Pruned-basis variational methods for solving the vibrational Schroedinger equation without approximating the potential

Gustavo Avila and Tucker Carrington

Chemistry Department
Queen's University

30 avril 2013

Objective

Compute a vibrational spectrum

- without approximating the potential
  - the potential is not re-represented as a sum-of-products
  - the potential is not re-represented as a sum of terms with one, two, etc coordinates
- without approximating the kinetic energy operator (KEO)
  - terms in the KEO are not neglected
  - coordinate-dependent functions in the KEO are not expanded
- Compute many levels

Solve the Schroedinger equation using a basis set

- represent wavefunctions with basis functions

$$\psi_k(\mathbf{q}) = \sum_n c_n^k \quad f_n(\mathbf{q})$$

- compute eigenvalues and eigenvectors of the Hamiltonian matrix

The most obvious basis functions are product functions :

$$f_{n_1,n_2,\cdots} = \phi_{n_1}(r_1)\phi_{n_2}(r_2)\cdots\phi_{n_D}(r_D)\cdots$$

Between 10 and 100 1-d functions required for each coordinate.

$\Rightarrow > 10^{3N-6}$ multi-d basis functions required.

The Hamiltonian matrix is

- too large to construct
- too large to store in memory
- too large to diagonalise

Lanczos Algorithm

$$\mathbf{H} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot \end{pmatrix} = \mathbf{T}$$

The cost of matrix-vector products is critical

- Eigenvalues and eigenvectors are computed by evaluating matrix-vector products.

Sequential summation makes using quadrature feasible

The cost of a matrix-vector product scales as n^{D+1}.

With a product basis and a product quadrature grid matrix-vector products are efficient

$$w_{l'm'} = \sum_{lm} V_{l'm',lm} \ x_{lm}$$

$$V_{l'm',lm} = \int d\theta \int d\phi F_{l'}(\theta) G_{m'}(\phi) V(\theta,\phi) G_m(\phi) F_l(\theta)$$

$$V_{l'm',lm} \approx \sum_{\beta\gamma} T_{l'\beta} \ Q_{m'\gamma} \ V(\theta_\beta,\phi_\gamma) \ Q_{m\gamma} \ T_{l\beta}$$

$$T_{l\beta} \sim F_l(\theta_\beta) \ , \qquad Q_{m\gamma} \sim G_m(\phi_\gamma)$$

$$w_{l'm'} = \sum_{lm} \sum_{\beta\gamma} \; T_{l'\beta} \; Q_{m'\gamma} \; V(\theta_\beta, \phi_\gamma) \; Q_{m\gamma} \; T_{l\beta} \;\; x_{lm}$$

$$w_{l'm'} = \sum_{\beta} \; T_{l'\beta} \sum_{\gamma} \; Q_{m'\gamma} \; V(\theta_\beta, \phi_\gamma) \sum_{m} \; Q_{m\gamma} \sum_{l} \; T_{l\beta} \;\; x_{lm}$$

The cost of each sum scales as n^{D+1}

The advantage of doing sums sequentially is a continuing theme

Reduce the basis size

The product basis functions are usually eigenfunctions of a zeroth-order Hamiltonian,

$$H = H_0 + \Delta$$

H_0 is a sum of 1d Hamiltonians (separable).

One can remove basis functions with large zeroth-order energies.

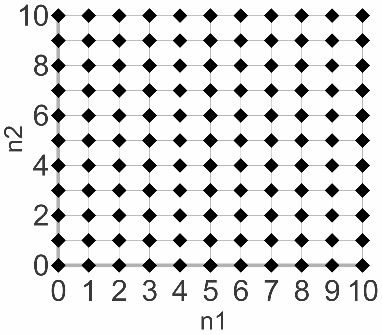
If all the 1d Hamiltonians are identical one simply removes basis functions for which

$$\sum_c n_c > b$$

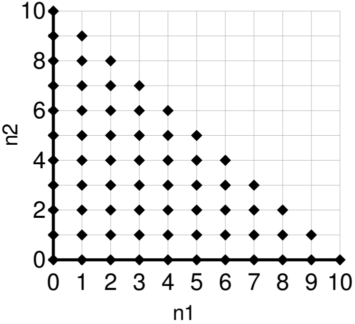
and retains only

$$\frac{(D+b)!}{D!b!}$$

Full basis for a 2d problem



Pruned basis for the 2d problem



If $3N - 6 = 15$ and 15 basis functions are used for each coordinate then the size of the direct product basis is 4×10^{17}.

By discarding all functions for which $\sum_c n_c > b = 15$ the size of the basis is reduced to 7.7×10^7.

Basis vector : 3×10^9 GB $\rightarrow$ 0.6 GB

For our problems this pruning scheme is much better than the "hyperbolic cross"

The size of the quadrature grid prevents one from solving difficult problems

With a product grid the required matrix-vector product is :

$$v2(n'_3, n'_2, n'_1) = \sum_{k_1=1}^{k_1^{\max}} T_{n'_1 k_1} \sum_{k_2=1}^{k_2^{\max}} T_{n'_2 k_2} \sum_{k_3=1}^{k_3^{\max}} T_{n'_3 k_3}$$
$$V(q_1^{k_1}, q_2^{k_2}, q_3^{k_3})$$
$$\sum_{n_3=0}^{n_3^{\max}} T_{n_3 k_3} \sum_{n_2=0}^{n_2^{\max}} T_{n_2 k_2} \sum_{n_1=0}^{n_1^{\max}} T_{n_1 k_1} \; v1(n_3, n_2, n_1)$$

The largest vectors have as many elements as there are quadrature points $\sim n^D$.

We need better quadrature

- For a 12D problem, a direct product quadrature has $\sim 15^{12}$ points. Storing one vector requires about 10^6 GB.

To do better we must

- Find a smaller quadrature grid that enables us to accurately compute all matrix elements
- Find such a grid that has enough structure that we can efficiently evaluate matrix-vector products

Smolyak grids

The Smolyak quadrature equation for integrating a function $g(x_1, x_2, \cdots, x_D)$ can be written as a sum of D-dimensional product quadrature grids,

$$S(D, H) = \sum_{i_1+i_2+\cdots \leq H} C_{i_1,\ldots,i_D} [Q^{i_1}(x_1) \otimes \cdots \otimes Q^{i_D}(x_D)],$$

Nested grids

Smolyak is most advantageous if one uses nested grids.

- The total number of points is smaller
- Enough structure to make efficient matrix-vector products possible.

Choice of 1d quadratures

Our 1d basis functions are not hat functions.

Two point and weight selection schemes :

- Choose points and weights to maximize the degree exactly integrated
- Choose points and weights to make the (1d) overlap (Gramian) matrix as close as possible to the identity.

How to implement Smolyak ?

A Smolyak grid will have fewer points.

However, the Smolyak quadrature involves not only sums over quadrature points, but also a sum over contributing grids :

$$S(D,H) = \sum_{i_1+i_2+\cdots\leq H} C_{i_1,\ldots,i_D}[Q^{i_1}(x_1)\otimes\cdots\otimes Q^{i_D}(x_D)],$$

Although doing the sums for each contributing grid and then adding the results is costly, this is frequently done.

Incorporate the sum over contributions into weights

A Smolyak quadrature can be written,

$$S(6,H)f(q_1,q_2,q_3,q_4,q_5,q_6)$$
$$= \sum_{k_1}^{N_1} \sum_{k_2}^{N_2} \sum_{k_3}^{N_3} \sum_{k_4}^{N_4} \sum_{k_5}^{N_5} \sum_{k_6}^{N_6} w(k_6,k_5,k_4,k_3,k_2,k_1)$$
$$\times f(q_1^{k_1}, q_2^{k_2}, q_3^{k_3}, q_4^{k_4}, q_5^{k_5}, q_6^{k_6})$$

with N_c a maximum number of points for coordinate q_c. In this equation N_i depends on k_j if $i > j$.

$$w(k_6,\cdots,k_1) = \sum_{i_1+i_2+\cdots\leq H} C_{i_1,\cdots,i_6} \, {}^{i_1}w_{k_1} \cdots {}^{i_D}w_{k_6},$$

where ${}^{i_c}w_{k_c}$ is the (1D) weight for the point $q_c^{k_c} \in Q^{i_c}$, and ${}^{i}w_k = 0$ if $q^k \notin Q^i$.

It is possible to find expressions for the N_c

Let $N_j = N(j)$

N_1 is independent of $k_{c \neq 1}$

$$N_1 = N\left((H - (D - 1))\right)$$

The other maxima are

$$N_c = N\left(H - \sum_{i=1}^{c-1} g(k_i) - (D - c)\right)$$

where $c = 2, 3, \cdots, 11$ and $g(k)$ is the smallest quadrature rule in the sequence of quadratures that contains point k.

Now the matrix-vector product

The potential matrix-vector product is computed by doing sums sequentially,

$$v2(n'_3, n'_2, n'_1) = \sum_{k_1=1}^{N_1} T_{n'_1 k_1} \sum_{k_2=1}^{N_2} T_{n'_2 k_2} \sum_{k_3=1}^{N_3} T_{n'_3 k_3}$$
$$w(k_3, k_2, k_1) V(q_1^{k_1}, q_2^{k_2}, q_3^{k_3})$$
$$\sum_{n_3=0}^{n_3^{\max}} T_{n_3 k_3} \sum_{n_2=0}^{n_2^{\max}} T_{n_2 k_2} \sum_{n_1=0}^{n_1^{\max}} T_{n_1 k_1}$$
$$v1(n_3, n_2, n_1) \ ,$$

where $T_{nk} = A_n H_n(q_k)$

Crucial advantage that there is no sum over contributing grids

Improve pruning of the basis and the grid to exploit properties of the potential

Retain only basis functions that are coupled by the largest terms in the expansion,

$$V(x_1, \cdots, x_{12}) = \sum_{i=1}^{12} V_i^1(x_i) + \sum_{i=1}^{12} \sum_{j\leq i}^{12} V_{i,j}^2(x_i, x_j) + \sum_{i=1}^{12} \sum_{j\leq i}^{12} \sum_{k\leq j}^{12} V_{i,j,k}^3(x_i, x_j, x_k) + \cdots$$

We wish to :

- Find a quadrature adapted for computing matrix-elements with this basis
- Implement the quadrature without jeopardizing the efficiency of the matrix-vector products

How is it possible to prune in this fashion and do matrix-vector products efficiently ?

Instead of

$$n_1 + \cdots + n_D \leq b$$

we use

$$g(n_1) + \cdots + g(n_D) \leq b \ ,$$

where $g(n)$ is chosen so that basis functions with many non-zero n_k are excluded

We use a quadrature with which all overlap matrix elements are exact,

$$S(D,H) = \sum_{i_1} \cdots \sum_{i_D} C_{i_1,\cdots,i_D} Q^{i_1}(x_1) \otimes Q^{i_D}(x_D)$$

with the condition

$$g(i_1) + \cdots + g(i_D) \leq H$$

Taking $g(0) = 0$ and $g(n) = n + s(n)$, for $n > 0$, where $s(n)$ is some positive shift, will yield a smaller basis

- Basis functions with two non-zero indices, n_i and n_j will only be included if $n_i + n_j \leq b - s(n_i) - s(n_j)$;
- Basis functions with three non-zero indices, n_i, n_j, and n_k, will only be included if $n_i + n_j + n_k \leq b - s(n_i) - s(n_j) - s(n_k)$.

We use

$$g(0) = 0, \qquad g(n) = n + 0.2, \;\; n > 0$$

Using $g(n)$ further reduces basis and grid sizes

- Vibrational levels of C_2H_4.
- The size of the basis set is determined by $g(n_1) + \cdots + g(n_D) \leq 9$.
  Basis size is 1.6×10^5
- Smolyak grid with $H = 25$ with Grid size 8.5×10^6

  Product Gauss grid of comparable accuracy has $\sim 10^{14}$ points

Compare our levels with those obtained by approximating the potential

Assig	4−MM	full−MM	MULTIMODE
Ground	0.29	0.81	11003.98
ν_{10}	−0.39	0.64	821.15
ν_8	−0.29	0.75	926.13
ν_7	−0.29	0.80	946.49
ν_4	−0.31	0.67	1025.28
ν_6	−0.42	0.34	1223.46
ν_3	−0.46	0.02	1341.42
ν_{12}	−0.34	0.60	1439.67
ν_2	−0.38	0.51	1623.17
$2\nu_{10}$	−0.28	1.68	1654.22
$\nu_8 + \nu_{10}$	−1.12	1.38	1750.05
$\nu_7 + \nu_{10}$	−1.11	1.52	1774.18

Table: Energies computed with $b = 9$ compared with those obtained with multimode

Assig	4−MM	full−MM	MULTIMODE
$\nu_4 + \nu_{10}$	−1.06	1.42	1848.32
$2\nu_8$	−0.20	2.07	1854.04
$\nu_7 + \nu_8$	−1.21	3.89	1866.06
$2\nu_7$	−0.14	2.05	1893.58
$\nu_4 + \nu_8$	−0.83	3.67	1946.54
$\nu_4 + \nu_7$	−1.01	3.68	1961.19
$\nu_6 + \nu_{10}$	−0.56	2.30	2038.34
$2\nu_4$	−0.14	1.81	2047.54
$\nu_6 + \nu_8$	−1.00	0.98	2157.66
$\nu_3 + \nu_{10}$	−0.85	1.34	2164.88
$\nu_6 + \nu_7$	−1.07	1.05	2172.44
$\nu_4 + \nu_6$	−1.06	0.96	2249.87
$\nu_{10} + \nu_{12}$	−0.45	2.92	2256.73

Table: Energies computed with $b = 9$ compared with those obtained with multimode

Assig	4−MM	full−MM	MULTIMODE
$\nu_3+ \nu_8$	−0.94	0.87	2263.77
$\nu_3+ \nu_7$	−0.89	0.93	2285.37
$\nu_8+ \nu_{12}$	−0.80	1.86	2361.04
$\nu_3+ \nu_4$	−0.93	0.60	2363.20
$\nu_2+ \nu_{10}$	−0.85	2.06	2434.07
$\nu_7+ \nu_{12}$	−0.84	1.91	2379.82
$2\nu_6$	−0.26	1.36	2446.56
$\nu_4+ \nu_{12}$	−0.82	1.50	2467.01
$3\nu_{10}$	−0.63	2.90	2492.97
$\nu_2+ \nu_8$	−0.71	1.18	2542.01
$\nu_3+ \nu_6$	−0.93	0.73	2562.51
$\nu_2+ \nu_7$	−0.83	1.08	2564.04
$\nu_8+ 2\nu_{10}$	−1.65	2.85	2583.95
$\nu_7+ 2\nu_{10}$	−1.78	3.03	2610.83
$\nu_2+ \nu_4$	−0.91	1.17	2645.42

Table: Energies computed with $b = 9$ compared with those obtained with multimode

Assig	4−MM	full−MM	MULTIMODE
$2\nu_8 + \nu_{10}$	−1.75	2.60	2682.68
$2\nu_7 + \nu_{10}$	−1.73	2.77	2728.67
$\nu_3 + \nu_{12}$	−0.90	1.19	2774.10
$3\nu_8$	−0.55	3.77	2782.78
$\nu_7 + 2\nu_8$	−1.93	7.65	2786.59
$2\nu_7 + \nu_8$	−1.62	6.90	2808.06
$\nu_2 + \nu_6$	−0.76	1.60	2831.56
$3\nu_7$	−0.24	3.09	2842.37
$\nu_6 + 2\nu_{10}$	−0.41	4.14	2864.92
$\nu_4 + 2\nu_8$	−1.47	8.13	2868.85
$2\nu_{12}$	−0.10	1.96	2870.54
$2\nu_4 + \nu_{10}$	−1.25	2.87	2873.11
$\nu_4 + 2\nu_7$	−1.36	6.54	2899.08
$2\nu_4 + \nu_8$	−0.92	7.13	2964.47
$2\nu_4 + \nu_7$	−1.52	7.29	2973.80

Table: Energies computed with $b = 9$ compared with those obtained with multimode

Assig	4−MM	full−MM	MULTIMODE
ν_1	−0.36	−0.24	3020.15
$3\nu_4$	0.27	3.16	3067.66
$\nu_2 + \nu_{12}$	−0.46	3.13	3070.71
ν_5	−0.88	−2.53	3079.79
$2\nu_{10} + \nu_{12}$	−0.62	3.47	3092.90
$\nu_6 + 2\nu_7$	−1.42	2.24	3122.88
$\nu_3 + 2\nu_8$	−0.42	2.65	3188.39
$\nu_3 + 2\nu_7$	−1.02	2.29	3230.61
$2\nu_2$	−0.32	2.81	3236.77
$2\nu_6 + \nu_{10}$	0.29	5.16	3255.55
$\nu_2 + 2\nu_{10}$	−0.57	3.67	3266.22
$2\nu_4 + \nu_6$	−1.21	2.48	3273.40
$2\nu_8 + \nu_{12}$	−0.88	3.99	3284.30
$2\nu_7 + \nu_{12}$	−0.72	3.88	3320.95
$4\nu_{10}$	0.37	4.91	3341.94

Table: Energies computed with $b = 9$ compared with those obtained with multimode

Assig	4−MM	full−MM	MULTIMODE
$2\nu_6 + \nu_8$	−0.77	2.29	3389.20
$\nu_8 + 3\nu_{10}$	−0.81	5.33	3426.54
$\nu_7 + 3\nu_{10}$	−0.89	5.73	3456.41
$\nu_2 + 2\nu_8$	−0.45	2.83	3462.25
$\nu_4 + 2\nu_6$	−0.91	2.37	3474.25
$2\nu_4 + \nu_{12}$	−0.54	3.19	3491.55
$\nu_2 + 2\nu_7$	−0.58	2.49	3504.74
$2\nu_3 + \nu_{10}$	0.44	3.61	3506.55
$2\nu_8 + 2\nu_{10}$	−0.11	6.06	3520.06
$\nu_4 + 3\nu_{10}$	−0.88	5.23	3524.68
$2\nu_7 + 2\nu_{10}$	−0.28	6.41	3571.53
$3\nu_8 + \nu_{10}$	−0.46	5.52	3618.23
$\nu_2 + 2\nu_4$	−0.49	2.74	3664.92
$3\nu_6$	0.52	3.06	3669.90
$\nu_{10} + 2\nu_{12}$	0.18	6.39	3684.05

Conclusion

- We solve a 12-D Schroedinger equation, *without approximating the potential to induce sparsity of the potential matrix*
- It is possible to efficiently evaluate matrix-vector products with a pruned basis and a Smolyak quadrature grid
- We obviate the need to add contributions from different grids.
- Grid size $\sim 5.7 \times 10^{13} \rightarrow 8.5 \times 10^{6}$
  memory cost 500 TB $\rightarrow$ 0.07 GB
- Very efficient parallelization

This work has been supported by the
Natural Sciences and Engineering
Research Council of Canada, the Réseau québécois de calcul
de haute performance, and the Canada Research Chairs
programme.

4-mode and 5-mode representations have different energies

Assig	Δ4-Bench	Δ5-Bench	Bench
ν_{10}	-1.1	-0.1	821.7
ν_6	-0.8	0.0	1223.8
ν_3	-0.5	0.0	1341.4
ν_{12}	-1.0	-0.1	1440.2
ν_2	-1.0	-0.1	1623.6
ν_{10}	-2.2	-0.2	1655.5
$2\nu_8$	-2.6	-0.3	1855.6
$\nu_7 + \nu_8$	-5.6	-0.6	1869.3
$2\nu_7$	-2.5	-0.3	1895.1
$\nu_4 + \nu_8$	-4.9	-0.5	1949.7
$\nu_4 + \nu_7$	-5.1	-0.5	1964.3
$\nu_6 + \nu_{10}$	-3.2	-0.2	2040.3

Assig	Δ4-Bench	Δ5-Bench	Bench
$2\nu_4$	-2.2	-0.3	2048.9
$\nu_3 + \nu_{10}$	-2.5	-0.2	2165.8
$\nu_{10} + \nu_{12}$	-3.8	-0.3	2259.2
$\nu_2 + \nu_{10}$	-3.3	-0.3	2435.6
$2\nu_6$	-1.8	-0.1	2447.6
$3\nu_{10}$	-4.2	-0.4	2494.3
$\nu_3 + \nu_6$	-1.9	-0.1	2562.9
$\nu_6 + \nu_{12}$	-2.6	-0.2	2659.3
$2\nu_3$	-2.7	-0.1	2680.9
$\nu_4 + 2\nu_{10}$	-3.6	-0.5	2683.4
$\nu_7 + \nu_8 + \nu_{10}$	-10.3	-0.9	2699.4
$2\nu_7 + \nu_{10}$	-5.4	-0.5	2729.6
$\nu_3 + \nu_{12}$	-7.6	-0.2	2774.9
$\nu_4 + \nu_8 + \nu_{10}$	-3.4	-0.8	2775.9
$\nu_4 + \nu_7 + \nu_{10}$	-9.2	-0.8	2793.7
$\nu_2 + \nu_6$	-2.8	-0.2	2832.7
$\nu_6 + 2\nu_{10}$	-5.4	-0.5	2867.5
$2\nu_{12}$	-2.7	-0.2	2872.1