the memory requirement is limited in the algorithm. The NNiDM is also very simple to use. The input information only includes a few parameters N^l_{max}, q, M, and E_{ref} in addition to the calculations of matrix-vector products.

2.2.2 Orthogonality and modifications for Hermitian matrix

It is well known that the Lanczos vectors will lose their orthogonality after tens iterations due to the machine precision.[27,72] This can be prevented by using full reorthogonalization procedures such as that in the Arnoldi algorithm.[16] Alternatively, in this work, we use the selective and partial reorthogonalization (PRO) method of Simon.[73] It is a semi-local method. In the PRO approach, the new Lanczos vector is only reorthogonalized against those vectors with respect to which it is not orthogonal, within a numerical tolerance ε (about 10^{-12}). The PRO Lanczos algorithm is written (dropping out the layer index) as[73]

$$\beta^{'}_{j+1}|\psi^{'}_{j+1}) = F(\tilde{H})|\psi_j) - \alpha_j|\psi_j) - \beta_j|\psi_{j-1}) - |f^{'}_j) \tag{31}$$

and

$$\beta_{j+1}|\psi_{j+1}) = \beta^{'}_{j+1}|\psi^{'}_{j+1}) - \sum_{k\in\mathcal{O}(j)} \beta^{'}_{j+1}(\psi_k|\psi^{'}_{j+1})|\psi_k) - |f_j), \tag{32}$$

where $\mathcal{O}(j) = \{k; |\omega_{j+i,k}| > \varepsilon\}$ is a set of indices for applying the reorthogonalization procedure. The values $\omega_{j,k}$ $(k = 1, \cdots, j-1)$ are the overlap matrix elements $\omega_{j,k} = (\psi_j|\psi_k)$ of the Lanczos vectors. In the PRO method, explicit calculations of the overlap matrix is avoided. Instead they are obtained via the following simple recurrence,[73]

$$\begin{aligned}
\omega_{k,k} &= 1, \text{ for } k = 1, \cdots, j \\
\omega_{k,k-1} &= \epsilon_k, \text{for } k = 2, \cdots, j \\
\beta_{k+1}\omega_{j+1,k} &= \beta_{k+1}\omega_{j,k+1} + (\alpha_k - \alpha_j)\omega_{j,k} + \beta_k\omega_{j,k-1} - \\
&\quad \beta_j\omega_{k+1,k} + (\psi_j|f_k) - (\psi_k|f_j), \text{ for } 1 \le k < j \\
\omega_{j,k+1} &= \omega_{k+1,j}
\end{aligned} \tag{33}$$

Here $\omega_{k,0} = 0$ and $\epsilon_k = (\psi_k|\psi_{k-1})$. In Eqs. (31) - (33), the vectors $|f'_j)$ and $|f_j)$ account for the local roundoff errors at the j^{th} step. In practice we have taken the local roundoff errors as zero since they are much smaller than ε. The PRO Lanczos method was used by us[2,41,74,75] for quantum reactive scattering calculations. The outcome is highly satisfying. The algorithm can not only avoid the spurious state problem but also resolve the degenerate states if any.

Until now the neural network iterative algorithm has been described for complex symmetric matrices. Actually the NNiDM protocol is also true for Hermitian Hamiltonian after some simple modifications. The essential modifications are as follows:

(1) The normalizations $(\psi_j|\psi_k)$ are replaced with $< \psi_j|\psi_k >$.

(2) All vectors $|\psi^l_j)$ in the Lanczos iterations and eigenvector constructions are replaced with $|\psi^l_j >$.

(3) All quantities are determined by the Hamiltonian $\hat{H}$ instead of $\tilde{H}$.

For instance, the cGSTL algorithm in Eq. (12) is changed to a real guided spectral transform Lanczos recurrence[14,45]

$$\beta^l_{j+1}|\psi^l_{j+1} >= F(\hat{H})|\psi^l_j > -\alpha^l_j|\psi^l_j > -\beta^l_j|\psi^l_{j-1} >, \tag{34}$$

and the Lanczos eigenvectors in Eq. (21) are replaced with

$$|\psi^l_i >= \sum_{j=1}^{N^{l-1}-1} \mathbf{P}^{l-1}_{ji}|\psi^{l-1}_j > . \tag{35}$$

As a result both the $\mathbf{T}^l_K$ matrices in Eqs. (10) and (11) are real symmetric for the Hermitian cases. Since the changes are straight-forward, it won't be repeated here.

2.3 Practical implementation with layered basis contraction

Although the neural network iterative method only requires Hamiltonian-vector products, the product calculations strongly rely on the choice of basis set and/or coordinates that determine the structure and sparseness of Hamiltonian matrix. In this subsection we will discuss a method on how to efficiently carry out the Hamiltonian-vector products for high dimensional problems. Here, we will focus on the calculations of vibrational spectroscopy (and/or quantum scattering dynamics via resonance states) of polyatomic molecules. For large polyatomic molecules beyond four atoms, the resulting direct product basis set is usually too huge to be feasible for directly doing the Hamiltonian-vector actions in such a basis set. In order to overcome this difficulty, many basis contraction methods[10–14, 24, 45, 65, 76–79] have been proposed. In particular, the multi-layered basis contraction schemes[13, 24, 65] are very attractive. In this work, we will use our recently developed multi-layer Lanczos iteration approach[45] by extending it to the extended Hamiltonian $\tilde{H}$. Since the approach has been well documented for the Hermitian Hamiltonian, only a brief description will be presented here. The reader can see Ref. [45] for more details.

The molecular vibrational Hamiltonian of system is represented in a set of orthogonal polyspherical coordinates.[80–84] For a molecule with N atoms, it has $(3N-6)$ internal variables as shown in Fig. 3. They are defined by $(N-1)$ radial coordinates ($\mathbf{R} = \{r_1, r_2, \cdots, r_{N-1}\}$) and $(2N-5)$ angular variables ($\mathbf{Q}$) including $(N-2)$ polar angles $\{\theta_i\}$ and $(N-3)$ azimuthal angles $\{\varphi_j\}$ of the radial vectors in the body-fixed (BF) frame. Then the Hamiltonian with the total angular momentum $J=0$ is written as[24, 83]

$$\tilde{H} = \hat{T}_R(\mathbf{R}) + \tilde{H}_Q(\mathbf{Q};\mathbf{R}) - iW_R \qquad (36)$$

with

$$\tilde{H}_Q(\mathbf{Q};\mathbf{R}) = \hat{T}_Q(\mathbf{Q};\mathbf{R}) + V(\mathbf{Q},\mathbf{R}) - iW_Q, \qquad (37)$$

where $-iW_R$ and $-iW_Q$ are the $R-$ and $Q-$dependent NIPs. Usually W_Q is null as molecules often dissociate only along the $\mathbf{R}$ coordinates. $V(\mathbf{Q},\mathbf{R})$ is the potential energy surface of the system. $\hat{T}_R(\mathbf{R})$ is the kinetic energy operator in the radial coordinates. It is given by a summation of one dimensional (1D) Hamiltonian $\hat{h}(r_i)$, i.e.,[45]

$$\hat{T}_R(\mathbf{R}) = \sum_{i=1}^{N-1} [\hat{h}(r_i) - V_0(r_i)], \qquad (38)$$

$$\hat{h}(r_i) = -\frac{\hbar^2}{2\mu_i r_i^2}\frac{\partial}{\partial r_i} r_i^2 \frac{\partial}{\partial r_i} + V_0(r_i). \tag{39}$$

Here $V_0(r_i)$ is a 1D reference potential in r_i with its associated reduced mass μ_i. $\hat{T}_Q(\mathbf{Q};\mathbf{R})$ is the kinetic operator in the angular coordinates. Generally, it can be partitioned as

$$\hat{T}_Q(\mathbf{Q};\mathbf{R}) = \sum_{i=1}^{N-1} \frac{1}{2\mu_i r_i^2}\hat{T}_Q^{(i)}(\mathbf{Q}), \tag{40}$$

where $\hat{T}_Q^{(i)}(\mathbf{Q})$ are R-independent. That is, $\hat{T}_Q$ only parametrically depends on $\mathbf{R}$ through pre-factors without any crossed partial derivative terms between the two coordinate groups.

The $\tilde{H}-$vector products are carried out in a contracted grid/diabatic basis functions in which the Lanczos vector $|v_k)$ is expressed as

$$|v_k) = \sum_{m,\alpha} C^k_{m,\alpha} f_m(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)|\mathbf{R}_\alpha>, \tag{41}$$

where $|\mathbf{R}_\alpha >= \Pi_{i=1}^{N-1}|r_{\alpha_i}>$ refer to the direct-product PO-DVR (potential optimized-discrete variable representation) basis functions in $\mathbf{R}$ with α being a collective DVR index. The 1D PO-DVRs are calculated using the lowest eigenstates of the Hamiltonian $\hat{h}(r)$ in Eq. (39). The direct-product basis functions are further contracted by discarding those PO-DVRs where the minimum potential energies in their corresponding $\mathbf{R}_\alpha$ sectors are larger than a threshold value (V_{th}). In addition, $f_m(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)$ (associated with eigenvalues $\mathcal{E}_m^0$) are the vibrationally diabatic basis functions in $\mathbf{Q}$. They will be calculated as discussed below.

By using this kind of basis set, the action of $\tilde{H}$ on vector $|v_k)$ is written as

$$\begin{aligned}\tilde{H}|v_k) \;=\; & \sum_{n,\beta}\sum_{m,\alpha}\Big\{< \mathbf{R}_\beta|\hat{T}_R(\mathbf{R})|\mathbf{R}_\alpha > \delta_{nm} + [\mathcal{E}_m^0 - iW_R(\mathbf{R}_\alpha)]\delta_{nm}\delta_{\beta\alpha} \\ & +(f_n|\Delta\hat{H}_Q(\mathbf{R}_\alpha)|f_m)\delta_{\beta\alpha}\Big\} C^k_{m,\alpha} f_n(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)|\mathbf{R}_\beta >,\end{aligned} \tag{42}$$

with

$$< \mathbf{R}_\beta|\hat{T}_R(\mathbf{R})|\mathbf{R}_\alpha > \;=\; \sum_{i=1}^{N-1}\Big\{< r_{\beta_i}|\hat{h}(r_i)|r_{\alpha_i} > -V_0(r_{\alpha i})\delta_{\beta_i\alpha_i}\Big\} \times \Pi_{j\neq i}^{N-1}\delta_{\beta_j\alpha_j}, \tag{43}$$

$$(f_n|\Delta\hat{H}_Q(\mathbf{R}_\alpha)|f_m) \;=\; \sum_{i=1}^{N-1}\Delta I_{\alpha i0}(r_{\alpha i})(f_n|\hat{T}_Q^{(i)}|f_m) + (f_n|\Delta V(\mathbf{Q};\mathbf{R}_\alpha)|f_m), \tag{44}$$

and

$$\Delta V(\mathbf{Q};\mathbf{R}_\alpha) = V(\mathbf{Q};\mathbf{R}_\alpha) - V(\mathbf{Q};\mathbf{R}_0^V), \tag{45}$$

$$\Delta I_{i0}(r_i) = \frac{1}{2\mu_i r_i^2} - \frac{1}{2\mu_i r_{i0}^2}, \tag{46}$$

where $\Delta I_{i0}(r_i)$ are the diagonal pre-factor matrix elements in DVR. $(f_n|\Delta V(\mathbf{Q};\mathbf{R}_\alpha)|f_m)$ are the potential residual matrix elements whereas $(f_n|\hat{T}_Q^{(i)}|f_m)$ are the $R-$independent matrix elements of the kinetic energy operators in $\mathbf{Q}$.

In Eq. (41) the diabatic basis functions f_m are formed by the lowest eigenstates of a reference reduced-dimension Hamiltonian $\tilde{H}_Q^0(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)$ in $\mathbf{Q}$, namely

$$\tilde{H}_Q^0(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V) f_m(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V) = \mathcal{E}_m^0 f_m(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V), \tag{47}$$

with

$$\tilde{H}_Q^0(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V) = \hat{T}_Q(\mathbf{Q};\mathbf{R}_0) + V(\mathbf{Q};\mathbf{R}_0^V) - iW_Q, \tag{48}$$

where $\mathbf{R}_0$ are the radial references in the kinetic energy operator. Usually, they are constant as $\{r_{i0}\}$. $\mathbf{R}_0^V$ are the references in the potential energy surface, and may be dependent of $\mathbf{Q}$ for a better reference potential.[85] The eigenvalue problem in Eq. (47) will be solved by using the neural network iterative diagonalization method. Here, we need both eigenvalues and eigenvectors so that the NNiDM is favorite.

In the NNiDM calculations, the required $\tilde{H}_Q^0(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)-$vector products are performed in a non-direct product FBR (finite basis representation) basis set $\{|l>\}$ that is used to properly deal with the singularities in polar angles.[24,83] By using a Lanczos vector $|v_k^{'}) = \sum_l C_l^k |l>$ in FBR, the Hamiltonian-vector action is written as

$$\tilde{H}_Q^0|v_k^{'}) = \sum_l \sum_{i=1}^{N-1} C_l^k \frac{1}{2\mu_i r_{i0}^2} \hat{T}_Q^{(i)}(\mathbf{Q})|l> + \sum_{l'}\sum_l\sum_\gamma \mathbf{U}^\dagger_{l'\gamma}[V(\mathbf{Q}_\gamma;\mathbf{R}_0^V) - iW_Q(\mathbf{Q}_\gamma)]\mathbf{U}_{\gamma l} C_l^k |l'>, \tag{49}$$

where $\mathbf{U}$ is the collocation matrix between the FBR and DVR ($\{|\gamma>\}$) basis sets, i.e.

$$|\gamma> = \sum_l \mathbf{U}_{\gamma l}|l> . \tag{50}$$

Here, l and γ are the collective FBR and DVR indices with a dimension of $2N-5$, respectively. In terms of Eq. (50), we have the dual representation

$$f_m(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)=\sum_l P_{lm}|l>=\sum_\gamma X_{\gamma m}|\gamma>,\ m=1,2,\cdots,N_{diab} \tag{51}$$

$$X_{\gamma m}=\sum_l \mathbf{U}_{\gamma l}P_{lm}, \tag{52}$$

for the diabatic basis functions. P_{lm} and $X_{\gamma m}$ are the eigenvector coefficients of f_m in FBR and DVR respectively. The DVR representation is good for basis contraction.[45]

Although the detailed formulae of $\hat{T}_Q^{(i)}$ and $\mathbf{U}$ depend on the definition of $\mathbf{Q}$, usually, the $\hat{T}_Q^{(i)}-|l>$ products are analytically computed on-the-fly in FBR. They are very simple and easy as discussed below. In practice, the last potential term in Eq. (49) is often carried out using a series of small sequential matrix-vector products in one-by-one degree via a pseudo-spectral transformation technique.[86–88] For penta-atomic molecules, both $\hat{T}^{(i)}$ and $\mathbf{U}$ with their actions on vectors are given in Ref. [45].

Generally, for polyatomics with N atoms, according to the convention of the orthogonal polyspherical coordinates in Fig. 3, the Q-dependent kinetic operator parts in Eq. (40) are given by[14]

$$\begin{aligned}\hat{T}_Q^{(i)}(\mathbf{Q}) &= \hat{j}_i^2,\ i=1,\cdots,N-2, &(53)\\ \hat{T}_Q^{(N-1)}(\mathbf{Q}) &= \sum_{i=1}^{N-2}\hat{j}_i^2+\sum_{i<k=1}^{N-2}(2\hat{j}_{iz}\hat{j}_{kz}+\hat{j}_{i+}\hat{j}_{k-}+\hat{j}_{i-}\hat{j}_{k+}), &(54)\end{aligned}$$

where $\hat{j}_i$, $\hat{j}_{iz}$ and $\hat{j}_{i\pm}$ are the angular momentum, its Z projection and the ladder operators respectively.[89] Without external field, a molecule always persists the inversion symmetry (labeled by its parity p). Therefore, a symmetrically adapted FBR (SA-FBR) is usually employed in the angular coordinates. For instance, it is taken as[14,82]

$$|l>=\begin{cases}\frac{1}{\sqrt{2(1+\delta_{0n})}}\left\{|j_1j_2n>+(-1)^p|j_1j_2-n>\right\}, & \text{for tetra-atomics}\\ \frac{1}{\sqrt{2(1+\delta_{0n}\delta_{0m}\delta_{0k})}}\left\{|j_1j_2j_3j_4nmk>+(-1)^p|j_1j_2j_3j_4-n-m-k>\right\}, & \text{for hexa-atomics}\end{cases} \tag{55}$$

with the orthonormal spherical harmonic basis functions $|jm>$, i.e.,[89]

$$< \theta\varphi|jm >= \Theta_j^m(\theta)\frac{1}{\sqrt{2\pi}}e^{im\varphi}, \tag{56}$$

where $\Theta_j^m(\theta)$ are the normalized associated Legendre polynomials. These functions are the common eigenstates of both $\hat{j}^2$ and $\hat{j}_z$, namely,

$$\begin{aligned} \hat{j}^2|jm > &= j(j+1)\hbar^2|jm >, & (57)\\ \hat{j}_z|jm > &= m\hbar|jm > . & (58)\end{aligned}$$

For the ladder operators, we have the relationship,

$$\hat{j}_{\pm}|jm >= [j(j+1) - m(m \pm 1)]^{1/2}\hbar|jm \pm 1 > . \tag{59}$$

Therefore the T_Q-FBR basis products can be easily computed by using Eqs. (57)-(59). The detailed expressions are given in Refs. [82] for tetra-atomics, [83] for penta-atomics, and [14] for hexa-atomics etc.

Furthermore, the corresponding collocation matrix in Eq. (49) is given by

$$\mathbf{U} = \begin{cases} CST_{1D}\mathbf{\Theta}_{\theta_2}^{(n)}\mathbf{\Theta}_{\theta_1}^{(n)}, & \text{for tetra-atomics} \\ CST_{3D}\mathbf{\Theta}_{\theta_4}^{(n+m+k)}\mathbf{\Theta}_{\theta_3}^{(k)}\mathbf{\Theta}_{\theta_2}^{(m)}\mathbf{\Theta}_{\theta_1}^{(n)}, & \text{for hexa-atomics}\end{cases} \tag{60}$$

where CST_{1D} and CST_{3D} is the one- and three-dimensional cosine/sine transformation in the corresponding azimuthal angles while the $\Theta_\theta^{(k)}$ are the one-dimensional collocation matrices in θ. They are defined by the normalized associated Legendre polynomials in Eq. (56).[82, 83, 90]

In summary, two sequential neural network iterative diagonalizations are carried out for computing vibrational energies and wavefunctions of polyatomic molecules. One is used to obtain the vibrationally diabatic basis functions f_m with the reduced-dimensional Hamiltonian $\tilde{H}_Q^0(\mathbf{Q};\mathbf{R}_0,\mathbf{R}_0^V)$. The other is employed to solve the whole eigenvalue problem of system with $\tilde{H}$ but in a compact contracted basis set. The use of the contracted basis set can substantially reduce the requirement of core memory in addition to speeding up the calculations. The two NNiDM calculations are done sequentially so that they can share the same core memory. Usually, the memory requirement is determined by the calculations of the diabatic functions f_m.

3 Applications

In this section we will discuss two applications of the neural network iterative diagonalization method to calculate the bound and resonance states of HO_2 and the vibrational states of CH_4. Our main goal is to illustrate the applicability for different types of molecules and the performance of the algorithm.

3.1 Bound and resonance states of HO_2

HO_2 is a benchmark molecule with a deep potential well for quantum dynamics calculations.[32,46,91–93] In this application, it is employed to study the bound and resonance states of HO_2, and mainly to examine the accuracy and efficiency of the NNiDM algorithm. In the calculations, we adapted the same atom-diatom Jacobi coordinates, potential energy surface (DMBE IV),[94] basis set and parameters as in previous works.[32,46] Some results are listed in Table 1, together with a comparison with previous theoretical results based on the same potential energy surface. They are in good agreement. Therefore, the NNiDM is an accurate method.

More interesting is the performance of the NNiDM algorithm. Figure 4 shows a comparison of convergence for calculating the 400 (M) lowest bound and resonance states of HO_2 with $q = 1$. A typical error evolution is displayed in Fig. 4(a) for a given maximum node number $N^l_{max} = 1000$. One can see that the 400 eigenstates can be well converged with five hidden layers. The total number of $F(\tilde{H})-$vector products is about 2960, which often determines the efficiency of the algorithm. In general, the smaller the maximum node parameter N^l_{max} is, the larger the hidden layers are needed to converge results. For instance, 28 hidden layers are required to obtain those 400 eigenstates. In contrast, the number of $F(\tilde{H})-$vector products gradually decreases with N^l_{max} increasing as shown in Fig. 4(b). But a large N^l_{max} value demands more core memory. According to this numerical study, the suggested parameter is $N^l_{max} \in [M/2, M]$ for a good balance between the efficiency and core memory requirement, where M is the number of wanted eigenstates. The parameter $q \leq 4$ (i.e. the number of additional restarted vectors) is enough. It is mainly employed to take into account of the degeneracy of the highest and/or smallest eigenstates of interest.

Three typical node functions $\mathcal{L}^l_j$ are shown in Fig. 5. They are strongly oscillating in energy. However, the node functions roughly work as a step function having small amplitudes toward high energies. The onset of step threshold of $\mathcal{L}^l_j$ slightly shifts upward as their node numbers j increase. Therefore, the node functions can be considered as an energy path that filters out those high energy components of vectors.

This numerical example also shows that the NNiDM algorithm is more efficient than the most popular Chebyshev-based filter diagonalization (FD)[92] and Lanczos-based FD[32] methods according to the total number of Hamiltonain-vector products for converging the same eigenstates. It is comparable with the cGSTL algorithm but only demands about one third of core memeory required in cGTSL.[46] Therefore, the NNiDM algorithm is very attractive for studying large molecular systems. In the next subsection, we will give such an example.

3.2 Vibrational states of CH_4

Methane (CH_4) is another benchmark molecule for the study of molecular spectroscopy, for instance, see Refs. [45,95–98] and references therein. Although CH_4 is a rigid five-atom molecule, it is very challenging to accurately calculate its vibrational states due to the strong couplings among the degenerate bend and stretch modes. In this subsection, we will present an application to calculating the vibrational states of methane based on the recently refined WC potential energy surface[97] of the *ab initio* SP(T8) surface of Schwenke and Partridge.[95] Here we will examine the performance of the NNiDM algorithm implemented with the layered basis contraction approach.

In the calculations, the vibrational Hamiltonian is expressed in the orthogonal (4+1) Radau coordinates, i.e., the "Icd=2" set in the PetroVib program.[83] As discussed in Sec. 2.3 a combined FBR/DVR basis set is used. Ten PO-DVR functions are used for each radial coordinate. The non-direct product FBR in the angular variables are formed by the largest quantum number $j_{max} = 28$. The symmetrically adapted SA-FBR/DVR basis with respect to inversion symmetry is used. Those parameters give a primitive DVR basis size of 3.96×10^7 (or a basis set of 5.65×10^6 FBR functions) in $\mathbf{Q}$ that leads to a basis set of 3.96×10^{11} in total. 480 diabatic basis functions f_m are employed. They are calculated by the neural network iterative diagonalization method with $N^l_{max} = 800$ and $q = 3$. For the calculations of whole system, the used parameters are $N^l_{max} = 200$ and $q = 3$ for the $M = 120$ lowest vibrational states. The potential threshold value is set at $V_{th} = 3.25\,\mathrm{eV}$ in the full dimensional DVR basis contraction while the criterion used for the DVR contraction of f_m is $\epsilon = 10^{-8}$. Those parameters can well converge the vibrational energies up to 6600 cm^{-1} within 0.05 cm^{-1}. The reference radial coordinates in the kinetic energy operators are $2.0524108\,\mathrm{a}_0$ for r_{i0} with its associated mass $\mu_i = 1.007825035\,\mathrm{amu}$ relative to $m_{\mathrm{C}} = 12.0\,\mathrm{amu}$.

Table 2 lists the calculated energy levels of CH_4 together with a comparison with experimental and previous theoretical results,[97,99–105] where the energy levels are labeled as "$(v_1v_3)(v_2v_4)$symmetry" following the polyads. $v_1(\mathrm{A}_1)$ and $v_3(\mathrm{F}_2)$ are the stretch modes of CH_4 while $v_2(\mathrm{E})$ and $v_4(\mathrm{F}_2)$ are the bend ones. As one can see, they are in excellent agreement. The refined WC surface is very accurate up to about $5800\,\mathrm{cm}^{-1}$ (most likely up to $7000\,\mathrm{cm}^{-1}$). As discussed by Wang and Carrington,[97] those vibrational states having a large difference with experiments should be checked experimentally. In the NNiDM calculations, the 480 vibrationally diabatic basis functions are obtained with four hidden layers (800, 800, 800, 640) while the 120 eigenstates of CH_4 are computed with five hidden layers (200, 200, 200, 200, 160). It was found that the NNiDM algorithm shows a better efficiency for the Hermitian matrices than that for the complex matrices in HO_2 although such a comparison is not fair owing to the difference of matrix structures. Importantly, by incorporating with the layered basis contraction procedure, the neural network iterative diagonalization method requires much less core memory than the two-layer Lanczos algorithm.[24] These results also imply that the NNiDM algorithm with the layered basis contraction technique is capable of studying larger polyatomic molecules.

4 Conclusions

We have developed a novel neural network iterative diagonalization method (NNiDM) for studying eigenvalues and eigenvectors of large sparse complex symmetric or Hermitian matrices. The algorithm has a multi-layered feed-forward neural network structure where the node functions are iteratively determined by the complex (or real) guided spectral transform Lanczos (cGSTL) method[46] and small matrix diagonalizations in terms of the thick restart technique.[37] The loss of orthogonality of Lanczos vectors is prevented by using the partial reorthogonalization (PRO) method of Simon.[73] The construction of node functions starts with a random vector, and only requires the action of matrix or Hamiltonian on vectors without any non-linear optimization. The NNiDM is capable of computing interior eigenstates in dense spectrum regions with the help of the guided spectral transform technique.[38,41,46] In particular, the NNiDM algorithm is a universal eigensolver for complex symmetric or Hermitian matrices.

By extending the multi-layer Lanczos iteration approach[45] to the complex matrix or extended Hamiltonian systems, we also described a general method on how to efficiently evaluate the action of Hamiltonian on vectors in orthogonal polyspherical coordinates for studying the vibrational spectroscopy and quantum scattering of polyatomic molecules. Since the calculations are carried out in a set of contracted basis functions via a reduced-dimensional manner, it allows us to investigate large polyatomic systems. The approach in Sec. 2.3 is suitable for any polyatomic molecule. But the computational ability will be limited by the calculations of the vibrationally diabatic basis functions f_m having a dimension of $2N-5$ in spite of that this value is much smaller than the full dimension $3N-6$. Therefore, future work would be very helpful to overcome this bottleneck.

The NNiDM algorithm has been applied for calculating energies, widths, and wavefunctions of two typical molecules HO_2 and CH_4 for numerical demonstrations. Results show that it is an efficient algorithm like other advanced iterative diagonalization methods but this algorithm also produce eigenvectors easily with the help of limited core memory requirement. Therefore, the neural network iterative diagonalization method is very attractive if both eigenvalues and eigenvectors are needed as in the layered basis contraction scheme, for instance, the CH_4 example. Compared with the two-layer Lanczos method,[24] the NNiDM algorithm only requires about one third of core memory without the loss of efficiency. Thus, this new alogrithm is very useful for studying the eigenstates of larger molecules. In addition, the key parameter N^l_{max} (the maximum node number in layer l) is suggested as a value between $M/2$ and M for optimizing both the efficiency and the core memory requirement, where M is the number of wanted eigenstates. Within this range, the performance of the NNiDM algorithm is less sensitive.

Acknowledgments

This work was performed at Brookhaven National Laboratory under Contracts No. DE-AC02-98CH10886 with the U.S. Department of Energy, Office of Science, and supported by its Division of Chemical Sciences, Geosciences, and Biosciences within the Office of Basic Energy Sciences. It also used the resource at the National Energy Research Scientific

Computing Center (NERSC) under Contract No. DE-AC02-05CH11231. The author thanks Dr. Xiao-Gang Wang for providing the Fortran code of the WC potential energy surface of methane.

References

[1] V. Barone, M. Biczysko, and J. Bloino, *Phys. Chem. Chem. Phys.* 2014, **16**, 1759.

[2] G. Nyman and H.-G. Yu, *Int. Rev. Phys. Chem.* 2013, **32**, 39.

[3] A.G. Csazar, C. Fabri, T. Szidarovsky, E. Matyus, T. Furtenbacher, and G. Czako, *Phys. Chem. Chem. Phys.* 2012, **14**, 1085.

[4] N. Moiseyev, *Non-Hermitian Quantum Mechanics* (Cambridge University Press, Cambridge, 2011).

[5] H.-D. Meyer, F. Gatti, and G. A. Worth eds, *Multidimensional Quantum Dynamics: MCTDH Theory and Applications* (Wiley-VCH, Weinheim, Germany, 2009).

[6] J.M. Bowman, T. Carrington Jr., and H.-D. Meyer, *Mol. Phys.* 2008, **106**, 2145.

[7] G. Jolicard, C. Leforestier, and E.J. Austin, *J. Chem. Phys.* 1988,**88**, 1026.

[8] K.F. Milfeld and N. Moiseyev, *Chem. Phys. Lett.* 1986, **130**, 145.

[9] H.O. Karlsson, *J. Phys. B* 2009, **42**, 125205.

[10] T. Hammer and U. Manthe, *J. Chem. Phys.* 2012, **136**, 054105.

[11] M. Schröder, F. Gatti, and H.-D. Meyer, *J. Chem. Phys.* 2011, **134**, 234307.

[12] O. Vendrell, F. Gatti, and H.-D. Meyer, *J. Chem. Phys.* 2007, **127**, 184303.

[13] X.-G. Wang and T. Carrington Jr., *J. Chem. Phys.* 2008, **129**, 234102.

[14] H.-G. Yu, *J. Chem. Phys.* 2004, **120**, 2270.

[15] B.N. Parlett, *The Symmetric Eigenvalue Problem* (SIAM, Philadelphia, 1998).

[16] Y. Saad, *Numerical Methods for Large Eigenvalue Problems, 2nd Ed.* (Manchester Unversity Press, Manchester, UK, 2011).

[17] W. Chen and B. Poirier, *J. Theor. Comput. Chem.* 2010, **9**, 825.

[18] H. Köppel, L.S. Cederbaum, and W. Domcke, *J. Chem. Phys.* 1982, **77**, 2014.

[19] C. Iung and C. Leforestier, *J. Chem. Phys.* 1995, **102**, 8453.

[20] C. Iung, C. Leforestier, and R.E. Wyatt, *J. Chem. Phys.* 1993, **98**, 6722.

[21] V.A. Mandelshtam and H.S. Taylor, *J. Chem. Phys.* 1995, **102**, 7390.

[22] V.A. Mandelshtam and H.S. Taylor, *J. Chem. Phys.* 1997, **106**, 5085.

[23] P. Pendergast, Z. Darakjian, E.F. Hayes, and D.C. Sorensen, *J. Compt. Phys.* 1994, **113**, 201.

[24] H.-G. Yu, *J. Chem. Phys.* 2002, **117**, 8190.

[25] X.-G. Wang and T. Carrington Jr., *J. Chem. Phys.* 2004, **121**, 2937.

[26] R.E. Wyatt, *Adv. Chem. Phys.* 1989, **73**, 231.

[27] C. Lanczos, *J. Res. Nat. Bur. Stand.* 1950, **45**, 255.

[28] D. Xie, R. Chen, and H. Guo, *J. Chem. Phys.* 2000, **112**, 5263.

[29] R. Chen and H. Guo, *J. Chem. Phys.* 1996, **105**, 1311.

[30] R. Chen and H. Guo, *J. Compt. Phys.* 1997, **136**, 494.

[31] H.-G. Yu and S.C. Smith, *Ber. Bunsenges. Phys. Chem.* 1997, **101**, 400.

[32] H.-G. Yu and S.C. Smith, *Chem. Phys. Lett.* 1998, **283**, 69.

[33] S.K. Gray and G.G. Balint-Kurti, *J. Chem. Phys.* 1998, **108**, 950.

[34] D. Neuhauser, *J. Chem. Phys.* 1990, **93**, 2611.

[35] M.R. Wall and D. Neuhauser, *J. Chem. Phys.* 1995, **102**, 8011.

[36] D. Calvetti, L. Reichel, and D.C. Sorensen, *Electr. Transact. Numer. Anal.* 1994, **2**, 1.

[37] K. Wu and H. Simon, *SIAM J. Matrix Anal. Appl.* 2001, **22**, 602.

[38] T. Ericsson and A. Ruhe, *Math. Comput.* 1980, **35**, 1251.

[39] F. Webster, P.J. Rossky, and R.A. Friesner, *Comp. Phys. Comm.* 1991, **63**, 494.

[40] H. Kono, *Chem. Phys. Lett.* 1993, **214**, 137.

[41] H.-G. Yu and G. Nyman, *J. Chem. Phys.* 1999, **110**, 7233.

[42] C. Leforestier, K. Yamashita, and N. Moiseyev, *J. Chem. Phys.* 1995, **103**, 8468.

[43] B. Poirier and J. Carrington Jr., *J. Chem. Phys.* 2002, **116**, 1215.

[44] R.E. Wyatt, *Phys. Rev.* 1995, **E51**, 3643.

[45] H.-G. Yu, *J. Chem. Phys.* 2015, **142**, 044106.

[46] H.-G. Yu, *J. Chem. Phys.* 2014, **141**, 244114.

[47] W. McCulloch and W. Pitts, *Bull. Math. Biophys.* 1943, **5**, 115.

[48] F. Rosenblatt, *Psych. Rev.* 1958 **65**, 386.

[49] M. Minsky and S.A. Papert, *Perceptrons* (MIT Press, Cambridge MA, 1969).

[50] S. Haykin, *Neural Networks and Learning Machines 3rd Edn* (Pearson, New York, 2009).

[51] J. Behler, *J. Phys.: Condens. Matter* 2014, **26**, 183001.

[52] W.A. Little, *Math. Biosci.* 1974, **19**, 101.

[53] T.B. Blank, S.D. Brown, A.W. Calhoun, and D.J. Doren, *J. Chem. Phys.* 1995, **103**, 4129.

[54] S. Manzhos, X.G. Wang, R. Dawes, and T. Carrington Jr., *J. Phys. Chem. A* 2006, **110**, 5295.

[55] S. Manzhos and T. Carrington Jr., *J. Chem. Phys.* 2006, **125**, 084109.

[56] B. Jiang and H. Guo, *J. Chem. Phys.* 2013, **139**, 054112.

[57] B.J. Braams and J.M. Bowman, *Int. Rev. Phys. Chem.* 2009, **28**, 577.

[58] J.N. Murrell, S. Carter, S.C. Farantos, P. Huxley, and A.J.C. Varandas, *Molecular Potential Energy Functions* (Wiley, Chichester, 1984).

[59] A.J.C. Varandas and H.-G. Yu, *Mol. Phys.* 1997, **91**, 301.

[60] N. Samardzija and R.L. Waterland, *Biol. Cybern.* 1991, **65**, 211.

[61] A. Cichocki and R. Uberhauen, *Biol. Cybern.* 1992, **68**, 155.

[62] Z. Yi, Y. Fu, and H.J. Tang, *Comput. Math. Appl.* 2004, **47**, 1155.

[63] I.E. Lagaris, A. Likas, and D.I. Fotiadis, *Comp. Phys. Comm.* 1997, **104**, 1.

[64] S. Manzhos, K. Yamashita, and T. Carrington Jr., *Chem. Phys. Lett.* 2009, **474**, 217.

[65] U. Manthe, *J. Chem. Phys.* 2009, **130**, 054109.

[66] C.M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, 1996).

[67] M.H. Gutknecht, *SIAM J. Matrix Anal. Appl.* 1992, **13**, 594.

[68] O. Kolin, C. Leforestier, and N. Moiseyev, *J. Chem. Phys.* 1988, **89**, 6836.

[69] S. Dallwig, N. Fahrer, and C. Schlier, *Chem. Phys. Lett.* 1992, **191**, 69.

[70] H.-G. Yu, *J. Chem. Phys.* 2005, **122**, 164107.

[71] H.-G. Yu and G. Nyman, *J. Chem. Phys.* 1999, **110**, 11133.

[72] C.C. Paige and M.A. Saunders, *SIAM J. Numer. Anal.* 1975, **12**, 617.

[73] H.D. Simon, *Math. Comput.* 1984, **42**, 115.

[74] H.-G. Yu and G. Nyman, *Chem. Phys. Lett.* 1998, **298**, 27.

[75] G. Nyman, *Int. J. Quantum Chem.* 2014, **114**, 1183.

[76] X.-G. Wang and T. Carrington Jr., *J. Chem. Phys.* 2002, **117**, 6923.

[77] A. Leclerc and T. Carrington Jr., *J. Chem. Phys.* 2014, **140**, 174111.

[78] D. Lauvergnat and A. Nauts, *Spectrochimica Acta A* 2014, **119**, 18.

[79] A. Shimshovitz, Z. Bačić, and D.J. Tannor, *J. Chem. Phys.* 2014, **141**, 234106.

[80] C. Iung and F. Gatti, *Int. J. Quantum Chem.* 2006, **106**, 130.

[81] Z. Bačić, *Comp. Phys. Comm.* 2000, **128**, 46.

[82] H.-G. Yu and J.T. Muckerman, *J. Mol. Spectrosc.* 2002, **214**, 11.

[83] H.-G. Yu, *J. Mol. Spectrosc.* 2009, **256**, 287.

[84] A.J.C. Varandas and H.-G. Yu, *Chem. Phys. Lett.* 1996, **259**, 336.

[85] H.-G. Yu, *J. Chem. Phys.* 2004, **121**, 6334.

[86] R.A. Friesner, *J. Chem. Phys.* 1986, **85**, 1462.

[87] C. Leforestier, *J. Chem. Phys.* 1994, **101**, 7357.

[88] H.-G. Yu, *Chem. Phys. Lett.* 1997, **281**, 312.

[89] A.R. Edmonds, *Angular Momentum in Quantum Mechanics* (Princeton University Press,, Princeton, NJ, 1960).

[90] G.C. Corey, J.W. Tromp, and D. Lemoine, *in NATO ARW Proceedings on Grid Methods in Atomic and Molecular Quantum Calculation* (Kluwer Academic, Dordrecht, The Netherland, 1993).

[91] B. Kendrick and R.T. Pack, *J. Chem. Phys.* 1996, **104**, 7502.

[92] V.A. Mandelshtam, T.P. Grozdanov, and H.S. Taylor, *J. Chem. Phys.* 1995, **103**, 10074.

[93] H. Zhang and S.C. Smith, *J. Chem. Phys.* 2001, **115**, 5751.

[94] M.R. Pastrana, L.A.M. Quintales, J. Brandão, and A.J.C. Varandas, *J. Phys. Chem.* 1990, **94**, 8073.

[95] D.W. Schwenke and H. Partridge, *Spectrochimica Acta* 2001, **A57**, 887.

[96] S.N. Yurchenko, J. Tennyson, R.J. Barber, and W. Thiel, *J. Mol. Spectrosc.* 2013, **291**, 69.

[97] X.-G. Wang and T. Carrington, Jr., *J. Chem. Phys.* 2014, **141**, 154106.

[98] A.V. Nikitin, M. Rey, and V.G. Tyuterev, *J. Chem. Phys.* 2015, **142**, 094118.

[99] S. Albert, S. Bauerecker, V. Boudon, L.R. Brown, J.P. Champion, M. Loëte, A. Nikitin, and M. Quack, *Chem. Phys.* 2009, **356**, 131.

[100] O. Robert, J.C. Hilico, M. Loëte, J.P. Champion, and L.R. Brown, *J. Mol. Spectrosc.* 2001, **209**, 14.

[101] A.V. Nikitin, L. Daumont, X. Thomas, L. Regalia, V. G. Tyuterev, and L.R. Brown, *J. Mol. Spectrosc.* 2011, **268**, 93.

[102] J.-C. Hilico, O. Robert, M. Loëte, S. Toumi, A.S. Pine, and L.R. Brown, *J. Mol. Spectrosc.* 2001, **208**, 1.

[103] R. Georges, M. Herman, J.-C. Hilico, and O. Robert, *J. Mol. Spectrosc.* 1998, **187**, 13.

[104] E. Venuti, L. Halonen, and R.G. Della Valle, *J. Chem. Phys.* 1999, **110**, 7339.

[105] R.Z. Martinez, D. Bermejo, J. Santo, J.P. Champion, and J.C. Hilico, *J. Chem. Phys.* 1997, **107**, 4864.

Table 1: A comparison of computed highest bound states (357-361) and lowest lying resonances of HO_2 with the total angular momentum $J = 0$ and odd O-O exchange parity. Energies E_n are relative to the asymptote H + O_2. Γ_n are the widths of resonances, where the numbers in parentheses are the power of 10. All units are in eV.

	KP[91]		MGT[92]		YS[32]		This work	
n	E_n	Γ_n	E_n	Γ_n	E_n	Γ_n	E_n	Γ_n
357			0.09031		0.09040		0.090407	
358			0.09121		0.09150		0.091503	
359			0.09228		0.09263		0.092631	
360			0.09421		0.09448		0.094480	
361			0.09656		0.09706		0.097059	
	0.098056	1.841(-4)	-	-	-	-	-	-
362	0.099152	6.387(-6)	0.099075	2.8(-5)	0.099177	3.454(-5)	0.099189	3.153(-6)
363	0.099661	6.151(-5)	0.09972	3.2(-5)	0.099896	2.687(-5)	0.099908	1.604(-5)
364	0.100274	8.227(-4)	-	-	0.100376	5.571(-4)	0.100956	6.184(-4)
365	0.102036	9.471(-5)	0.10189	8.6(-5)	0.101886	7.214(-5)	0.101907	4.361(-5)
366	0.103760	1.527(-5)	0.103717	1.56(-5)	0.103758	1.671(-5)	0.103761	1.168(-5)
367	0.104668	2.334(-5)	0.104622	3.2(-5)	0.104857	6.096(-5)	0.104862	4.399(-5)
368	0.107208	1.066(-4)	0.106964	7.6(-5)	0.106997	5.051(-5)	0.107005	3.022(-5)
369	0.110424	1.779(-4)	0.11034	1.14(-4)	0.110754	1.427(-4)	0.110770	5.101(-5)
370	0.112204	9.403(-4)	0.11214	7.4(-4)	0.112588	3.611(-4)	0.112582	6.896(-4)
371	0.114299	6.209(-5)	0.11391	1.8(-4)	0.114234	1.020(-4)	0.114264	4.404(-5)
372	0.115788	6.300(-7)	0.11561	2.6(-7)	0.115654	1.200(-6)	0.115572	5.308(-7)
373	-	-	0.11872	2.30(-3)	0.117880	3.558(-3)	0.118760	2.717(-3)
374	0.118964	1.060(-5)	0.11878	1.56(-5)	0.119039	1.254(-5)	0.119037	3.084(-6)
375	0.120778	3.827(-5)	0.12058	3.6(-5)	0.120751	3.763(-5)	0.120750	4.893(-5)

Table 2: A comparison of vibrational energy levels E_n in cm^{-1} with previous theoretical results[97] calculated on the same WC potential energy surface and experimental values.[99–105] $\Delta_{WC}/\Delta_{expt}$ are the differences between this work and WC/experiments.

label	This work	WC[97]	Δ_{WC}	Expt[99–105]	Δ_{expt}
(00)(01)F_2	1310.47	1310.47	0.01	1310.76	-0.29
(00)(10)E	1533.47	1533.47	0.00	1533.33	0.14
(00)(02)A_1	2587.00	2586.99	0.01	2587.04	-0.04
(00)(02)F_2	2613.84	2613.83	0.01	2614.26	-0.42
(00)(02)E	2624.45	2624.44	0.01	2624.62	-0.16
(00)(11)F_2	2830.21	2830.20	0.01	2830.32	-0.11
(00)(11)F_1	2845.97	2845.96	0.01	2846.07	-0.10
(10)(00)A_1	2917.23	2917.18	0.05	2916.48	0.75
(01)(00)F_2	3019.53	3019.43	0.10	3019.49	0.04
(00)(20)A_1	3063.78	3063.77	0.01	3063.65	0.13
(00)(20)E	3065.31	3065.30	0.01	3065.14	0.17
(00)(03)F_2	3870.59	3870.57	0.02	3870.49	0.11
(00)(03)A_1	3908.84	3908.82	0.02	3909.20	-0.36
(00)(03)F_1	3920.26	3920.24	0.02	3920.51	-0.25
(00)(03)F_2	3931.01	3930.99	0.02	3930.92	0.09
(00)(12)E	4101.49	4101.46	0.03	4101.39	0.10
(00)(12)F_1	4128.72	4128.69	0.03	4128.76	-0.04
(00)(12)A_1	4132.63	4132.60	0.03	4132.86	-0.24
(00)(12)F_2	4142.70	4142.67	0.03	4142.86	-0.17
(00)(12)E	4151.23	4151.21	0.02	4151.20	0.02
(00)(12)A_2	4161.91	4161.89	0.02	4161.85	0.06
(10)(01)F_2	4224.24	4224.19	0.06	4223.46	0.78
(01)(01)F_2	4319.10	4319.00	0.10	4319.21	-0.11
(01)(01)E	4322.17	4322.07	0.10	4322.18	-0.00
(01)(01)F_1	4322.44	4322.34	0.10	4322.59	-0.15
(01)(01)A_1	4322.81	4322.71	0.10	4322.70	0.11
(00)(21)F_2	4348.65	4348.62	0.04	4348.72	-0.06
(00)(21)F_1	4363.62	4363.59	0.03	4363.61	0.01
(00)(21)F_2	4378.97	4378.94	0.03	4378.95	0.02
(10)(10)E	4436.02	4435.97	0.05	4435.12	0.90
(01)(10)F_1	4537.62	4537.52	0.10	4537.55	0.07
(01)(10)F_2	4543.91	4543.81	0.10	4543.76	0.15
(00)(30)E	4592.16	4592.13	0.03	4592.03	0.13
(00)(30)A_2	4595.39	4595.36	0.02	4595.27	0.12
(00)(30)A_1	4595.66	4595.63	0.02	4595.50	0.15

(continued)

(00)(04)A_1	5122.42	5122.33	0.09	5121.34	1.08
(00)(04)F_2	5143.79	5143.70	0.09	5143.36	0.42
(00)(04)E	5167.85	5167.76	0.08	5167.16	0.69
(00)(04)F_2	5210.75	5210.67	0.07	5211.29	-0.54
(00)(04)E	5229.09	5229.02	0.07	5228.91	0.18
(00)(04)F_1	5230.89	5230.82	0.07	5230.78	0.11
(00)(04)A_1	5241.08	5241.02	0.07	5239.98	1.10
(00)(13)F_2	5370.96	5370.86	0.09	5376.95	-5.99
(00)(13)F_1	5390.12	5390.03	0.09	5393.69	-3.57
(00)(13)E	5425.08	5424.98	0.11	5424.66	0.42
(00)(13)F_2	5429.83	5429.73	0.10	5429.58	0.25
(00)(13)F_1	5437.58	5437.49	0.09	5436.79	0.79
(00)(13)F_2	5444.82	5444.73	0.09	5445.12	-0.30
(00)(13)F_1	5463.23	5463.13	0.09	5462.92	0.31
(10)(02)A_1	5494.00	5493.92	0.07	5486.44	7.56
(10)(02)F_2	5522.38	5522.32	0.07	5517.17	5.21
(10)(02)E	5534.61	5534.54	0.06	5532.18	2.43
(01)(02)F_2	5588.23	5588.12	0.11	5587.98	0.25
(01)(02)A_1	5604.87	5604.75	0.12	5606.32	-1.45
(00)(22)A_1	5613.22	5613.11	0.12	5616.42	-3.20
(00)(22)E	5613.64	5613.56	0.08	5618.27	-4.63
(01)(02)F_2	5615.28	5615.16	0.12	5623.01	-7.73
(01)(02)F_1	5615.76	5615.64	0.11	5617.99	-2.23
(01)(02)E	5619.09	5618.99	0.10	5625.28	-6.19
(01)(02)F_1	5626.23	5626.11	0.11	5632.10	-5.87
(01)(02)F_2	5627.66	5627.55	0.11	5628.40	-0.74
(00)(22)F_2	5642.57	5642.47	0.10	5640.66	1.91
(00)(22)E	5654.23	5654.14	0.09	5653.60	0.63
(00)(22)F_1	5656.00	5655.90	0.09	5655.30	0.70
(00)(22)A_2	5663.98	5663.89	0.08	5662.36	1.62
(00)(22)F_2	5668.77	5668.68	0.09	5668.98	-0.21
(00)(22)A_1	5681.92	5681.84	0.08	5682.44	-0.52
(00)(22)E	5691.34	5691.26	0.08	5691.42	-0.08
(10)(11)F_2	5727.57	5727.50	0.07	5729.68	-2.11
(10)(11)F_1	5745.84	5745.77	0.06	5756.05	-10.21
(20)(00)A_1	5790.02	5789.72	0.30	5790.25	-0.23
(01)(11)F_2	5822.48	5822.36	0.13	5826.65	-4.17
(01)(11)F_1	5825.24	5825.13	0.11	5827.29	-2.05
(01)(11)E	5832.24	5832.10	0.13	5837.04	-4.80

(continued)

(01)(11)A_1	5834.93	5834.79	0.14	5831.40	3.53
(01)(11)E	5842.81	5842.70	0.11	5841.08	1.73
(01)(11)A_2	5842.91	5842.81	0.11	5831.89	11.02
(01)(11)F_2	5844.07	5843.96	0.11	5849.30	-5.23
(01)(11)F_1	5847.14	5847.03	0.11	5845.89	1.25
(11)(00)F_2	5860.84	5860.72	0.12	5819.72	41.12
(00)(31)F_2	5867.52	5867.46	0.06	5867.66	-0.14
(00)(31)F_1	5879.75	5879.69	0.06	5879.02	0.73
(00)(31)F_2	5894.48	5894.41	0.07	5894.12	0.36
(00)(31)F_1	5909.25	5909.18	0.07	5909.71	-0.46
(10)(20)A_1	5940.19	5939.90	0.29	5971.52	-31.33
(10)(20)E	5953.33	5953.27	0.06	5974.59	-21.26
(02)(00)A_1	5968.67	5968.27	0.40	5968.15	0.52
(02)(00)F_2	6004.77	6004.43	0.33	6004.63	0.14
(02)(00)E	6044.18	6043.64	0.54	6043.80	0.38
(01)(20)F_2	6054.54	6054.42	0.11	6054.64	-0.10
(01)(20)F_1	6060.65	6060.55	0.10	6059.30	1.35
(01)(20)F_2	6065.68	6065.58	0.10	6065.32	0.36
(00)(40)A_1	6116.82	6116.74	0.08	6116.75	0.07
(00)(40)E	6118.60	6118.53	0.07	6118.62	-0.02
(00)(40)E	6124.12	6124.06	0.06	6124.17	-0.05

Figure 1: A feed-forward neural network structure with an output result $\mathcal{R} = y_1^3$, where the bias nodes are not shown.

Figure 2: Schematic structure for the multi-layered feed-forward neural network iterative diagonalization method, where the input node is a unit formally orthogonal Lanczos (FOL) polynomial, i.e., $\mathcal{L}_1^0 = 1$.

Figure 3: Orthogonal polyspherical coordinates formed by a set of Jacobi, Radau, and/or orthogonal satellite vectors for a polyatomic molecule with N atoms. The body-fixed $Z-$axis is chosen to be coincident with the $\mathbf{r}_{N-1}$ vector while the $\mathbf{r}_{N-2}$ vector lies in the XZ plane with a positive X direction.

Figure 4: A plot of convergence for computing the 400 lowest-lying bound and resonance states of HO_2 using the neural network iterative diagonalization method: (a) the average errors of those 400 states as a function of the number of $F(\tilde{H})-$vector products with a structure of five hidden layers (1000,1000,1000,1000,570), where the dashed lines divide the layers; (b) the total number of $F(\tilde{H})-$vector products needed to converge the 400 states as a function of maximum number (N^l) of nodes in each layer.

Figure 5: An amplitude plot of three typical node functions $\mathcal{L}_j^l$ at different layer l and node number j for HO_2, calculated as in Fig. 4(a).

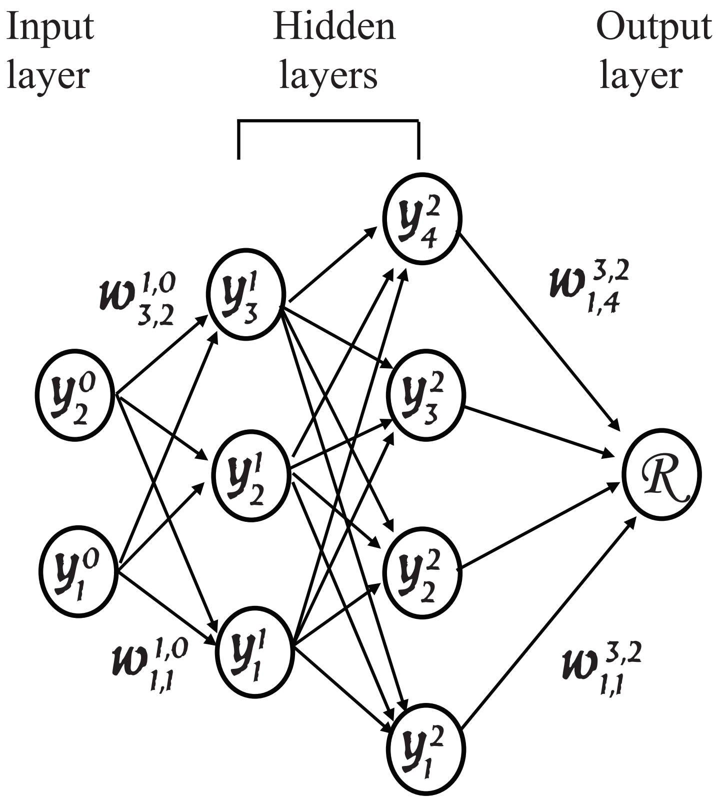
Input layer
Hidden layers
Output layer
y^2_4
y^1_3
y^2_3
y^0_2
y^1_2
$\mathcal{R}$
y^2_2
y^0_1
y^1_1
y^2_1
$w^{1,0}_{3,2}$
$w^{1,0}_{1,1}$
$w^{3,2}_{1,4}$
$w^{3,2}_{1,1}$

Fig. 1

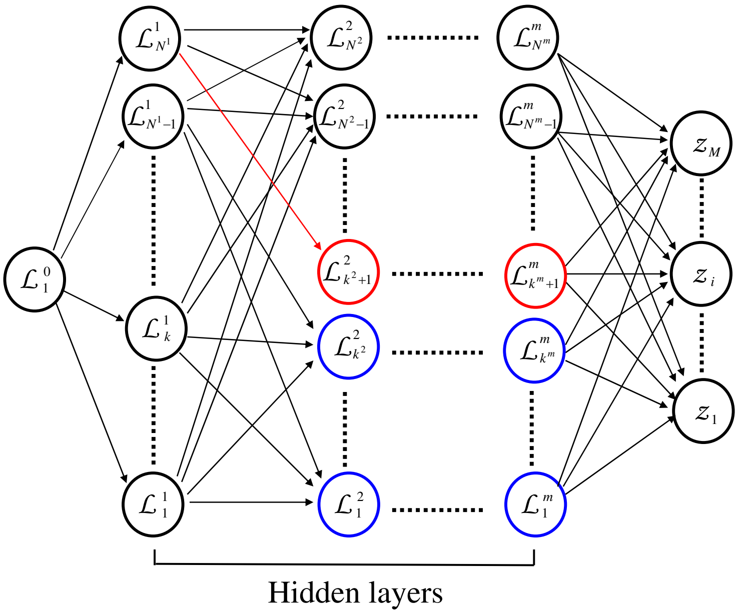
$\mathcal{L}^0_1$
$\mathcal{L}^1_{N^1}$
$\mathcal{L}^1_{N^1-1}$
$\mathcal{L}^1_k$
$\mathcal{L}^1_1$
$\mathcal{L}^2_{N^2}$
$\mathcal{L}^2_{N^2-1}$
$\mathcal{L}^2_{k^2+1}$
$\mathcal{L}^2_{k^2}$
$\mathcal{L}^2_1$
$\mathcal{L}^m_{N^m}$
$\mathcal{L}^m_{N^m-1}$
$\mathcal{L}^m_{k^m+1}$
$\mathcal{L}^m_{k^m}$
$\mathcal{L}^m_1$
z_M
z_i
z_1
Hidden layers

Fig. 2

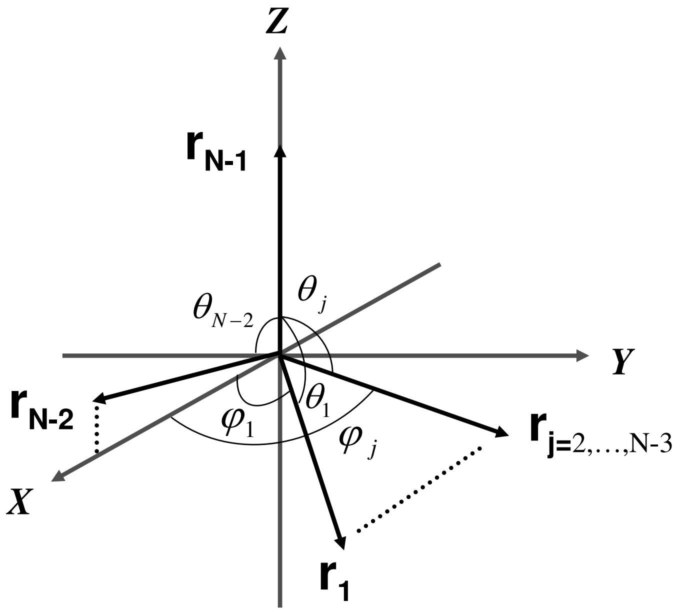
Z
$\mathbf{r}_{\mathbf{N-1}}$
θ_j
θ_{N-2}
Y
$\mathbf{r}_{\mathbf{N-2}}$
θ_1
φ_1
φ_j
$\mathbf{r}_{\mathbf{j=}2,\ldots,\mathrm{N}\text{-}3}$
X
$\mathbf{r}_{\mathbf{1}}$

Fig. 3

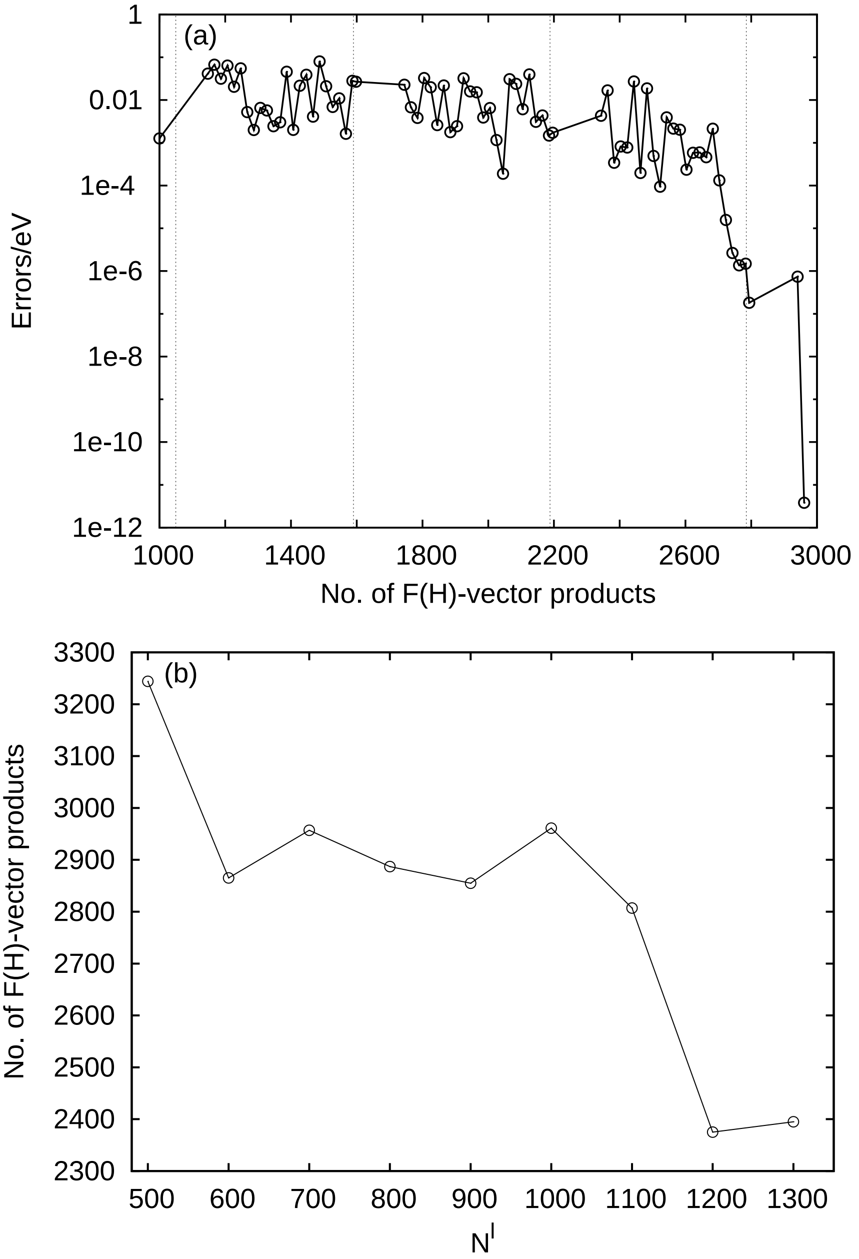
(a)
Errors/eV
1
0.01
1e-4
1e-6
1e-8
1e-10
1e-12
1000
1400
1800
2200
2600
3000
No. of F(H)-vector products
(b)
No. of F(H)-vector products
3300
3200
3100
3000
2900
2800
2700
2600
2500
2400
2300
500
600
700
800
900
1000
1100
1200
1300
N^I

Fig. 4

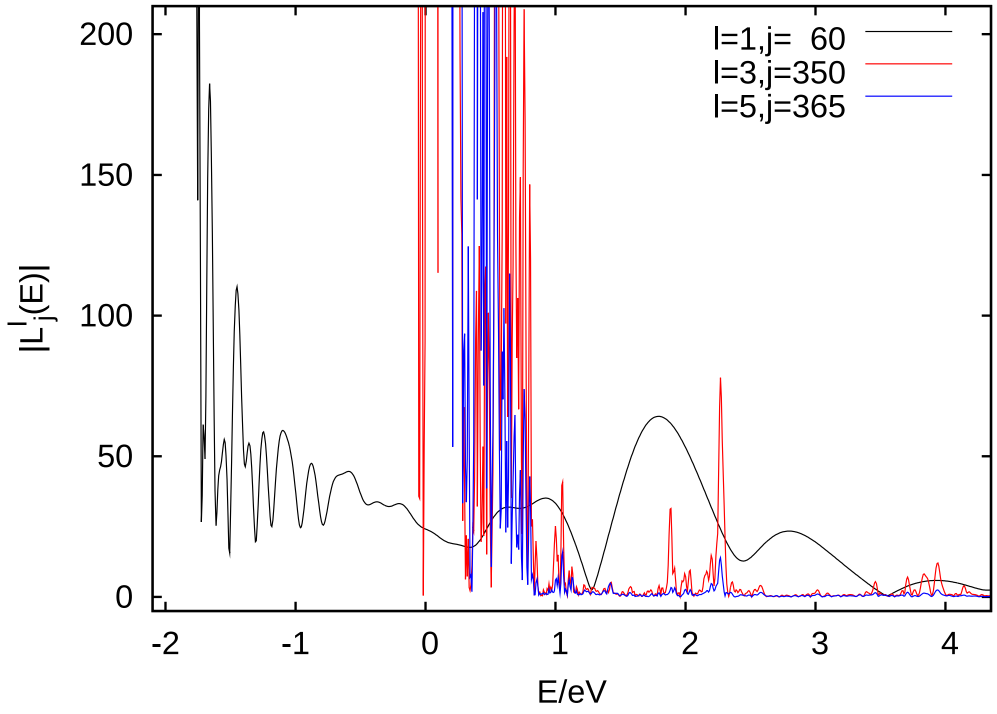

Fig. 5