New strategies for solving the vibrational Schroedinger equation

Gustavo Avila, Jason Cooper, and Tucker Carrington

Chemistry Department
Queen's University

21 octobre 2009

How does one solve the Schroedinger equation ?

- represent wavefunctions with basis functions

$$\psi_n(\mathbf{r}, \boldsymbol{\theta}) = \sum_k c_k^n \quad f_k(\mathbf{r}, \boldsymbol{\theta})$$

- evaluate Hamiltonian matrix elements (integrals)
- compute eigenvalues and eigenvectors of the Hamiltonian matrix

The most obvious basis functions are product functions :

$$f_{k_1,k_2,\cdots} = \phi_{k_1}(r_1)\phi_{k_2}(r_2)\cdots\phi_{k_N}(\theta_1)\cdots$$

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

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

Diagonalization of the Hamiltonian matrix is very costly.

Both the CPU time [(basis size)^3] and the memory [(basis size)^2] scale very badly.

Better basis set methods must

- avoid diagonalization
- avoid direct product basis

Avoid diagonalization

Diagonalization can be avoided by using iterative methods

- Energy levels, intensities, rate constants, cross sections can be computed from time-independent methods that require only evaluating matrix-vector products
- Matrix-vector products can be done without storing the matrix
- Only a few vectors are stored
- Four-atom systems have been studied using these methods for more than a decade.

Is the direct product basis + iterative algorithm combination good enough ?

Advantages of product basis sets

- simplicity : very few parameters to optimise
- matrix-vector products can be computed efficiently

Disadvantages of product basis sets

- They are huge. For example, for CH_5^+ one would need a direct product basis of $\sim 20^{12}$ functions to compute hundreds of vibrational levels. For a single vector one needs 32×10^6 GB of memory.
- the size of vectors scales as n^D

Formaldehyde (1994) Methane (2009)
This remains a popular and powerful approach.

Matrix-vector products in a direct product basis – Sequential summation

Because the basis is huge it would be far too costly to form the potential matrix and explicitly multiply the matrix with vectors.

Matrix-vector products with direct product bases can be evaluated efficiently because sums over indices are independent.

To illustrate the computation of a matrix-vector product consider,

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

replace

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

If the potential is a sum of products, then for one term, $T(\theta)P(\phi)$, the matrix-vector product is of the form,

$$w_{l'm'} = \sum_{lm} \left(\int d\theta F_{l'}(\theta) T(\theta) F_l(\theta) \right) \left(\int d\phi G_{m'}(\phi) P(\phi) G_m(\phi) \right) x_{lm}$$

Move summation signs to the right,

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

where

$$t_{l'l} = \left(\int d\theta F_{l'}(\theta) T(\theta) F_l(\theta) \right); \quad p_{m'm} = \left(\int d\phi G_{m'}(\phi) P(\phi) G_m(\phi) \right)$$

If potential matrix elements are evaluated by quadrature

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

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

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

The largest vector is labelled by grid indices.

To study the dynamics of systems with more than five atoms

- Contracted basis functions (CH_4, CH_5^+)
- Prune a direct product basis, i.e., keep only the important basis functions.
  $n^D \longrightarrow N_r$

Isn't it trivial to discard less important basis functions ?

No, because the algorithm used to evaluate matrix-vector products in the direct product basis exploits the structure of the basis and much of this structure is lost when basis functions are discarded.

Sequential summation no longer works :

$$w_{l'm'} = \sum_{lm} t_{l'l} p_{m'm} \; x_{lm} \neq \sum_{l} t_{l'l} \sum_{m} p_{m'm} \; x_{lm}$$

Which basis functions should one keep ?

We retain basis functions whose diagonal matrix elements are below a threshold.

Sum of products Hamiltonian

If the Hamiltonian is a sum of products of a single variable

$$H = \sum_{l=1}^{g} \prod_{d=1}^{D} \hat{t}^{(d,l)}(q_k),$$

matrix-vector products can be evaluated efficiently, but term by term.

- Almost all KEOs are of this form
- Many potentials are also in this form,
- To use the multiconfiguration time-dependent Hartree (MCTDH) method one needs a potential of this form.

With this approach we can compute spectra of Hamiltonians with as many as 32 coordinates.

The previous maximum dimensionality was 15.

How are matrix-vector products evaluated ?

To evaluate a matrix-vector product for a single term

- Successively apply $N_r \times N_r$ matrices for each of the $\hat{t}^{(d,l)}(q_k)$ factors to a vector.
- Exploit the fact that for every factor, there is a basis set order in which the matrix representation of the factor is block diagonal.

Model problems

Start with an uncoupled Hamiltonian,

$$H_{sep}(\mathbf{x}) = \sum_{n=1}^{D} -\frac{1}{2}\frac{\partial^2}{\partial x_n^2} + \sum_{n=1}^{D}\left[c_2\left(x_n^2\right) + c_3\left(x_n^3\right) + c_4\left(x_n^4\right)\right]$$

Make a coordinate transformation.

$$\mathbf{x} = \mathbf{M}\mathbf{q}$$

Two Hamiltonians :

H_L, for which we have exact solutions, and H_S with fewer terms.

H_S is obtained by making the coordinate transformation only for the potential terms and simply replacing x_i with q_i in the KEO.

Despite the simple model these Hamiltonians are complicated.

With $D = 32$: H_L has 528 KEO terms and 608 potential terms. H_s has 32 KEO terms and 608 potential terms.

The size of the direct product basis from which we select basis functions to retain is 14^D.

We use the Lanczos algorithm to compute spectra.

Results

$N_r = 30'000$	DP	Exact
-30.92854	-30.92876	-30.92875 (1)
-28.72144	-28.72200	-28.72200 (6)
-26.81071	-26.81414	-26.81414 (6)
-26.51365	-26.51524	-26.51524 (15)
-26.08389	-26.09853	-26.09853 (6)
-25.35198	-25.36882	-25.36809 (6)

TABLE I: H_L with $D = 6$. The direct product basis has 7,529,536 functions. Figures in parentheses are the degeneracy of the exact levels.

Diagonal	$N_r = 300'000$	Exact
-163.58067	-164.95162	-164.95338 (1)
-161.38613	-162.73736	-162.74662 (32)
-159.49880	-160.79848	-160.83878 (32)
-159.19158	-160.40700	-160.53987 (496)
-158.85445	-159.97062	-160.12560 (32)
-158.08712	-159.22553	-159.39582 (32)

TABLE I: H_L with $D = 32$ The DP basis has $\sim 5 \times 10^{36}$ functions. Figures in parentheses give the degeneracy of the exact levels.

Assessment

What is good ?

- D=32 is possible.
- Unlike most basis set methods, the memory cost is not a serious problem. The 32-d problem requires only a few hundred MB.

What is not good ?

- Cannot use unless the potential is a sum of products (MCTDH has the same weakness)
- The CPU cost scales linearly with the number of terms (which increases with D)

Quadrature grids with fewer points than direct product grids

In this part of the talk I shall use the basis truncation criterion :

$$k_1 + \cdots + k_D \leq b$$

where $k_1, \cdots, k_D$ are basis function labels.

e.g., in 2-d, with b=10, I keep $k_1 = 5; k_2 = 5$ but not $k_1 = 5; k_2 = 7$

I assume that zeroth-order energies of all the coordinates are similar so that retained basis functions with the same value of $k_1 + k_2 + \cdots + k_D$ have approximately the same zeroth-order energy.

We know that basis truncation of this sort is effective.

Restricting the sum of the indices is similar to keeping basis functions with the smallest diagonal elements and for $D = 32$ this makes it possible to reduce the size of the basis by a factor of $\sim 10^{32}$

If one uses quadrature and retains basis functions from a direct product basis with n^D functions it is standard to use a direct product ($\sim N^D$) grid.

The grid is huge !

Being able to reduce the basis size is good but we are still stuck with a huge quadrature grid.

Often it is the size of the quadrature grid and not the size of the basis that prevents one from solving interesting problems.

Examples,

- To compute a spectrum of CH_5^+ we use 1341 7-d bend functions but 215×10^6 bend quadrature points.

- MCTDH is a scheme for reducing the basis size, but if the potential is not of sum of products form then the only option is to use a direct product grid

Why is a huge grid bad ?

The more grid points one has, the more CPU time is required.

The huge grid also creates a memory problem. Recall the 2-d example,

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

There are intermediate vectors as large as the grid !

Gauss grids are too good

If the direct product basis from which one selects basis functions is made from products of n Hermite, Laguerre, or Jacobi factors for each coordinate, it is common to use a grid (with which all overlap integrals are exact) with n^D Gauss points.

Even if the $\mathbf{x}$ vector has only 300'000 components, once it is transferred to the grid it has 10^{36}.

If only basis functions with $k_1 + k_2 + \cdots \leq b$ are retained, this quadrature is overkill

What Gauss does versus what we need

The product Gauss grid with $(N_1, \cdots, N_D)$ points integrates correctly the multivariate monomials

$$x_1^{l_1} ... x_D^{l_D}, \quad \text{with} \quad l_1 \leq 2N_1 - 1, ..., l_D \leq 2N_D - 1.$$

The maximum total degree of multivariate polynomials integrated correctly by the PG grid is $\sum_{i=1}^{D}(2N_i - 1)$, which is much larger than the total degree of the polynomials we must integrate to compute matrix-vector products (or potential integrals).

The highest required total degree is $2 * b + p$

p is the total degree of the potential.

TABLE I: The number of monomials $x_1^{d_1} \cdots x_D^{d_D}$ with $d_c \leq 2N-1, c = 1, \cdots, D$ of maximum degree d (necessary), the number of superfluous monomials, and the number of PG points that exactly integrate both types

	N	d	necessary	superfluous	Gaussian grid points
6D	1	1	7	57	1
	2	3	84	4012	64
	3	5	462	46194	729
	4	7	1716	260428	4096
	5	9	5005	994995	15625
	6	11	12376	2973608	46656
	7	13	27132	7502404	117649
	8	15	54264	16722952	262144
	9	17	100947	33911277	531441
	10	19	177100	63822900	1000000
12D	1	1	13	4083	1
	2	3	455	16776761	4096
	3	5	6188	2176776148	531441
	4	7	50388	68719426348	16777216
	5	9	293930	999999706070	244140625
	6	11	1352078	8916099096178	2176782336
	7	13	5200300	56693907174996	13841287201
	8	15	17383860	281474959326796	68719476736
	9	17	51895935	1156831329530241	282429536481
	10	19	141120525	4095999858879475	1000000000000

Smolyak algorithm

A grid for integrating only the necessary and not the superfluous polynomials.

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

$$S(D,K) = \sum_{D \le |i| \le K+D-1} C_{i_1,\ldots,i_D} [Q^{i_1}(x_1) \otimes \cdots \otimes Q^{i_D}(x_D)],$$

where $|i| = \sum_{\nu=1}^{D} i_\nu$

$Q^{i_q}(x_q)$ $(i_q = 1, ..., K)$ is a sequence of quadrature rules for coordinate x_q.

For example $Q^1(x)$, $Q^2(x)$, and $Q^3(x)$ might be (1d) quadrature rules with 3,5, and 7 points.

$Q_i(x)$ is an operator which when applied to a univariate function $g(x)$ gives

$$Q^j(x)g(x) = \sum_{\nu=1}^{N_j} w_\nu^j g(x_\nu^j),$$

$S(D, K)$ is an operator which when applied to a multivariate function gives a quadrature approximation for the integral of the function multiplied by an appropriate weight function.

When is the quadrature numerically exact ?

It has been proven that if the 1D quadratures $Q^j(x)$ are chosen so that 1D integrands whose degree is at most $d_j = 2j - 1$ are integrated exactly, then the Smolyak quadrature is exact for all integrals

$$\int_{a_1}^{b_1} dx_1 \int_{a_2}^{b_2} dx_2 \cdots \int_{a_D}^{b_D} dx_D \; x_1^{l_1} \ldots x_D^{l_D} \quad w_1(x_1) \cdots w_D(x_D)$$

with $l_1 + \ldots + l_D \leq 2K - 1$.

Nested grids

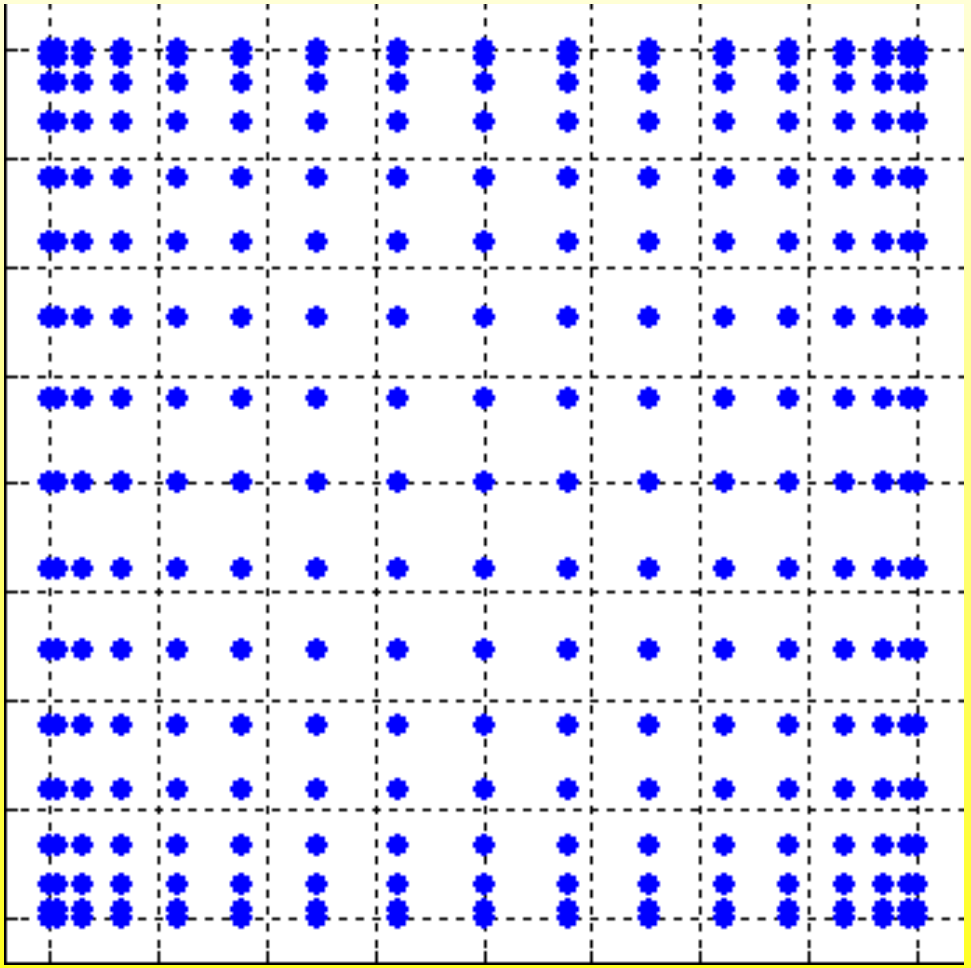
Smolyak is most advantageous if one uses nested grids.

In this case some multi-d points occur in more than one term.

This reduces the number of effective points.

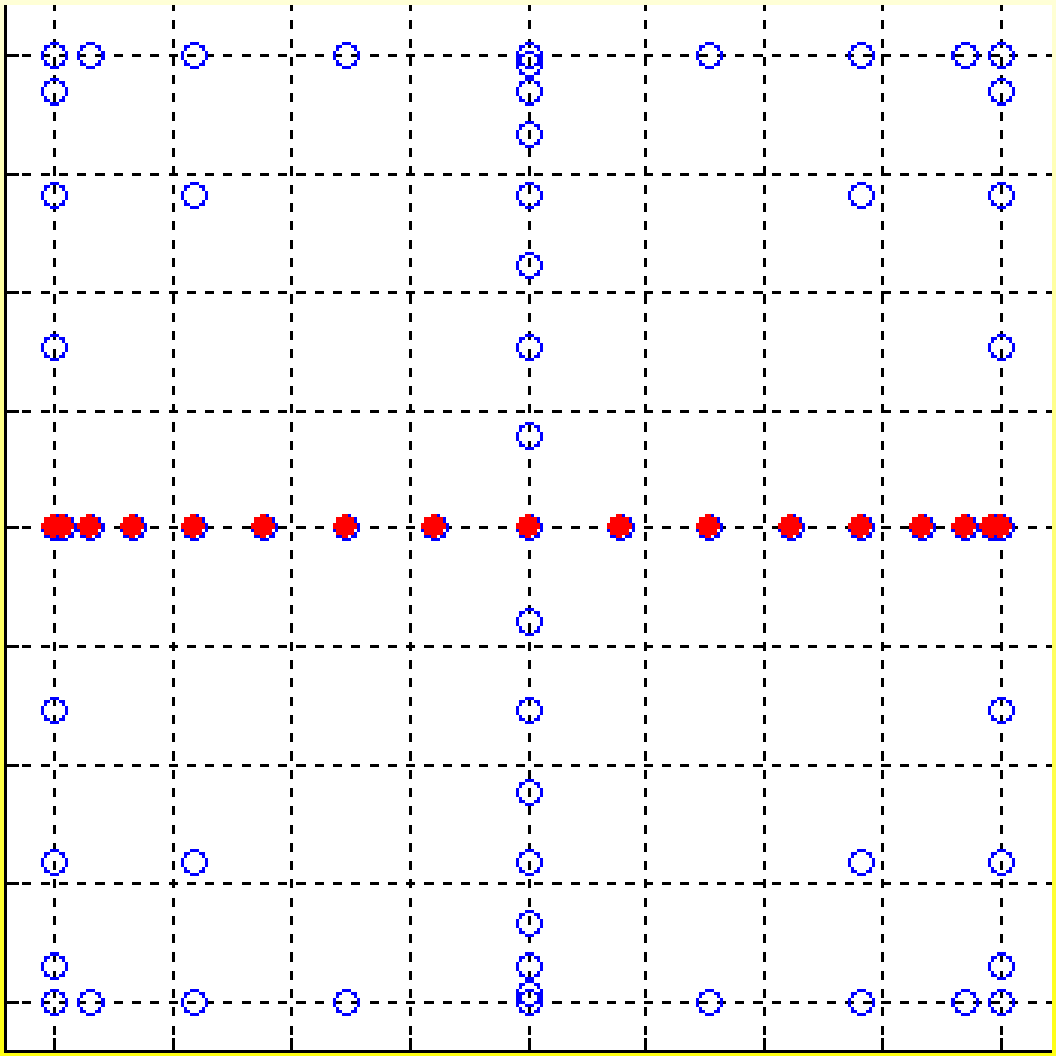
Many applications of the Smolyak idea use Clenshaw-Curtis points.

Smolyak Quadrature: 2D Order 17 Product Rule

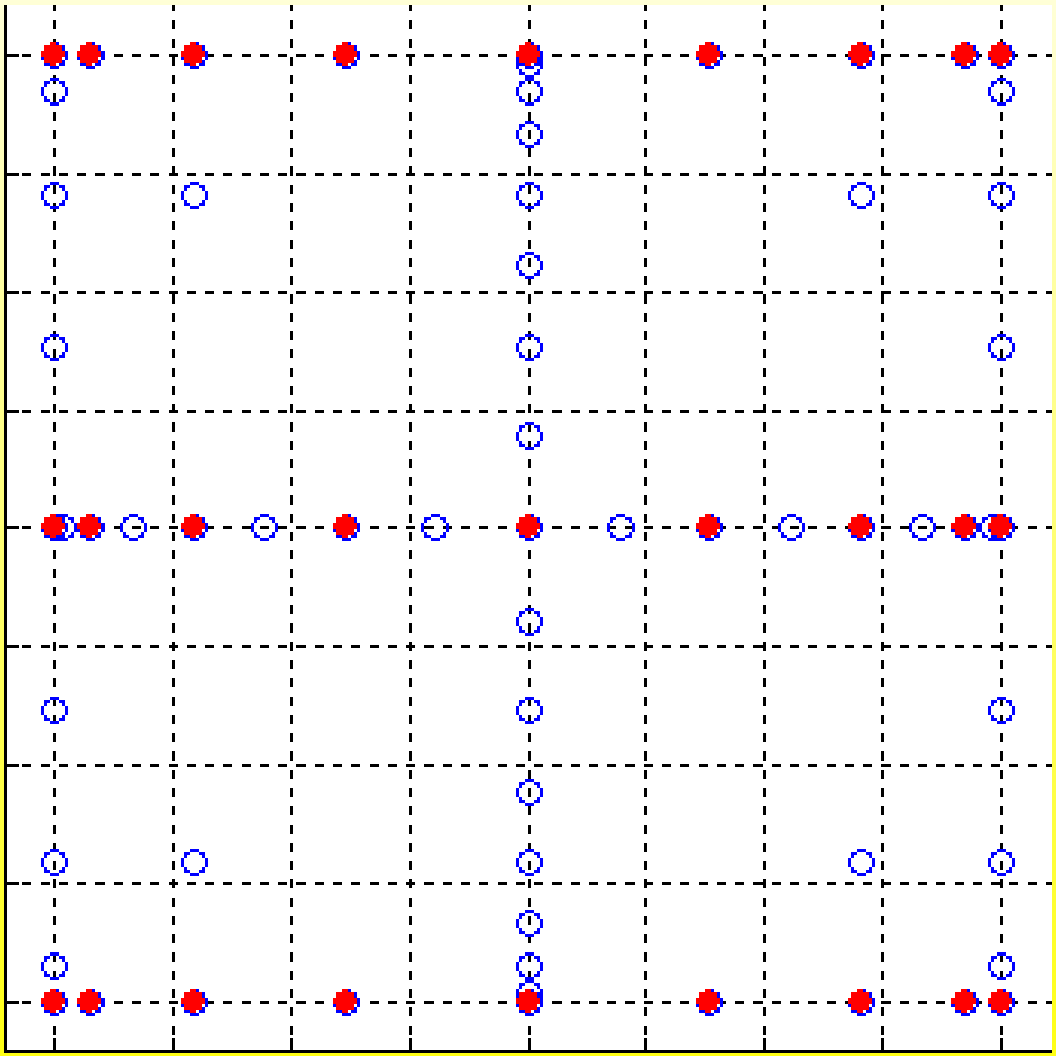


A 17x17 product grid (289 points).

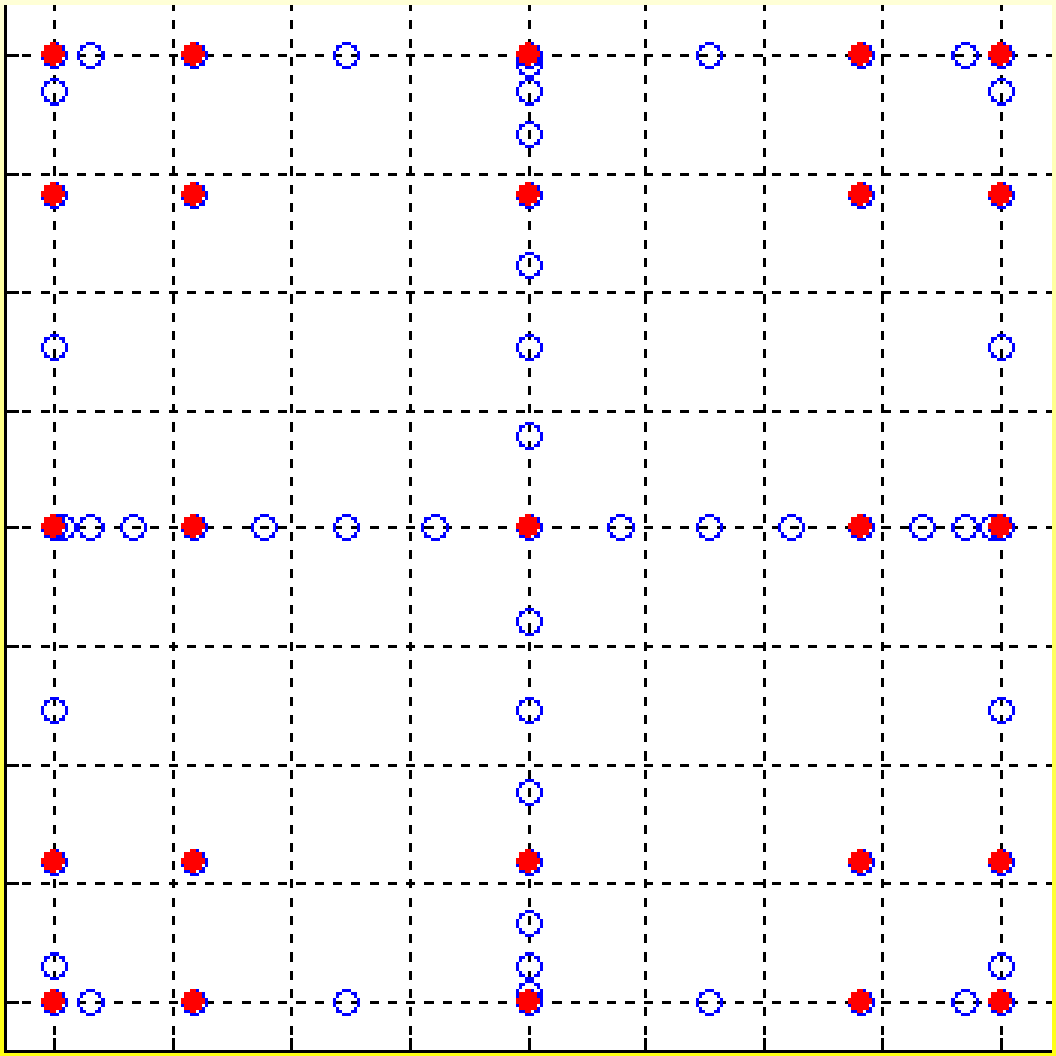
Smolyak Quadrature: 2D Level4 17x1 component



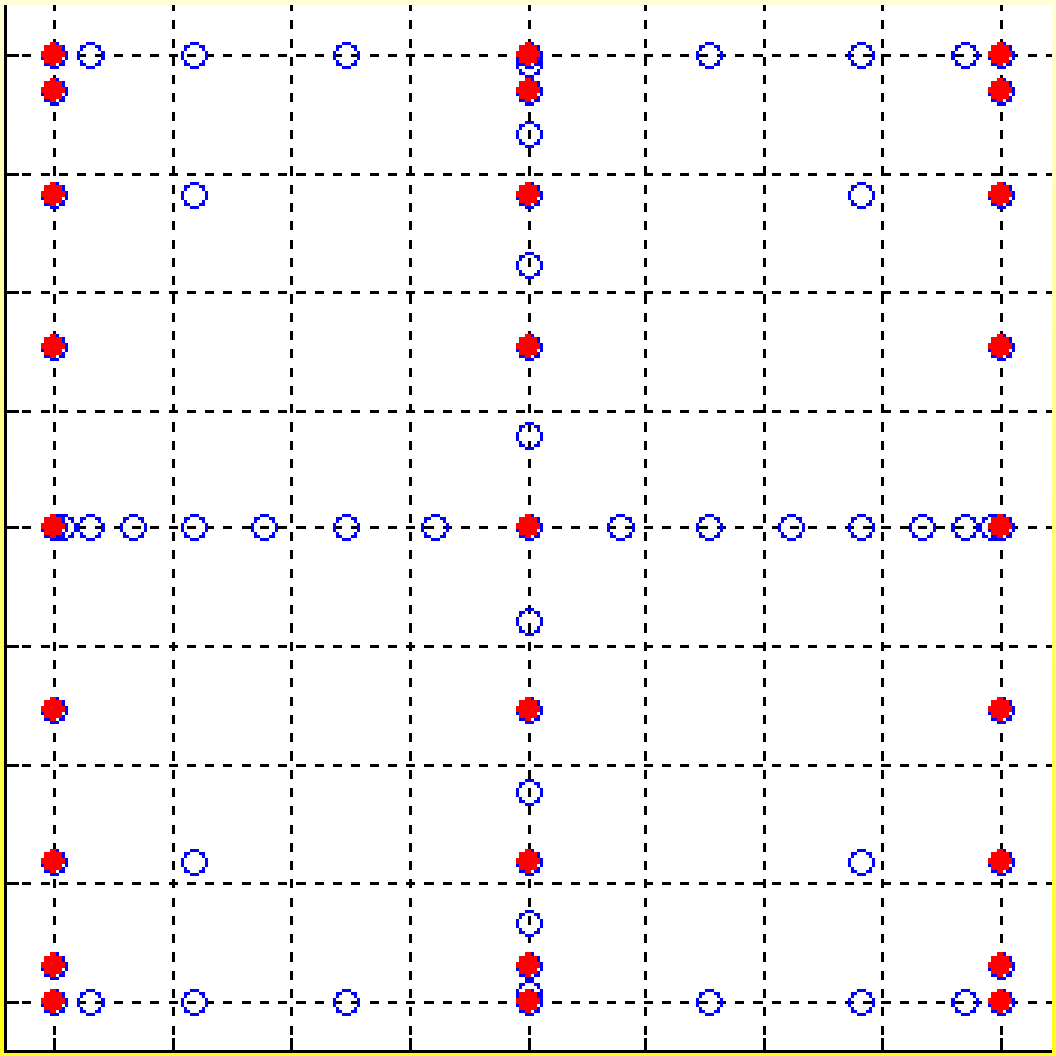
Smolyak Quadrature: 2D Level4 9x3 component



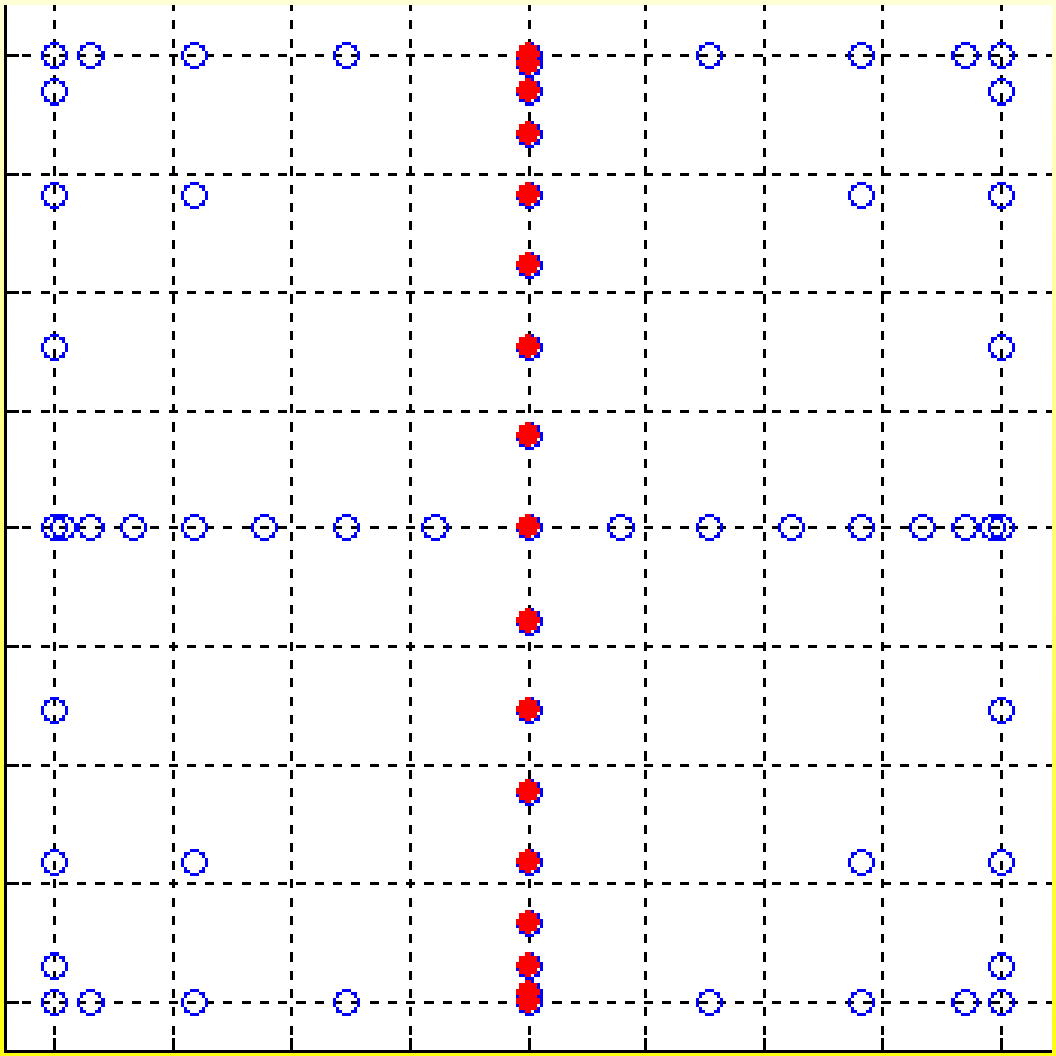
Smolyak Quadrature: 2D Level4 5x5 component



Smolyak Quadrature: 2D Level4 3x9 component



Smolyak Quadrature: 2D Level4 1x17 component



To build a Smolyak quadrature one must choose the 1d quadratures $Q^1, Q^2, \cdots$.

Clenshaw Curtis quadratures are poor for the purpose of solving the vibrational Schroedinger equation because they are designed for a constant weight function.

In 1-d Gauss quadrature is great, but used in conjunction with Smolyak, Gauss points are not the best because they are not nested.

We have developed two options for choosing the 1d quadratures. Both work with weight functions used when do vibrational calculations.

Smolyak adapted quadratures

$$\text{Smolyak} - \text{adapted} - \text{quadratures}$$

$$N_i = 1, 3, 5, 7, 9, 11, 13, 15, ...,$$

$$d_i = 1, 3, 5, 7, 9, 11, 13, 15, ...$$

TABLE I: Number of Smolyak points for $D = 12$

K	d	CC	SAQ	GPQ
2	3	25	25	4'096
4	7	2'649	2'625	2×10^7
6	11	93'489	85'305	2×10^9
8	15	2×10^6	1×10^6	7×10^{10}
10	19	3×10^7	1×10^7	1×10^{12}
12	23	3×10^8	1×10^8	9×10^{12}
14	27	3×10^9	6×10^8	6×10^{13}
16	31		3×10^9	3×10^{14}

How are the SAQ points determined ?

- Algorithm by Patterson : Start with n points, add m new points and find $n + m$ weights to determine a quadrature rule of degree $n + 2m - 1$
- Points are often complex or outside the required interval
  - Begumisa suggested reducing the maximum degree.
  - This can be done by using one arbitrary point.
  - Enables us to obtain nested quadratures.
- Weights are determined by solving a set of over-determined linear equations to minimize the error of integrals of monomials.

Test on SF_6

Local mode model – only stretch coordinates.

TABLE I: Vibrational energies of SF_6.

ν	Symm	SAQ/cm^{-1}	Numer exact/cm^{-1}	Δ/cm^{-1}
ν_2	E_g	643.64	643.64	0.00
ν_1	A_{1g}	770.86	770.85	0.00
ν_3	F_{1u}	947.22	947.22	0.00
$2\nu_2$	A_{1g}	1286.35	1286.35	0.00
$2\nu_2$	E_g	1287.00	1287.00	0.00
$\nu_1 + \nu_2$	E_g	1413.50	1413.50	0.00
$2\nu_1$	A_{1g}	1541.47	1541.47	0.00
$\nu_2 + \nu_3$	F_{1u}	1587.56	1587.56	0.00
$\nu_2 + \nu_3$	F_{2u}	1591.58	1591.58	0.00

Far fewer SAQ points are needed than would be required with Gauss

- The product Gauss grid required to evaluate exactly all of the multivariate polynomials $z_1^{l_1} z_2^{l_2} z_3^{l_3} z_4^{l_4} z_5^{l_5} z_6^{l_6}$ with $l_1 + l_2 + l_3 + l_4 + l_5 + l_6 \leq 27$ that appear in the integral has 7'529'536 points

- The SAQ Smolyak grid with $K = 14$ that does the same integrals exactly has only 579'125 points.

Matrix-vector products

Of critical importance is the fact that that potential matrix-vector products can be computed at a cost that scales as N^{D+1}, **despite the fact that the quadrature grid is not a direct product grid**.

The first step requires writing a Smolyak quadrature as a constrained sum over points.

This is a bit tricky because the Smolyak equation is a constrained sum over quadratures (and not points).

Pseudo Gauss Quadratures

The 1-d quadratures have $1, 2, 3, \cdots$ (nested) points and minimize the error of integrals of monomials with degree $1, 3, 5, \cdots$ (but the integrals are not exact).

How are the PGQ points defined ?

Points and weights are chosen so that 1-d overlap integrals are as exact as possible.

Q^1 is a one-point Gauss quadrature.

To make Q^N from Q^{N-1} we add a point, x_N, and determine (N) weights by solving $PW = L$, where W is a vector of weights, $W = (w_1, w_2, ...w_N)^T$,

$$P_{n,m;i} = f_n(x_i) f_m(x_i),$$

$$L_{n,m} = \int w(x) f_n(x) f_m(x) dx$$

and n, m are all n, m pairs with $n + m \leq (2N - 1)$

We then vary x_N to minimize

$$g(x_N) = \sum_{m=0}^{d} \sum_{n=0}^{d} \mid \delta_{m,n} - \sum_{i=1}^{N} w_i h_n^{-1/2} h_m^{-1/2} f_n(x_i) f_m(x_i) \mid, \quad n+m \leq d,$$

TABLE I: Vibrational energies of SF_6

ν	Symm	PGQ /cm^{-1}	Numer exact/cm^{-1}	Δ/cm^{-1}
ν_2	E_g	643.64	643.64	0.00
ν_1	A_{1g}	770.86	770.86	0.00
ν_3	F_{1u}	947.22	947.23	−0.01
$2\nu_2$	A_{1g}	1286.35	1286.35	0.00
$2\nu_2$	E_g	1287.00	1287.00	0.00
$\nu_1 + \nu_2$	E_g	1413.50	1413.52	−0.02
$2\nu_1$	A_{1g}	1541.47	1541.44	0.03
$\nu_2 + \nu_3$	F_{1u}	1587.56	1587.56	0.00
$\nu_2 + \nu_3$	F_{2u}	1591.58	1591.59	−0.01
$\nu_1 + \nu_3$	F_{1u}	1717.00	1717.05	−0.04
$2\nu_3$	E_g	1889.54	1889.59	−0.05
$2\nu_3$	A_{1g}	1889.58	1889.69	−0.10

Pseudo Gauss Quadrature grid is more than two orders of magnitude smaller than product Gauss

The quadrature is designed to accurately compute integrals with integrands of maximum total degree $l_1 + l_2 + l_3 + l_4 + l_5 + l_6 \leq 25$.

- The PGQ Smolyak grid has 18'564 points.
- The corresponding product Gauss quadrature that does the overlaps exactly has $13^6 = 4'826'809$

Conclusion

- Using a pruned basis set in conjunction with an iterative eigensolver it is possible to solve vibrational problems with 32 coordinates. This works if the potential is a sum of products because in this case no (multi-d) quadrature is necessary

- For a general potential, quadrature *is* necessary. In this case, a combination of ideas of Smolyak and our new sets of 1-d quadratures makes it possible to drastically reduce the number of required quadrature points.

- In 12D, the SAQ grid required to integrate polynomials of total degree 31 is a factor of 106069 smaller than the standard Gauss grid. With PGQ in 12D the reduction factor is $\sim 2 \times 10^7$!

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.