

New methods for calculating  
vibrational energy levels of molecules

Xiao-Gang Wang

and

Tucker Carrington Jr.

Université de Montréal, Canada

BIRS Workshop on Matrix  
Eigenvalue Problems

From eigenvalues chemists compute spectra and rate constants.

## Spectra

- shapes of molecules
- forces that hold molecules together

## Rate constants

- atmospheric chemistry
- combustion chemistry

To compute a ro-vibrational spectrum one must solve the Schroedinger equation

$$\hat{H}\psi_n = E\psi_n . \quad (1)$$

This is almost always done by converting it into a matrix eigenvalue problem

$$\psi_n = \sum_k c_{nk} \phi_k \quad (2)$$

$$(\hat{H} - E) \sum_k c_{nk} \phi_k = 0 \quad (3)$$

Galerkin

$$(\mathbf{H} - E\mathbf{I})\phi = 0 \quad (4)$$

To follow this procedure one needs to:

- determine  $\hat{H}$
- choose basis functions  $\phi_k$
- calculate matrix elements
- compute eigenvalues and eigenvectors

$\hat{H}$  depends on orientational and shape coordinates

$H(r_1, r_2, \theta, \text{orientational coordinates})$

$$\hat{H} = \hat{T} + \hat{V} \quad (5)$$

$\hat{T}$  = differential operator

E.g.

$$\hat{T} = \alpha \frac{\partial^2}{\partial r_1^2} + \beta \frac{\partial^2}{\partial r_2^2} + \rho \frac{\partial^2}{\partial \theta^2} + \dots \quad (6)$$

$\hat{V}$  = function of the shape coordinates

E.g.  $\hat{V} = V(r_1, r_2, \theta)$

The first basis functions used in this field were product basis functions. For H<sub>2</sub>O

$$\phi_k(r_1, r_2, \theta) = \gamma_{n_1}(r_1)\psi_{n_2}(r_2)\chi_{n_3}(\theta) \quad (7)$$

and  $k=(n_1, n_2, n_3)$

In this basis the Hamiltonian matrix elements are

$$(\mathbf{H})_{\mathbf{n}'_1 \mathbf{n}'_2 \mathbf{n}'_3, \mathbf{n}_1 \mathbf{n}_2 \mathbf{n}_3} \quad (8)$$

For H<sub>2</sub>O the size of the matrix required for the vibrational problem is

$$(number\ of\ n_1)(number\ of\ n_2)(number\ of\ n_3) = n_{1d}^3 \quad (9)$$

$3N_{at} - 6$  coordinates are required to specify the shape of a molecule with  $N_{at}$  atoms.

For a molecule with  $N_{at}$  atoms the size of the matrix is  $n^{3N_{at}-6}$

$$n_{1d} \approx 10 \quad (10)$$

For a four-atom molecule the size of the matrix is  $\approx 10^6$ .

For a five-atom molecule the size of the matrix is  $\approx 10^9$ .

It is not possible to use Householder transformations and the QR algorithm to compute the eigenvalues of a  $10^6 \times 10^6$  matrix.

The two most obvious options are

- Better basis functions
- Iterative methods

The first option: Better basis functions

If there were only two coordinates

Diagonalise blocks

$$\text{basis} \rightarrow \gamma_{n_1}(q_1) f_{m_2}^{n_1}(q_2) \quad (11)$$

Retain only some of the eigenvectors of each block.

We call these contracted basis functions.

If elements of the off-diagonal blocks are small compared to differences of eigenvalues of the diagonal blocks then this contracted basis is good (much smaller than the original basis).

## Examples

H<sub>2</sub>O

5000 product → 800 contracted

HCN/HNC

9000 product → 1000 contracted

The second option: Use a product basis and calculate eigenvalues with an iterative method

E.g. Lanczos

- No need to store  $\mathbf{H}$

Matrix-vector products are evaluated by exploiting the structure of the product basis (not the sparsity of the matrix).

# Basis Function Dichotomy

Product

- Basis functions are products of functions of a single variable

Contracted

- Basis functions are eigenfunctions of blocks of the Hamiltonian matrix

- $\psi_{n_1 n_2 n_3} = \theta_{n_1}(r_1) \phi_{n_2}(r_2) \chi_{n_3}(\theta)$
- $\psi_{j n_3} = \rho_j^{n_3}(r_1, r_2) \chi_{n_3}(\theta)$

## Advantages of product basis

- Basis functions are simple and need not be determined numerically

## Advantages of contracted basis

- Fewer basis functions are required to represent wavefunctions of interest

# Eigensolver Dichotomy

## Direct

- Memory:  $N^2$
- CPU:  $N^3$
- Householder transformations + QR method

## Iterative

- Memory:  $N$
- CPU:  $M \cdot$ 

cost of a single matrix-vector product
----------------------------------------
- Lanczos, Filter Diagonalisation, Davidson

In the product basis the Hamiltonian matrix is a sum of “tensor products”.

Because the simplicity of the product basis makes computing matrix-vector products inexpensive it is natural to combine a product basis with an iterative eigensolver.

This yields a powerful combination

- Memory cost  $\sim N = (n_{1d})^f$
- CPU cost  $\sim MN \log N \approx M(n_{1d})^{f+1}$

Because the compactness of the contracted basis reduces  $N^2$  and  $N^3$ , it is natural to combine a contracted basis with a direct eigensolver.

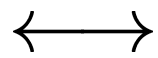
This combination is also advantageous.

Basis

Eigensolver

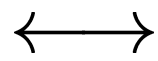


Product



Iterative

Contracted



Direct

The success of the contracted/direct strategy is due to the small size of the contracted basis.

HCN/HNC	H <sub>2</sub> O
1000 < 9000	800 < 5000

The success of the product/iterative strategy is due to the CPU cost scaling relation,

$$N^3 \quad vs. \quad M \cdot \boxed{\text{cost of a single matrix-vector product}}$$
$$< MN^2$$
$$< N^3$$

$(N_{cntcd})^2$  and  $(N_{cntcd})^3$  are significant problems for those who use contracted/direct strategies.

For a molecule with four atoms  $N_{cntcd} \approx 10'000$

For a molecule with five atoms  $N_{cntcd} \gg 10'000$

$N_{prod}$  and  $n_{1d}N_{prod}$  are significant problems for those who use product/iterative strategies.

For a molecule with five atoms  $N_{prod} \approx 10^9$

## **Proposal:**

Consider the possibility of exploiting *both* the smallness of the contracted basis *and* the efficiency of iterative solvers.

## **Objections:**

- An iterative eigensolver will only be efficient if computing matrix-vector products is inexpensive and contracted basis functions make this impossible
- To do matrix-vector products in a contracted basis it is natural to transform back to the product basis but this obviates the storage advantage of the contracted basis.

Contracted basis functions obtained by diagonalising blocks are good in the sense that they reduce the size of the Hamiltonian matrix, but to do matrix-vector products in this basis one must transform from the contracted basis to a primitive basis, evaluate the matrix-vector product in the primitive basis, and then transform back to the contracted basis. *There is therefore no way to avoid storing vectors with as many components as there are primitive basis functions.*

For H<sub>2</sub>O the difficult matrix-vector product is

$$\begin{aligned}
 z_{n'_3 j'} &= \sum_{n_3 j} \langle n'_3 j' | \mu(r_1, r_2) \hat{O}(\theta) | n_3 j \rangle u_{n_3 j} \\
 &= \sum_{\alpha\beta} C_{\alpha\beta j'}^{n'_3} \mu(r_{1\alpha}, r_{2\beta}) \sum_{n_3} O_{n'_3 n_3} \sum_j C_{\alpha\beta j}^{n_3} u_{n_3 j}
 \end{aligned}$$

which could be evaluated

$$= \sum_{n_3} \sum_j M_{j'j}^{n'_3, n_3} O_{n'_3 n_3} u_{n_3 j}$$

with

$$M_{j'j}^{n'_3, n_3} = \sum_{\alpha\beta} C_{\alpha\beta j'}^{n'_3} \mu(r_{1\alpha}, r_{2\beta}) C_{\alpha\beta j}^{n_3}$$

but this matrix vector product is costly (because **M** depends on 4 indices) and storing **M** requires a lot of memory.

To facilitate the calculation of matrix-vector products we use basis functions that are not as good (in the sense of reducing the size of the Hamiltonian matrix) as the functions obtained by diagonalising blocks.

Our basis functions have advantages

- They are inexpensive to compute. It is not necessary to diagonalise many large blocks.
- Hamiltonian matrix-vector products can be done efficiently without storing large vectors or matrices

## New contracted basis

$$H = T_{ben}(\theta, r) + T_{str}(r) + V(\theta, r)$$

We use as basis functions  $X_b(\theta)$   $Y_s(r)$  where

$$H^{(b)} X_b(\theta) = E_b X_b(\theta)$$

with

$$H^{(b)} = T_{ben}(\theta, r_e) + V(\theta, r_e).$$

and

$$H^{(s)} Y_s(r) = E_s Y_s(r)$$

with

$$H^{(s)} = T_{str}(r) + V(\theta_e, r).$$

We write the Hamiltonian,

$$H = H^{(b)} + H^{(s)} + \Delta T + \Delta V$$

where

$$\Delta V(r, \theta) = V(r, \theta) - V(r_e, \theta) - V(r, \theta_e)$$

Matrix-vector products with  $\Delta T$  are easy and inexpensive (because it is factorizable).

Matrix-vector products with  $\Delta V$

We choose the primitive stretch basis so that matrix representations of functions of the stretch coordinates are diagonal.

$$\langle b's' | \Delta V(\theta, r) | bs \rangle = \sum_{\beta, \alpha} C_{\beta b'} D_{\alpha s'} \Delta V_{\beta \alpha} D_{\alpha s} C_{\beta b}$$

where

$$X_b(\theta) = \sum_{\beta} C_{\beta b} f_{\beta}(\theta)$$

and

$$Y_s(r) = \sum_{\alpha} D_{\alpha s} g_{\alpha}(r)$$

$$\langle b's' | \Delta V(\theta, r) | bs \rangle = \sum_{\alpha} D_{\alpha s'} F_{b'b, \alpha} D_{\alpha s}$$

The potential matrix in the contracted basis is

For a large calculation this is so big that it cannot be stored.

Instead we write,

$$\langle b's' | \Delta V(\theta, r) | bs \rangle = \sum_{\alpha} D_{\alpha s'} F_{b'b, \alpha} D_{\alpha s}$$

with

$$F_{b'b, \alpha} = \sum_{\beta} C_{\beta b'} C_{\beta b} \Delta V_{\beta \alpha} .$$

$F$  is stored and the matrix-vector product is evaluated in three steps

$$\begin{aligned} u_{b\alpha}^{(1)} &= \sum_s D_{\alpha s} \\ u_{bs} u_{b'\alpha}^{(2)} &= \sum_b F_{b'b\alpha} u_{b\alpha}^{(1)} \\ u_{b's'}^{(2)} &= \sum_{\alpha} D_{\alpha s'} u_{b'\alpha}^{(2)} \end{aligned}$$

The F matrix is

This is easy to store

$$Cost \approx n_\alpha(n_b^2 + 2n_b n_s) \rightarrow n_\alpha n_b^2 \quad (n_b > n_s)$$

$$\ll n_{1d}^{f+1}$$

$$\ll n_\alpha n_\beta n_b$$

$F$  is calculated **once**,  
*before* the Lanczos recursion.

## Results

T8 potential of Schwenke and Partridge.

Basis	$l_{max} = m_{max}$	$n_{bend}$	$E_b^{cut}$	$n_b$	$n_i$	$n_{stretch}$	$E_s^{cut}$	$n_s$	$n_{final}$
Basis I	25	3.26M	8090	280	10	5049	20000	260	72800
Basis II	25	3.26M	7670	246	10	5049	20000	260	63960

$$33 \times 10^9 \rightarrow 72 \times 10^3$$

reduction of almost six orders of magnitude

Other subjects I look forward to discussing

Preconditioned inexact spectral transform  
method

- Goal: calculate many interior eigenvalues
- I apply Lanczos to  $(\sigma \mathbf{I} - \mathbf{H})^{-1}$ . I use preconditioning and TFQMR to solve the linear equations.
- I project onto the basis of the inexact Lanczos vectors. The quality of the eigenvalues I compute is not limited by the inexactness of the linear solves.
- Not Jacobi-Davidson

## Symmetric indefinite Lanczos method

- $\mathbf{A} \mathbf{U} = \mathbf{S} \mathbf{U} \mathbf{E}$
- Both  $\mathbf{A}$  and  $\mathbf{S}$  are indefinite
- Numerical error makes it impossible to calculate accurate eigenvalues

This work has been supported by the  
Natural Sciences and Engineering  
Research Council of Canada.