# The fitting of potential energy surfaces using neural networks: Application to the study of vibrational levels of $H_3^+$

Cite as: J. Chem. Phys. **109**, 8801 (1998); https://doi.org/10.1063/1.477550
Submitted: 18 May 1998 . Accepted: 20 August 1998 . Published Online: 13 November 1998

Frederico V. Prudente, Paulo H. Acioli, and J. J. Soares Neto

J. Chem. Phys. **109**, 8801 (1998); https://doi.org/10.1063/1.477550 **109**, 8801

# The fitting of potential energy surfaces using neural networks: Application to the study of vibrational levels of $H_3^+$

Frederico V. Prudente,[a] Paulo H. Acioli, and J. J. Soares Neto
*Núcleo de Física Atômica, Molecular e de Fluidos, Instituto de Física—Universidade de Brasília, CP 04455, 70910-900 Brasília—DF, Brazil*

(Received 18 May 1998; accepted 20 August 1998)

A back-propagation neural network is utilized to fit the potential energy surfaces of the $H_3^+$ ion, using the *ab initio* data points of Dykstra and Swope, and the Meyer, Botschwina, and Burton *ab initio* data points. We used the standard back-propagation formulation and have also proposed a symmetric formulation to account for the symmetry of the $H_3^+$ molecule. To test the quality of the fits we computed the vibrational levels using the correlation function quantum Monte Carlo method. We have compared our results with the available experimental results and with results obtained using other potential energy surfaces. The vibrational levels are in very good agreement with the experiment and the back-propagation fitting is of the same quality of the available potential energy surfaces. © *1998 American Institute of Physics.* [S0021-9606(98)30644-3]

## I. INTRODUCTION

The study of rovibrational levels of $H_3^+$ ion have interested experimentalists[1] and theoreticians[2–6] for many decades. The relative simplicity of the hydrogen molecular ion makes it a good target for extensive quantum chemical calculation. For example, the *ab initio* calculations of the electronic structure for a set of nuclear geometries are performed, the data points are fit to define the potential energy surface (PES) over the whole nuclear configuration space and the rovibrational energy levels determined for a given PES. For each of these steps, accurate methods are necessary to obtain good results.

The usual method to fit a many-dimensional PES is the use of a power series in an appropriate coordinate system. Examples of this methodology are the bond order potential,[7] the Simons–Parr–Finlan (SPF) expansion,[8] and the Morse-type symmetry adapted deformation coordinates.[3,4] The main difficulty, in this case, is obtaining the best set of coefficients for the expansion to fit the surfaces. Other alternative methods have been utilized to describe PES, like, for example, the genetic programming,[9] the truncated singular value decomposition method,[10] the interpolation of local Taylor expansion method using Cartesian coordinates[11] and multi-layer neural network.[12,13]

Neural network (NN) methods have been used in a variety of applications in chemistry and physics.[14,15] Particularly in chemical physics NNs have been recently applied with great success. For example, NN have been used to calculate the ground-state eigenenergy of two-dimensional harmonic oscillators,[16] to solve nonhomogeneous ordinary and partial differential equations (e.g., Schrödinger and Dirac equations),[17] to predict the rainbow trajectories in atomic and molecular classical collisions[18] and to obtain the electronic correlation energy for atoms and diatomic molecules.[19] To interpolate a PES, the NN method does not need to know *a priori* the shape of the surface and utilizes a small set of *ab initio* points. The outcome of the interpolation is smooth, continuous, and includes all features of the surface. Other features are: the expression for the PES is relatively simple and accurate, their evaluations are cheap, and the derivatives of the PES are also easily obtainable.[12,13]

There are many algorithms to train a multi-layer neural network. A relatively simple and widely used procedure to train the NN is the back-propagation network (BPN) algorithm.[14,15] The BPN algorithm is an iterative gradient descendent technique that minimizes the global error between the exact output and the neural network output. Other methods replace the gradient descendent methods (first-order optimization) by more sophisticated minimization algorithms such as second-order optimization methods.[20] Examples of these methods are: conjugate or scaled conjugate gradient procedure[21] and the adaptive, global, extended Kalman filters method.[22] We can also mention the use of the Bayesian probability theory to train the supervised neural network.[23,24]

In the present work, we use the BPN algorithm to generate the PES for the $H_3^+$ ion and calculate the vibrational energy spectrum. The $H_3^+$ ion is symmetric by permutation of two nuclei. *A priori*, the usual NN method does not consider any symmetry of the problem. To account for this symmetry we propose the use of symmetric neurons in NN methods. For the calculation of the vibrational energy levels, we utilize the correlation function quantum Monte Carlo (CFQMC).[25,26] The CFQMC approach is a very appealing method to use to treat vibrational and rotational states. The great advantage of the NN and CFQMC methods is that they can be used together to treat larger molecules without major modifications.

This paper was ordered as follows: in Sec. II we describe the usual NN methods (Subsection II A) and propose the symmetric neural network (SNN) method (Subsection II B). The CFQMC method developed by Bernu and co-workers[25] is reviewed in Sec. III. In Subsection IV A we present the

[a] Author to whom all correspondence should be addressed; electronic mail: fred@fis.unb.br

PES fitted with usual NN using the Dykstra and Swope (DS) *ab initio* points[5] and the PES fitted with SNN using the Meyer, Botschwina, and Burton (MBB) *ab initio* points[3] in Subsection IV B. The vibrational energy spectra is shown in Subsection IV C. The last section is dedicated to concluding remarks.

## II. NEURAL NETWORK

### A. Usual formulation

An artificial neural network is a highly flexible nonlinear parallel computational device inspired by the structure of the brain. It is a system made up of a number of simple processing elements called ''neurons,'' typically arranged in layers and interconnected via a set of ''links'' through unidirectional information. One neuron receives a set of input signals $\{x_i\}$ and transforms it, emitting an output signal, as a synapse. For this, the neuron multiplies each input by a constant (connection weight), sums over all inputs, and applies a nonlinear transfer function in the result of this sum. The use of a nonlinear transfer functions is essential to add nonlinearities into the NN besides other advantages.[27] Then, the output $y_i$ of the $i$th neuron is given by

$$y_i=\phi\left(\sum_{j=1}^{N} w_{ij}x_j+w_{i0}\right), \tag{1}$$

where $x_j$, $j=1,...,N$ are the inputs, $w_{i0}$ is a bias, $\{w_{ij}\}$ is a set of the connection weights of the links, and $\phi(z)$ is a transfer function. The inputs can be input data or output of neurons in the previous layer. The form of the transfer function $\phi(z)$ determines the behavior of the neural network. There are many types of transfer functions, for example, the sigmoidal function:

$$\phi(z)=\frac{1}{1+e^{-z}}, \tag{2}$$

the tangent hyperbolic function:

$$\phi(z)=\tanh(z), \tag{3}$$

the bond order function:

$$\phi(z)=e^{-z}, \tag{4}$$

and the linear function:

$$\phi(z)=z. \tag{5}$$

In particular, we utilize the bond order function (4) for the hidden layers and the linear function (5) for the output layer.

The multi-layer perception (MLP) is a relatively simple and widely used neural network type for interpolation problems.[28] A MLP is a NN that has various layers, with the neurons of one layer connected with the neurons of the neighboring layers (see Fig. 1). For a MLP with three layers fully connected, the outputs are given by

$$y_i^2=\phi_2\left(\sum_j w_{ij}^{(2)}y_j^1+w_{i0}^{(2)}\right) \tag{6}$$

with

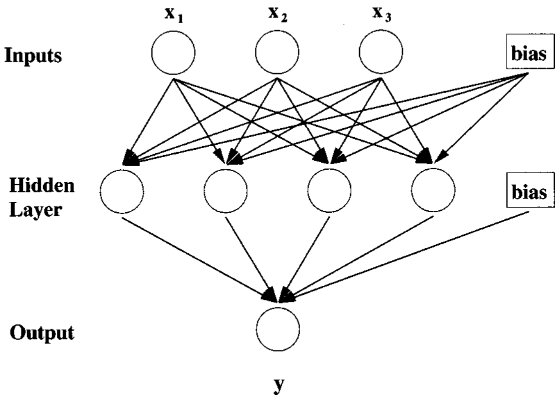

FIG. 1. Schematic representation of the neural network architecture used to fit the PES. $X_i$ represent the inputs, the internal distances, and $y$ represent the output the potential energy. The actual number of neurons in the hidden layer varies according to the kind of fitting, UNN or SNN.

$$y_j^1=\phi_1\left(\sum_k w_{jk}^{(1)}y_k^0+w_{j0}^{(1)}\right), \tag{7}$$

where $w_{j0}^{(i)}$ are the bias, the index ''2'' refers to the output layer, the index ''1'' refers to the hidden layer, and the input data $\{y_k^0=x_k\}$ are considered as the output of the first layer or input layer (the index ''0''). Since we want to fit the PES for a triatomic molecule, the NN will have a three-dimensional input (the nuclei distances $r_{AB}$, $r_{BC}$, $r_{CA}$) and have as output the potential energy value for the particular nuclear geometry $[V(r_{AB},r_{BC},r_{CA})]$. We can notice that there is no symmetry in the input data in Eq. (6). This fact influences in the network architecture (number of layers, number of neurons in each layer, and the transfer function type). A simple way to describe the neural network architecture is to list the number of input data (neurons in input layer) and the number of neurons in each layer (hidden and output), followed by letters denoting the form of the transfer functions used in the layer [$b$ for bond order function (4) and $l$ for linear function (5)]. In this notation, the NN of Fig. 1 has a $3-4b-1l$ structure.

Now, we show how to obtain the best set of weights that describes the *ab initio* calculation of the energy potential for a few nuclear configurations. The BPN algorithm has the following steps: the initial weights are set randomly in the $-0.5$ and 0.5 range and they are not a good description of the surface. Then, we correct the weights in the following way:

$$\Delta w_{ji}^{(l)}=w_{ji}^{(l)(\text{new})}-w_{ji}^{(l)(\text{old})}=-\eta\frac{\partial\epsilon}{dw_{ji}^{(l)}}, \tag{8}$$

where $\Delta w_{\underline{ji}}^{(l)}$ is the correction of the weight $w_{ji}^{(l)}$, $\epsilon=(1/2)\Sigma_j(y_j-y_j^{\text{out}})^2$ is the global error function, and $\eta$ is a positive numerical factor. To evaluate Eq. (8), we use the following chain rule:

$$\frac{\partial\epsilon}{\partial w_{ji}^{(l)}}=\frac{\partial\epsilon}{\partial y_j^l}\frac{\partial y_j^l}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{ji}^{(l)}}=-\delta_j^l\frac{\partial z_j^l}{\partial w_{ji}^{(l)}}, \tag{9}$$

where $z_j^l=\Sigma_k w_{jk}^{(l)}y_k^{l-1}$. Thus we obtain[13]

$$\Delta w_{ji}^{(l)}=\eta\delta_j^l y_i^{l-1}+\mu\Delta w_{ji}^{(l)(\text{previous})}, \tag{10}$$

where $y_i^{l-1}$ is the output of the $i$th neuron of the $(l-1)$th layer, $\eta$ is the learning rate, and $\mu$ is the momentum constant, both ranging between zero and one. The last term in Eq. (10) is added to eliminate the stagnancy at local minima. The generalized delta function $\delta_j^l$ of Eq. (6) has the following definition:

$$\delta_j^{\text{out}}=(\overline{y_j}-y_j)\,\frac{\partial\phi_{\text{out}}(z)}{\partial z} \tag{11}$$

for the output layer, where $\overline{y_j}$ is the exact output, and

$$\delta_j^l=\left(\sum_{k=1}^{p}\delta_k^{l+1}w_{kj}^{l+1}\right)\frac{\partial\phi_l(z)}{\partial z} \tag{12}$$

for the hidden layers, where $p$ represents the $i$th neuron of the $(l+1)$th layer. This procedure continues until the global error is less than a given tolerance.

### B. Symmetrical formulation

In triatomic molecules that have three identical nuclei, as the $H_3^+$ ion, the potential energy is invariant by permutation operation of nuclei:

$$\begin{aligned}V(r_{AB},r_{BC},r_{CA})&=V(r_{BC},r_{AB},r_{CA})\\&=V(r_{CA},r_{BC},r_{AB})\\&=V(r_{AB},r_{CA},r_{BC})\\&=V(r_{CA},r_{AB},r_{BC})\\&=V(r_{BC},r_{CA},r_{AB}).\end{aligned} \tag{13}$$

As seen in the previous subsection, the usual formulation of the NN methods does not consider this symmetry of the problem.

To formulate the symmetric form of the NN methods, it is convenient to introduce the symmetric symbol $\varepsilon_{ijk}$ defined by

$$\begin{aligned}&\varepsilon_{123}=\varepsilon_{231}=\varepsilon_{312}=\varepsilon_{132}=\varepsilon_{213}=\varepsilon_{321}=1,\\&\text{all others }\ \varepsilon_{ijk}=0\end{aligned} \tag{14}$$

and name $y_1^0=r_{AB}$, $y_2^0=r_{BC}$, and $y_3^0=r_{CA}$. Note that $\varepsilon_{ijk}$ is totally symmetric with respect to all pairs of indices.

Initially, we define symmetric neurons, that are given by

$$\tilde{y}_m^1=\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\varepsilon_{ijk}\phi(w_{m1}y_i^0+w_{m2}y_j^0+w_{m3}y_k^0+w_{m0}), \tag{15}$$

where the transfer function $\phi(z)$ is given by Eqs. (2), (3), (4), or (5). Then, we propose the following form of a symmetric multi-layer perception (SMLP): the first hidden layer constituted of symmetric neurons [Eq. (15)] while the other hidden layers and the output layer are defined of the usual form. For an SMLP with three layers fully connected, the outputs are given by

$$y_i^2=\phi_2\left(\sum_j w_{ij}^{(2)}\tilde{y}_j^1+w_{i0}^{(2)}\right). \tag{16}$$

Here, the notation used is the same as in the usual formulation.

As in the usual NN formulation, we use the BPN algorithm to train the SMLP. But, for the first hidden layer, we evaluate the partial derivatives in Eq. (8) as:

$$\frac{\partial\epsilon}{\partial w_{ji}^{(1)}}=\frac{\partial\epsilon}{\partial\tilde{y}_j^1}\,\frac{\partial\tilde{y}_j^1}{\partial w_{ji}^{(1)}} \tag{17}$$

with the global error function defined in the previous subsection. The first term of the Eq. (17) can be estimated using Eq. (12),

$$\frac{\partial\epsilon}{\tilde{y}_j^1}=-\left(\sum_{k=1}^{p}\delta_k^2w_{kj}^{(2)}\right), \tag{18}$$

where $p$ is the neuron number in the third layer. To evaluate the second term of the Eq. (17), we substitute Eq. (15) in to obtain:

$$\begin{aligned}\frac{\partial\tilde{y}_j^1}{\partial w_{j1}^{(1)}}&=\sum_{k=1}^{3}\sum_{l=1}^{3}\sum_{m=1}^{3}\varepsilon_{klm}y_k^0\,\frac{\partial\phi_1(z_{klm})}{\partial z_{klm}},\\\frac{\partial\tilde{y}_j^1}{\partial w_{j2}^{(1)}}&=\sum_{k=1}^{3}\sum_{l=1}^{3}\sum_{m=1}^{3}\varepsilon_{klm}y_l^0\,\frac{\partial\phi_1(z_{klm})}{\partial z_{klm}},\\\frac{\partial\tilde{y}_j^1}{\partial w_{j3}^{(1)}}&=\sum_{k=1}^{3}\sum_{l=1}^{3}\sum_{m=1}^{3}\varepsilon_{klm}y_m^0\,\frac{\partial\phi_1(z_{klm})}{\partial z_{klm}},\\\frac{\partial\tilde{y}_j^1}{\partial w_{j0}^{(1)}}&=\sum_{k=1}^{3}\sum_{l=1}^{3}\sum_{m=1}^{3}\varepsilon_{klm}\,\frac{\partial\phi_1(z_{klm})}{\partial z_{klm}},\end{aligned} \tag{19}$$

where $z_{klm}=w_{j1}^{(1)}y_k^0+w_{j2}^{(1)}y_l^0+w_{j3}^{(1)}y_m^0+w_{j0}^{(1)}$. Thus the weight correction is given by

$$\Delta w_{ji}^{(1)}=\eta\left(\sum_{r=1}^{r}\delta_k^2w_{kj}^{(2)}\right)\frac{\partial\tilde{y}_j^1}{\partial w_{ji}^{(1)}}+\mu\Delta w_{ji}^{(1):(\text{previous})}. \tag{20}$$

The weight corrections of other hidden layers and the output layer are computed using Eq. (10) with the generalized delta functions given by Eqs. (11) and (12).

## III. CORRELATION FUNCTION QUANTUM MONTE CARLO

In this section we describe the CFQMC method used to compute the vibrational energy levels for a given PES. We borrow the notation from Ref. 25. The problem to be solved is the eigenvalue equation

$$H\Phi_i(\mathbf{R})=E_i\Phi_i(\mathbf{R}) \tag{21}$$

with

$$H=-\sum_{i=1}^{N}\frac{\hbar^2}{2m_i}\nabla_i^2+V(\mathbf{R}), \tag{22}$$

where $\mathbf{R}$ is the vector of the coordinates of all $N$ particles of the system, $V(\mathbf{R})$ is the potential energy, and $E_i$ and $\Phi_i$ are the eigenvalues and eigenvectors of $H$. Let $\{f_\alpha(\mathbf{R})\}$ be a basis set of $m$ known functions, we can define the overlap and Hamiltonian matrix elements as

TABLE I. Convergence information for the different fittings of the $H_3^+$ PES.

| | UNN—PES | SNN—PES | MBB—PES |
|---|---|---|---|
| rms (a.u.) | $1.2\times10^{-4}$ | $1.0\times10^{-4}$ | $7.0\times10^{-5}$ |
| Smallest diff. ($cm^{-1}$) | 1.5 | $2.9\times10^{-2}$ | 0.0 |
| Largest diff. ($cm^{-1}$) | 108.8 | 41.3 | 41.3 |

$$N_{\alpha\beta}(t)=\int d\mathbf{R}_1\ d\mathbf{R}_2 f_\alpha(\mathbf{R}_2)e^{-tH}f_\beta(\mathbf{R}_1),$$
$$H_{\alpha\beta}(t)=\int d\mathbf{R}_1\ d\mathbf{R}_2 H f_\alpha(\mathbf{R}_2)e^{-tH}f_\beta(\mathbf{R}_1). \tag{23}$$

Associated to these matrices we can define the generalized eigenvalue problem

$$\sum_{\beta=1}^{m}[H_{\alpha\beta}(t)-\Lambda_k(t)N_{\alpha\beta}(t)]d_{k\beta}(t)=0 \tag{24}$$

with $d_k(t)$ being the $k$th eigenvector and $\Lambda_k(t)$ its associated eigenvalue. It has been shown[25] that

$$\lim_{t\to\infty}\Lambda_k(t)=E_k,\quad 1\leqslant k\leqslant m, \tag{25}$$

as long as the basis set $\{f_\alpha(\mathbf{R})\}$ has some overlap with the eigenstates $\{\Phi_i\}$, and is linearly independent. Although, in Ref. 25 it is shown that $\Lambda_k(t)$ is monotonically decreasing and converges exponentially fast to $E_k$, it is desirable that the overlap between $\{f_\alpha(\mathbf{R})\}$ and $\{\Phi_i\}$, be the largest possible. This condition will assure that the convergence is reached before the exponentially growing round-off errors dominate the procedure.

The matrix elements defined in Eq. (23) are evaluated using Monte Carlo methods. The use of Monte Carlo assures that the procedure can be applied to larger molecules, which is not the case of standard quadrature methods. The algorithm devised by Bernu *et al.*[25] to compute the matrix elements in Eqs. (23) and (24) consists of three major steps: (1) the generation of the random walks; (2) the calculation, during the walks, of the basis functions $f_\alpha$ and the accumulation of the matrix elements $N_{\alpha\beta}(t)$ and $H_{\alpha\beta}(t)$; (3) the diagonalization of Eq. (24) where the eigenvalues of $H$ are determined.

The problem of projecting the stationary states of the Hamiltonian can be recast into the following diffusion equation in imaginary time[29]

$$-\frac{\partial g(\mathbf{R},t)}{\partial t}=-d\nabla^2 g(\mathbf{R},t)+[E_L(\mathbf{R})-E_T]g(\mathbf{R},t)$$
$$+d\nabla(g(\mathbf{R},t)G(\mathbf{R})), \tag{26}$$

where $g(\mathbf{R},t)=\Psi(\mathbf{R})\Phi(\mathbf{R},t)$, $d=\hbar^2/2m$, $\Psi$ is a trial (guiding) wave function, $E_L=\Psi^{-1}H\Psi$ is the local energy, $E_T$ is a trial energy introduced to keep the random walks stable, and $G(\mathbf{R})=2\nabla\Psi/\Psi$. To generate the random walks according to this equation we use the following updating scheme of the coordinates at a time step $\tau$

$$\mathbf{R}_{i+1}=\mathbf{R}_i+\tau d\Psi^{-1}(\mathbf{R}_i)\nabla\Psi(\mathbf{R}_i)+(2\tau d)^{1/2}\chi_i, \tag{27}$$

where $\chi_i$ is a normally distributed random variable vector with zero mean and unit variance. After successive applications of Eq. (27) we obtain a "trajectory" in phase space with probability $|\Psi(\mathbf{R})|^2$. To determine the matrix elements, each fragment of the trajectory must be weighted by

$$W_{n,n+k}=\exp\left[-0.5\tau\sum_{j=n}^{n+k-1}[E_{L_\Psi}(\mathbf{R}_j)+E_{L_\Psi}(\mathbf{R}_{j+1})]\right], \tag{28}$$

where the local energy function is defined as

$$E_{L_\Psi}=\Psi^{-1}H\Psi. \tag{29}$$

The matrix elements, after symmetrization, are evaluated as

$$n_{\alpha\beta}(k\tau)=\frac{1}{2p}\sum_{i=1}^{p}[F_\alpha(\mathbf{R}_i)F_\beta(\mathbf{R}_{i+k})$$
$$+F_\alpha(\mathbf{R}_{i+k})F_\beta(\mathbf{R}_i)]W_{i,i+k},$$

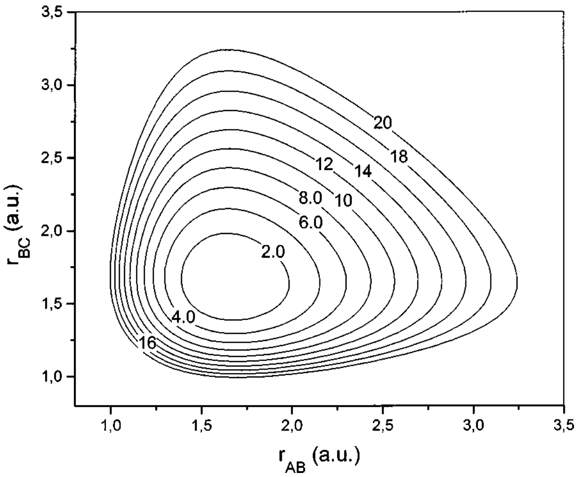

FIG. 2. UNN fitting of the Dykstra–Swope *ab initio* $H_3^+$ energies in internal nuclear distances coordinates for $C_{2v}$ symmetry. The labels in the contour plots are given in units of 1000 $cm^{-1}$.

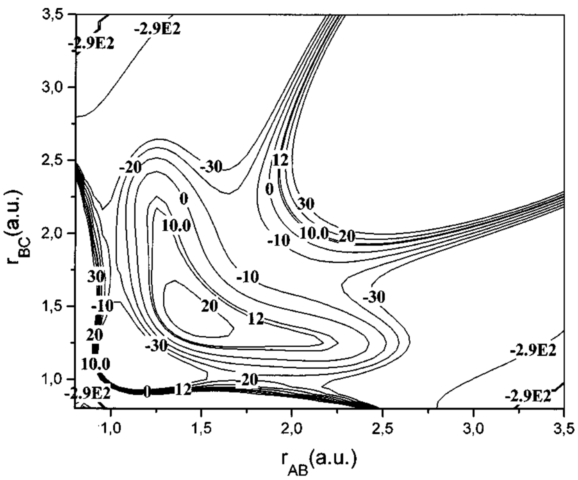

FIG. 3. Difference between UNN-PES and MBB-PES. The labels in the contour plots are given in $cm^{-1}$.

TABLE II. Connection weights for the SNN-PES.

| $l$ | $j$ | $i$ | $w_{ji}^{(l)}$ | $l$ | $j$ | $i$ | $w_{ji}^{(l)}$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0.655 601 325 181 143 11 | 1 | 10 | 3 | 0.712 343 281 462 884 73 |
| 1 | 1 | 2 | 0.713 525 417 861 489 82 | 1 | 10 | 4 | −0.230 881 621 106 248 82 |
| 1 | 1 | 3 | 0.735 413 219 920 566 74 | 1 | 11 | 1 | 0.432 333 389 145 059 31 |
| 1 | 1 | 4 | 0.171 385 905 915 072 33 | 1 | 11 | 2 | 1.035 715 087 528 823 24 |
| 1 | 2 | 1 | −0.026 004 414 852 464 25 | 1 | 11 | 3 | 1.005 615 702 460 030 07 |
| 1 | 2 | 2 | 0.611 058 551 277 053 22 | 1 | 11 | 4 | −0.162 974 796 414 466 13 |
| 1 | 2 | 3 | −0.023 487 945 983 030 29 | 1 | 12 | 1 | 0.820 041 831 337 589 01 |
| 1 | 2 | 4 | 1.224 893 675 142 328 50 | 1 | 12 | 2 | 0.879 225 197 779 645 92 |
| 1 | 3 | 1 | 0.003 452 547 807 931 53 | 1 | 12 | 3 | 1.156 606 605 270 443 34 |
| 1 | 3 | 2 | 3.724 360 000 174 900 43 | 1 | 12 | 4 | −0.309 459 994 614 071 98 |
| 1 | 3 | 3 | 0.003 452 539 138 008 57 | 1 | 13 | 1 | 3.488 829 112 162 201 26 |
| 1 | 3 | 4 | 0.146 491 496 905 901 57 | 1 | 13 | 2 | 0.135 310 343 075 184 64 |
| 1 | 4 | 1 | 0.155 951 137 463 906 40 | 1 | 13 | 3 | −0.101 171 299 896 525 92 |
| 1 | 4 | 2 | 0.884 784 048 015 142 65 | 1 | 13 | 4 | −0.789 304 871 259 971 06 |
| 1 | 4 | 3 | 1.125 360 763 322 320 38 | 1 | 14 | 1 | 0.666 071 504 115 773 32 |
| 1 | 4 | 4 | 0.016 781 833 002 258 26 | 1 | 14 | 2 | 0.890 870 104 376 975 00 |
| 1 | 5 | 1 | 1.304 473 897 887 720 17 | 1 | 14 | 3 | 0.901 470 353 596 685 81 |
| 1 | 5 | 2 | 1.143 890 621 979 500 64 | 1 | 14 | 4 | −0.201 078 416 677 049 29 |
| 1 | 5 | 3 | 0.374 018 351 507 655 22 | 1 | 15 | 1 | 3.302 666 481 306 507 42 |
| 1 | 5 | 4 | −0.319 130 572 857 041 88 | 1 | 15 | 2 | 0.352 962 036 821 814 54 |
| 1 | 6 | 1 | 0.647 544 465 953 624 80 | 1 | 15 | 3 | −0.287 161 826 848 343 52 |
| 1 | 6 | 2 | 1.148 147 246 770 428 12 | 1 | 15 | 4 | −0.898 412 217 476 907 80 |
| 1 | 6 | 3 | 0.707 248 572 696 428 27 | 2 | 1 | 1 | −0.211 973 986 444 823 76 |
| 1 | 6 | 4 | −0.309 923 041 684 717 79 | 2 | 1 | 2 | −0.773 851 247 927 557 99 |
| 1 | 7 | 1 | 0.646 356 128 195 743 17 | 2 | 1 | 3 | 3.024 165 433 144 128 11 |
| 1 | 7 | 2 | 0.820 976 437 227 810 44 | 2 | 1 | 4 | 0.237 158 419 260 480 56 |
| 1 | 7 | 3 | 0.952 700 342 482 787 23 | 2 | 1 | 5 | −0.077 663 242 982 765 56 |
| 1 | 7 | 4 | −0.231 213 510 789 191 75 | 2 | 1 | 6 | −0.034 741 512 220 674 64 |
| 1 | 8 | 1 | −0.210 559 209 162 910 74 | 2 | 1 | 7 | −0.136 588 705 031 477 99 |
| 1 | 8 | 2 | 1.660 619 542 603 544 83 | 2 | 1 | 8 | 0.858 765 756 180 830 29 |
| 1 | 8 | 3 | 0.331 247 783 710 864 88 | 2 | 1 | 9 | −0.235 951 025 883 784 80 |
| 1 | 8 | 4 | 0.329 713 600 117 669 87 | 2 | 1 | 10 | 0.049 586 818 555 204 51 |
| 1 | 9 | 1 | 0.775 455 942 952 160 47 | 2 | 1 | 11 | 0.110 735 716 696 296 13 |
| 1 | 9 | 2 | 0.770 560 488 373 439 38 | 2 | 1 | 12 | 0.120 200 879 434 381 19 |
| 1 | 9 | 3 | 0.760 360 558 536 967 49 | 2 | 1 | 13 | −0.076 636 345 900 629 57 |
| 1 | 9 | 4 | −0.127 505 179 827 894 25 | 2 | 1 | 14 | −0.100 441 550 849 502 24 |
| 1 | 10 | 1 | 0.560 686 456 244 066 88 | 2 | 1 | 15 | −0.310 580 465 321 524 89 |
| 1 | 10 | 2 | 1.206 999 233 468 548 43 | 2 | 1 | 16 | 0.375 523 945 573 847 38 |

$$
\begin{aligned}
h_{\alpha\beta}(k\tau)=\frac{1}{4p}\sum_{i=1}^{p}\,&[F_\alpha(\mathbf{R}_i)F_\beta(\mathbf{R}_{i+k})E_{L_\beta}(\mathbf{R}_{i+k})\\
&+F_\alpha(\mathbf{R}_{i+k})F_\beta(\mathbf{R}_i)E_{L_\beta}(\mathbf{R}_i)\\
&+E_{L_\alpha}(\mathbf{R}_i)F_\alpha(\mathbf{R}_i)F_\beta(\mathbf{R}_{i+k})\\
&+E_{L_\alpha}(\mathbf{R}_{i+k})F_\alpha(\mathbf{R}_{i+k})F_\beta(\mathbf{R}_i)]W_{i,i+k}, \qquad (30)
\end{aligned}
$$

where

$$F_\alpha(\mathbf{R})=f_\alpha(\mathbf{R})/\Psi(\mathbf{R}), \qquad (31)$$

$$E_{L_\beta}(R)=f_\beta^{-1}(\mathbf{R})Hf_\beta(\mathbf{R}). \qquad (32)$$

These symmetrized equations assure that $H$ is Hermitian and $N$ is symmetric. After the system has reached the stationary state the generalized eigenvalue problem is solved and the vibrational spectrum is obtained.

### A. Trial wave functions

In this work we used trial wave functions similar to those used by Bernu *et al.* Namely, for the ground state

$$\Psi_0=\exp\left(\sum_{\nu,\mu}\Delta S_\nu A_{\nu,\mu}\Delta S_\mu\right), \qquad (33)$$

with $\Delta S_\nu=S_\nu-S_\nu^0$, $S_\nu=|\mathbf{r}_i-\mathbf{r}_j|$ the distance between atoms $i$ and $j$, and $S_\nu^0=|\mathbf{r}_i-\mathbf{r}_j|^0$ the equilibrium distance between atoms $i$ and $j$. The variational parameters $\{A_{\nu\mu}\}$ are optimized in order to minimize the variational energy or its variance. For the excited trial functions we used

$$f_\alpha=\Psi_0\prod_{\nu=1}^{3}(\Delta S_\nu)^{n_\nu(\alpha)}=(n_1,n_2,n_3). \qquad (34)$$

Our choice of guiding function $\Psi$ to generate the random walks is

$$\Psi(\mathbf{R})=\Psi_0^{1/4}(\mathbf{R}). \qquad (35)$$

This guiding function is chosen in order to assure a good integration of all the states included in our calculations.

## IV. RESULTS

### A. Dykstra–Swope $H_3^+$ potential surface

Dykstra and Swope[5] calculated the potential energy of the $H_3^+$ ion for 68 nuclear geometries. They utilized internal coordinates named displacement coordinates in cylindrical polar form ($S$, $R$, and $\phi$). The $H_3^+$ potential surface in this coordinate system has a 120° period in terms of $\phi$. Accordingly, $\phi = 0°,120°,240°$ corresponds to equivalent positive symmetric bending motions, $\phi = 60°,180°,300°$ corresponds to equivalent negative symmetric bending motions, and $\phi = 30°,150°,270°$ to equivalent asymmetric stretch motions.[30] Carney *et al.*[6] fitted the DS *ab initio* points using the SPF expansion. They deleted four geometries (56, 63, 65, and 66 identification numbers as found in Table I of Dykstra and Swope paper) because of an energy acceptance criteria.

We fit 64 *ab initio* points calculated by Dykstra and Swope (DS points) using the usual NN method and utilize the internal nuclear distances as our coordinate system. For this, we generate all symmetric nuclear configurations, totaling 274 geometries of the PES to train the NN. The neural network structure used is $3-40b-1l$ in the total of 201 connection weights. Since the usual NN does not consider the symmetry of the problem, we have to obtain the potential energy for the six symmetric geometries and calculate the average value of the potential energy. Table I has the convergence information and Fig. 2 shows our fitting of the DS-$H_3^+$ PES. The difference between the new results and the MBB potential[3] is plotted in Fig. 3. We named this surface usual neural network (UNN) PES.

### B. Meyer–Botschwina–Burton $H_3^+$ potential surface

Meyer, Botschwina, and Burton[3] calculated the potential energy of the $H_3^+$ ion for 69 nuclear geometries. They utilized internal coordinates named Morse-type symmetry adapted deformation coordinates (MSADC) $S_a$, $S_x = S_e \cos(\phi)$, and $S_y = S_e \sin(\phi)$. This coordinate system has the same property with respect to the $\phi$ angle that the coordinate system used by Dykstra and Swope. Meyer *et al.* fitted the $H_3^+$ PES using a power order and calculated the vibrational energy levels of $H_3^+$ and its isotopomers.

We fit 69 *ab initio* points calculated by Meyer *et al.* (MBB-$H_3^+$) using the symmetric NN method and utilize the internal nuclear distances as our coordinate system. Here, we only need the original 69 MBB-$H_3^+$ *ab initio* points, instead of the 274 used to fit the Dykstra–Swope energies. The neural network structure used is $3-15b-1l$ totalling 76 connection weights. Table II has the convergence information, the optimized connection weights are shown in the Table II. Figure 4 shows our fitting of the MBB-$H_3^+$ PES using the symmetric NN, the difference between the symmetric neural network (SNN) PES, and the MMB potential[3] is plotted in Fig. 5.

To obtain the SNN-PES we need less connection weights and *ab initio* points than in the UNN-PES. Another important feature of the SNN-PES is that the symmetry is a property, while for UNN-PES we introduced the symmetry *a posteriori*. Thus the SNN-PES consumes less CPU time than the UNN-PES.

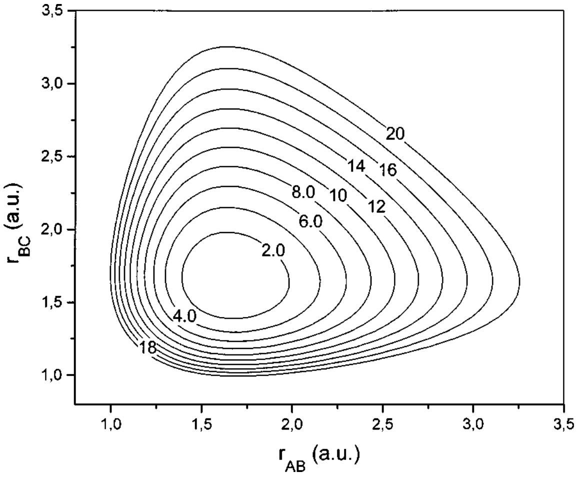

FIG. 4. SNN fitting of the Meyer *et al.* *ab initio* $H_3^+$ energies. The labels in the contour plots are given in units of 1000 cm$^{-1}$.

### C. Vibrational levels calculation

To test the PES's we performed calculations of the vibrational levels of the $H_3^+$ ion using the CFQMC discussed in Sec. III. For the sake of comparison we performed identical simulations for all three PES tested, i.e., the trial wave functions, the initial random number seed, and positions of the nuclei and the length of the simulation were identical. The parameters of the trial wave function [Eq. (33)] were the same and we used the same number of excited trial wave functions, in a total of 455 basis function in the variational Monte Carlo calculation. For the projection of the final states [Eqs. (23), (24), (25)] we kept the 50 states with lowest energy, and obtained the spectra shown in Table III.

In Table III we compare the results of the CFQMC calculations of the vibrational levels of the $H_3^+$ ion for the two PES's fitted in this work; the usual neural network (UNN) fitting of the Dykstra–Swope *ab initio* points; the symmetric neural network fitting of the Meyer *et al.* *ab initio* points and

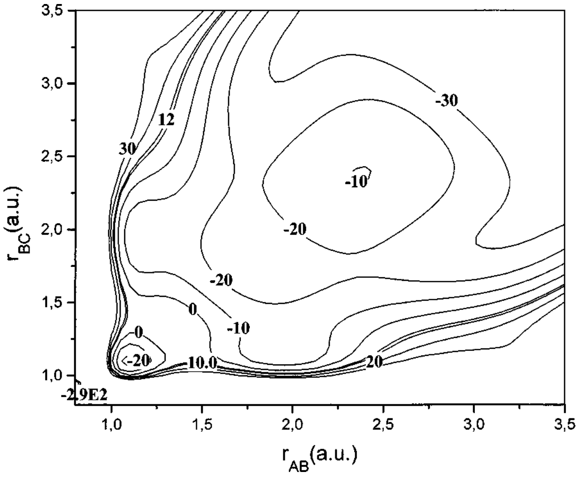

FIG. 5. Difference between SNN-PES and the MBB-PES. The labels in the contour plots are given in cm$^{-1}$.

TABLE III. Vibrational states of $H_3^+$ for the UNN, SNN, and MBE PES calculated using CFQMC. All values are expressed in $cm^{-1}$.

| *i* | UNN | SNN | MBB | EXP (Ref. 31) |
|---|---|---|---|---|
| 1 | 2522.9 | 2519.8 | 2523.1 | 2521.4 |
| 2 | 3175.1 | 3172.8 | 3173.1 | 3178.4 |
| 3 | 4782.0 | 4773.4 | 4782.0 | |
| 4 | 5002.0 | 4994.2 | 5001.2 | 4997.965 |
| 5 | 5555.1 | 5547.1 | 5550.0 | 5554.063 |
| 6 | 6258.8 | 6258.0 | 6249.2 | |
| 7 | 7014.5 | 7000.4 | 7013.1 | |
| 8 | 7297.2 | 7279.0 | 7292.6 | |
| 9 | 7499.4 | 7486.5 | 7497.2 | 7492.91 |
| 10 | 7772.6 | 7762.9 | 7767.4 | |
| 11 | 7878.0 | 7861.8 | 7868.5 | |
| 12 | 8487.3 | 8476.7 | 8471.6 | |
| 13 | 9027.1 | 9008.5 | 9023.6 | |
| 14 | 9142.6 | 9122.8 | 9138.0 | |
| 15 | 9237.7 | 9241.2 | 9215.3 | |
| 16 | 9671.2 | 9648.0 | 9663.1 | |
| 17 | 10 001.9 | 9983.7 | 9992.0 | |
| 18 | 10 011.9 | 9994.1 | 10 001.4 | |
| 19 | 10 214.6 | 10 190.3 | 10 200.6 | |
| 20 | 10 595.1 | 10 580.8 | 10 577.3 | |
| 21 | 10 662.7 | 10 641.5 | 10 641.2 | |
| 22 | 11 033.3 | 11 014.7 | 11 028.8 | |
| 23 | 11 133.1 | 11 116.4 | 11 127.8 | |
| 24 | 11 324.5 | 11 314.0 | 11 295.8 | |
| 25 | 11 552.6 | 11 528.8 | 11 545.7 | |
| 26 | 11 816.8 | 11 790.7 | 11 803.3 | |
| 27 | 11 927.0 | 11 910.7 | 11 914.5 | |
| 28 | 12 128.2 | 12 139.8 | 12 090.7 | |
| 29 | 12 212.2 | 12 187.1 | 12 195.5 | |
| 30 | 12 242.9 | 12 219.4 | 12 230.8 | |
| 31 | 12 405.3 | 12 380.9 | 12 389.5 | |
| 32 | 12 765.8 | 12 749.2 | 12 744.3 | |
| 33 | 12 817.2 | 12 788.2 | 12 793.0 | |

the power order fitting PES of Meyer *et al.* of their own *ab initio* points (MBB). We compared the results for each of the surfaces with the available experimental results (1st, 2nd, 4th, 5th, and 9th levels in Table III). The average difference of the UNN, SNN, and MBB values with these experimental values were 3.3, 4.8, and 3.7 $cm^{-1}$, respectively. Although the SNN PES had the larger difference they had the most consistent result, as all the values were smaller than the experiment. The MBB results oscillated the most with three values above the experiment and two below. The UNN had the smallest average difference with experiment and were very consistent, having four levels above the experimental values and only one of them below, the second excited state. One can see that all three PES's lead to the second excited state energy lower than experiment. This suggests that the problem could be in the *ab initio* energies and not the fitting. The overall accuracy of all three PES's with the experimental values were of the same order and smaller than 5 $cm^{-1}$. This is a remarkable result taking in account that there are differences in the fittings ranging from $-30$ to 30 $cm^{-1}$ in some regions. The results in Table III show that the fitting of PES's using neural networks is of the same quality as other procedures, at least for the lower excited states. For the higher excited states (up to 12 900 $cm^{-1}$) the differences were larger for some states but in average the results between the UNN and SNN differ by 15.6 $cm^{-1}$, between the UNN and MBB differ by 11.1 $cm^{-1}$, and between the SNN and MBB differ by 10.6 $cm^{-1}$. It is important to remember that the MBB PES contains corrections after the fitting of the *ab initio* points in order to better reproduce some vibrational levels, while in the current work no such corrections are made.

## V. CONCLUSIONS AND FINAL REMARKS

In this paper we reported the use of neural networks to fit the PES of the $H_3^+$ ion from *ab initio* energies. We fit the set of energies of Dykstra and Swope using the usual backpropagation neural network formulation. Because of the invariance upon permutation of two nuclei we have proposed a symmetric neural network formulation and used it to fit the Meyer *et al.* energies. The main advantage of the symmetric formulation is that the symmetry is a property of the PES. This property allows us to use a smaller set of *ab initio* points and connection weights. Thus the computation of the energy for a given geometry is faster for the SNN-PES than for the UNN-PES.

In Figs. 2 and 4 the two neural network fittings are almost identical. But Figs. 3 and 5 show that there are regions where the differences of these fittings with the MBB-PES are large, ranging from $-30$ to 30 $cm^{-1}$. These results suggest that it may be a good idea to compute *ab initio* energies at some points in these regions, improve the fit, and making them less dependent on the functional forms used to fit the PES's.

We also report the results of CFQMC calculations of the vibrational levels for each of these PES's. The comparison of these results with experiment shows that the average difference is of the same order for all three surfaces (UNN, SNN, and MBB), being smaller than 5 $cm^{-1}$ for the five levels compared.

We conclude that the neural network fitting of PES's is of the same quality as the power order fitting of Meyer *et al.* The neural network has the advantage of not requiring previous knowledge of the shape of the surface and requires a small set of *ab initio* points for the fit. In addition, the generalization to larger molecules does not require major modification of the method. Thus the neural network approach is a good candidate for use in fitting of many-dimensional PES's.

## ACKNOWLEDGMENTS

This work has been supported by CNPq through grants to the FVP and JJSN. The computational work was supported through grants prjscd (PHA) and prjcds (FVP) of CENAPAD-MG/CO.

[1] T. Oka, Phys. Rev. Lett. **45**, 531 (1980).
[2] G. D. Carney and R. N. Porter, J. Chem. Phys. **60**, 4251 (1974).
[3] W. Meyer, P. Botschwina, and P. Burton, J. Chem. Phys. **84**, 891 (1986).
[4] W. Cencek, J. Rychlewsk, R. Jaquet, and W. Kutzelnigg, J. Chem. Phys. **108**, 2831 (1998); R. Jaquet, W. Cencek, W. Kutzelnigg, and J. Rychlewski, J. Chem. Phys. **108**, 2837 (1998).
[5] C. E. Dykstra and W. C. Swope, J. Chem. Phys. **70**, 1 (1979).
[6] G. D. Carney, S. M. Adler-Golden, and D. C. Lesseski, J. Chem. Phys. **84**, 3921 (1986).

[7] E. Garcia and A. Laganà, Mol. Phys. **56**, 621 (1985); , 629 (1985).

[8] G. Simons, R. G. Parr, and J. M. Finlan, J. Chem. Phys. **59**, 3229 (1973).

[9] D. E. Makarov and H. Metiu, J. Chem. Phys. **108**, 590 (1998).

[10] G. S. Kedziora and I. Shavitt, J. Chem. Phys. **106**, 8733 (1997).

[11] K. C. Thompson, M. J. T. Jordan, and M. A. Collins, J. Chem. Phys. **108**, 564 (1998).

[12] D. F. R. Brown, M. N. Gibbs, and D. C. Clary, J. Chem. Phys. **105**, 7597 (1996).

[13] F. V. Prudente and J. J. Soares Neto, Chem. Phys. Lett. **287**, 585 (1998).

[14] J. Zupan and J. Gasteiger, *Neural Network for Chemists* (VHC, Weinheim, 1993).

[15] B. G. Sumpter, C. Getino, and D. W. Noid, Annu. Rev. Phys. Chem. **45**, 439 (1994).

[16] J. A. Darsey, D. W. Noid, and B. R. Upadhyaya, Chem. Phys. Lett. **177**, 189 (1991).

[17] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Comput. Phys. Commun. **104**, 1 (1997).

[18] A. P. Braga, J. P. Braga, and J. C. Belchior, J. Chem. Phys. **107**, 9954 (1997).

[19] G. M. Silva, P. H. Acioli, and A. C. Pedroza, J. Comput. Chem. **18**, 1407 (1997).

[20] R. P. Brent, IEEE Trans. Neural Netw. **2**, 346 (1991).

[21] M. F. Moller, Neural Networks **6**, 525 (1993).

[22] T. B. Blank and S. D. Brown, J. Chemometrics **8**, 391 (1994); T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, J. Chem. Phys. **103**, 4129 (1995).

[23] D. J. C. Mackay, Neural Comput. **4**, 415 (1992); **4**, 448 (1992).

[24] D. J. C. MacKay, Netw.: Comp. Neural Systems **6**, 469 (1995).

[25] D. M. Ceperley and B. Bernu, J. Chem. Phys. **89**, 6316 (1988); B. Bernu, D. M. Ceperley, and W. A. Lester, Jr., *ibid.* **93**, 552 (1990).

[26] P. H. Acioli and J. J. Soares Neto, J. Mol. Struct.: THEOCHEM (accepted for publication).

[27] E. D. Sontag, Neural Comput. **1**, 470 (1989).

[28] C. M. Bishop and C. M. Roach, Rev. Sci. Instrum. **63**, 4450 (1982).

[29] P. J. Reynolds, D. M. Ceperley, B. J. Alder, and W. A. Lester, Jr., J. Chem. Phys. **77**, 5593 (1982).

[30] D. Frye, A. Preiskorn, G. C. Lie, and E. Clementi, J. Chem. Phys. **92**, 4948 (1990).

[31] Experimental values taken from references contained in Ref. 4.