POTFIT and Multigrid POTFIT Transforming general multi-dimensional potential energy surfaces to product form Applications to $H_3O_2^-$

Hans-Dieter Meyer

Theoretische Chemie
Universität Heidelberg

Banff, May 2nd, 2013

Outline

Content

Multiconfiguration time-dependent Hartree, MCTDH

The *ansatz* for the MCTDH wavefunction reads

$$\begin{aligned}\Psi(q_1, \cdots, q_f, t) &= \sum_{j_1=1}^{n_1} \cdots \sum_{j_f=1}^{n_f} A_{j_1,\cdots,j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_\kappa}^{(\kappa)}(q_\kappa, t) \\ &= \sum_{J} A_J \, \Phi_J\end{aligned}$$

Multiconfiguration time-dependent Hartree, MCTDH

The *ansatz* for the MCTDH wavefunction reads

$$\begin{aligned}\Psi(q_1,\cdots,q_f,t) &= \sum_{j_1=1}^{n_1}\cdots\sum_{j_f=1}^{n_f} A_{j_1,\cdots,j_f}(t)\prod_{\kappa=1}^{f}\varphi_{j_\kappa}^{(\kappa)}(q_\kappa,t)\\ &= \sum_{J} A_J\,\Phi_J\end{aligned}$$

Single-particle functions:

$$\varphi_{j_\kappa}^{(\kappa)}(q_\kappa,t)=\sum_{l=1}^{N_\kappa} c_{j_\kappa l}^{(\kappa)}(t)\,\chi_l^{(\kappa)}(q_\kappa)$$

MCTDH equations of motion

$$
\begin{aligned}
i\dot{A}_J &= \sum_L \langle \Phi_J | \hat{H} | \Phi_L \rangle A_L \\
i\dot{\varphi}_j^{(\kappa)} &= \left(1 - P^{(\kappa)}\right) \sum_{k,l} \rho_{j,k}^{(\kappa)^{-1}} \langle \hat{\mathbf{H}} \rangle_{k,l}^{(\kappa)} \varphi_l^{(\kappa)}
\end{aligned}
$$

The computation of the Hamiltonian matrix $\langle \Phi_J \mid \hat{H} \mid \Phi_L \rangle$ and the mean-fields $\langle \hat{\mathbf{H}} \rangle_{k,l}^{(\kappa)}$ requires the evaluation of multi-dimensional integrals. It is essential that these integrals are done fast.

Product representation of the Hamiltonian

We require the Hamiltonian to be in product form

$$\hat{H} = \sum_{r=1}^{s} c_r \prod_{\kappa=1}^{f} \hat{h}_r^{(\kappa)}$$

where $\hat{h}_r^{(\kappa)}$ operates on the κ-th degree of freedom only.

Product representation of the Hamiltonian

We require the Hamiltonian to be in product form

$$\hat{H} = \sum_{r=1}^{s} c_r \prod_{\kappa=1}^{f} \hat{h}_r^{(\kappa)}$$

where $\hat{h}_r^{(\kappa)}$ operates on the κ-th degree of freedom only.

The multi-dimensional integrals can then be written as a sum of products of one- or low-dimensional integrals

$$\langle \Phi_J \mid \hat{H} \mid \Phi_L \rangle = \sum_{r=1}^{s} c_r \langle \varphi_{j_1}^{(1)} \mid \hat{h}_r^{(1)} \mid \varphi_{l_1}^{(1)} \rangle \ldots \langle \varphi_{j_f}^{(f)} \mid \hat{h}_r^{(f)} \mid \varphi_{l_f}^{(f)} \rangle$$

Product representation of the Hamiltonian

We require the Hamiltonian to be in product form

$$\hat{H} = \sum_{r=1}^{s} c_r \prod_{\kappa=1}^{f} \hat{h}_r^{(\kappa)}$$

where $\hat{h}_r^{(\kappa)}$ operates on the κ-th degree of freedom only.

The multi-dimensional integrals can then be written as a sum of products of one- or low-dimensional integrals

$$\langle \Phi_J \mid \hat{H} \mid \Phi_L \rangle = \sum_{r=1}^{s} c_r \langle \varphi_{j_1}^{(1)} \mid \hat{h}_r^{(1)} \mid \varphi_{l_1}^{(1)} \rangle \ldots \langle \varphi_{j_f}^{(f)} \mid \hat{h}_r^{(f)} \mid \varphi_{l_f}^{(f)} \rangle$$

An alternative fast algorithm is the CDVR method of U. Manthe.

See also Ávila and Carrington JCP **134** (2011) 054126. (Smolyak)

Content

Potfit

The most direct way to the product form is an expansion in a product basis. Hence we approximate some given potential V by

$$V^{\mathrm{PF}}\left(q^{(1)},\ldots,q^{(f)}\right)=\sum_{j_1=1}^{m_1}\cdots\sum_{j_f=1}^{m_f}C_{j_1\ldots j_f}\,v_{j_1}^{(1)}(q^{(1)})\ldots v_{j_f}^{(f)}(q^{(f)})$$

Potfit

The most direct way to the product form is an expansion in a product basis. Hence we approximate some given potential V by

$$V^{\mathrm{PF}}\left(q^{(1)},\ldots,q^{(f)}\right)=\sum_{j_1=1}^{m_1}\cdots\sum_{j_f=1}^{m_f}C_{j_1\ldots j_f}\,v_{j_1}^{(1)}(q^{(1)})\ldots v_{j_f}^{(f)}(q^{(f)})$$

working with grids we set:

$$V(q_{i_1}^{(1)},\ldots,q_{i_f}^{(f)})=V_{i_1\ldots i_f}\quad\text{and}\quad v_{ij}^{(\kappa)}=v_j^{(\kappa)}(q_i^{(\kappa)})$$

This yields:

$$V_{i_1\ldots i_f}^{\mathrm{PF}}=\sum_{j_1=1}^{m_1}\cdots\sum_{j_f=1}^{m_f}C_{j_1\ldots j_f}\,v_{i_1j_1}^{(1)}\ldots v_{i_fj_f}^{(f)}$$

Tucker decomposition $\quad i_\kappa=1\cdots N_\kappa\quad j_\kappa=1\cdots m_\kappa$

Potfit

The coefficients are given by overlap

$$C_{j_1 \ldots j_f} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_p=1}^{N_f} v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f} \, V_{i_1 \ldots i_f}$$

Potfit

The coefficients are given by overlap

$$C_{j_1 \ldots j_f} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_p=1}^{N_f} v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f} \, V_{i_1 \ldots i_f}$$

More difficult is to find optimal single-particle potentials (SPPs). We define the SPPs as eigenvectors of the potential density matrices

$$\varrho^{(\kappa)}_{k k'} = \sum_{I^\kappa} V_{i_1 \ldots i_{\kappa-1} k i_{\kappa+1} \ldots i_f} \, V_{i_1 \ldots i_{\kappa-1} k' i_{\kappa+1} \ldots i_f}$$

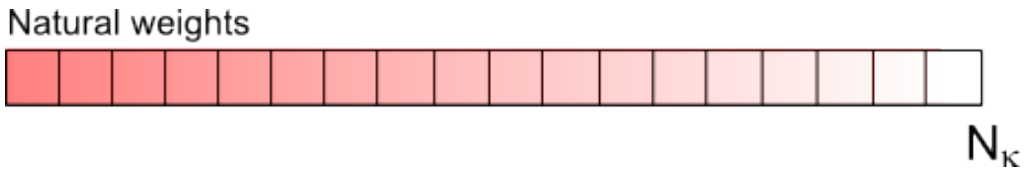
Eigenvalues: $\lambda^{(\kappa)}_{j_\kappa}$ Eigenvectors: $\mathbf{v}^{(\kappa)}_{j_\kappa}$

The approximated potential

The *natural potentials* will be the taken as SPPs for the approximated potential expansion:

$$V^{\mathrm{PF}}_{i_1 \ldots i_f} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \ldots j_f} \, v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f}$$

The *natural weights* $(\lambda^{(\kappa)}_{j_\kappa})$ provide us an estimation of the number of expansion functions or *expansion orders* $(m_\kappa \leq N_\kappa)$ that we need to describe our potential up to a certain accuracy:

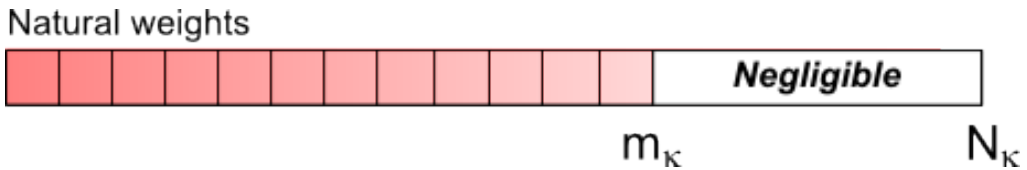


The approximated potential

The *natural potentials* will be the taken as SPPs for the approximated potential expansion:

$$V^{\mathrm{PF}}_{i_1 \ldots i_f} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \ldots j_f} \, v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f}$$

The *natural weights* $(\lambda^{(\kappa)}_{j_\kappa})$ provide us an estimation of the number of expansion functions or *expansion orders* $(m_\kappa \leq N_\kappa)$ that we need to describe our potential up to a certain accuracy:

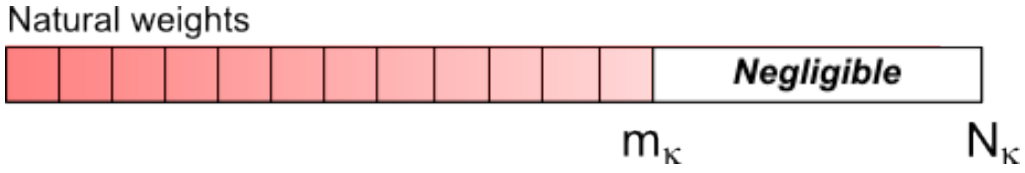


The approximated potential

The *natural potentials* will be the taken as SPPs for the approximated potential expansion:

$$V^{\mathrm{PF}}_{i_1 \ldots i_f} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \ldots j_f} \, v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f}$$

The *natural weights* $(\lambda^{(\kappa)}_{j_\kappa})$ provide us an estimation of the number of expansion functions or *expansion orders* $(m_\kappa \leq N_\kappa)$ that we need to describe our potential up to a certain accuracy:



POTFIT (1996), HOSVD

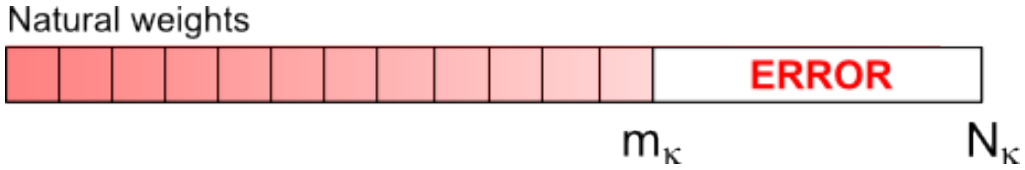
Error estimation

The L^2-error is defined as: $\Delta^2 = \sum_I \left(V_I - V_I^{PF}\right)^2$.

The error is bound by:

$$\frac{1}{f-1} \sum_{\substack{\kappa=1 \\ \kappa \neq \nu}}^{f} \sum_{j=m_\kappa+1}^{N_\kappa} \lambda_j^{(\kappa)} \leq \Delta_{opt}^2 \leq \Delta^2 \leq \sum_{\substack{\kappa=1 \\ \kappa \neq \nu}}^{f} \sum_{j=m_\kappa+1}^{N_\kappa} \lambda_j^{(\kappa)}$$

The error is determined by the eigenvalues of the *neglected* SPPs.
For $m_\kappa = N_\kappa$ one recovers the exact potential on the grid.



POTFIT problems

The applicability of POTFIT is limited by the computation of the density matrix and the overlaps, which run over the **complete** grid:

$$\rho^{(\kappa)}_{kk'} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_{\kappa-1}=1}^{N_{\kappa-1}} \sum_{i_{\kappa+1}=1}^{N_{\kappa+1}} \cdots \sum_{i_f=1}^{N_f} V_{i_1 \ldots i_{\kappa-1} k i_{\kappa+1} i_f} \; V_{i_1 \ldots i_{\kappa-1} k' i_{\kappa+1} i_f}$$

$$C_J = C_{j_1 \ldots j_f} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_f=1}^{N_f} v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f} \; V_{i_1 \ldots i_f}$$

We cannot deal with problems with more than
10^9 grid points (6-8 DOFs).

POTFIT problems

The applicability of POTFIT is limited by the computation of the density matrix and the overlaps, which run over the **complete** grid:

$$\rho^{(\kappa)}_{kk'} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_{\kappa-1}=1}^{N_{\kappa-1}} \sum_{i_{\kappa+1}=1}^{N_{\kappa+1}} \cdots \sum_{i_f=1}^{N_f} V_{i_1 \ldots i_{\kappa-1} k i_{\kappa+1} i_f} \; V_{i_1 \ldots i_{\kappa-1} k' i_{\kappa+1} i_f}$$

$$C_J = C_{j_1 \ldots j_f} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_f=1}^{N_f} v^{(1)}_{i_1 j_1} \cdots v^{(f)}_{i_f j_f} \; V_{i_1 \ldots i_f}$$

We cannot deal with problems with more than 10^9 grid points (6-8 DOFs).

Multigrid POTFIT (MGPF) has been conceived to (*partially*) overcome these difficulties.

Content

From coarse to fine grid

Let us consider a system which **requires** to be described by an **exceedingly large** primitive grid (hereafter **fine grid**, $\tilde{I}$) such that it cannot be *potfitted*. Let us define a coarse grid (I), which is subset of the fine grid ($I \subset \tilde{I}$) and which is small enough (with n_κ rather than N_κ grid points) to be potfittable. We first potfit the PES on the coarse grid and then, DOF after DOF we replace the coarse grid SPPs ($\mathbf{v}$) with fine grid SPPs ($\tilde{\mathbf{v}}$).

Approximated potential on the fine grid

$$V_I^{\mathrm{PF}} = \sum_J C_J \prod_{\kappa=1}^{f} v_{i_\kappa j_\kappa}^{(\kappa)}$$

$$M_{I^\kappa, j_\kappa}^{(\kappa)} = \sum_{J^\kappa} C_J \prod_{\kappa' \neq \kappa} v_{i_{\kappa'} j_{\kappa'}}^{(\kappa')}$$

$$min \left\{ \sum_{I^\kappa} \sum_{\tilde{i}_\kappa} \left(V_{I^\kappa_{\tilde{i}_\kappa}} - \sum_{j_\kappa} M_{I^\kappa, j_\kappa}^{(\kappa)} \; \tilde{v}_{\tilde{i}_\kappa, j_\kappa}^{(\kappa)} \right)^2 \right\}$$

$$V_{\tilde{I}}^{\mathrm{MGPF}} = \sum_J C_J \prod_{\kappa=1}^{f} \tilde{v}_{\tilde{i}_\kappa, j_\kappa}^{(\kappa)}$$

MGPF working equations

A **full-representation** on the coarse grid ($m = n$) implies that $V_I^{\mathrm{PF}} \equiv V_I$ and then the equations for the fine grid SPPs read

$$\tilde{\mathbf{v}}^{(\kappa)} = \boldsymbol{\rho}^{(\kappa)'} \boldsymbol{\rho}^{(\kappa)^{-1}} \mathbf{v}^{(\kappa)}$$

- MGPF is a *function-driven* interpolation $\boldsymbol{\rho}^{(\kappa)'} \boldsymbol{\rho}^{(\kappa)^{-1}}$,
- MGPF potential is EXACT on ALL coarse grid points.
- MGPF SPPs are NOT natural potentials, but one can transform them to orthonormal, importance ordered natural potentials.

MGPF Computational effort

	Total effort	# Potential evaluations
POTFIT	$f \cdot N^{f+1}$	N^f
MGPF	$2 \cdot f \cdot N \cdot n^f$	$f \cdot N \cdot n^{f-1}$

Note that MGPF is linear in N.

$$\text{gain}_{effort} = \frac{1}{2}\left(\frac{N}{n}\right)^f$$

$$\text{gain}_{eval} = \frac{1}{f}\left(\frac{N}{n}\right)^{f-1}$$

MGPF Computational effort

	Total effort	# Potential evaluations
POTFIT	$f \cdot N^{f+1}$	N^f
MGPF	$2 \cdot f \cdot N \cdot n^f$	$f \cdot N \cdot n^{f-1}$

Note that MGPF is linear in N.

$$\text{gain}_{effort} = \frac{1}{2}\left(\frac{N}{n}\right)^f \qquad \text{gain}_{eval} = \frac{1}{f}\left(\frac{N}{n}\right)^{f-1}$$

$f = 12,\ N = 25,\ n = 4$ or 3

	POTFIT	MGPF(4)	MGPF(3)
Evaluations	$6.0 \cdot 10^{16}$	$1.3 \cdot 10^9$	$5.3 \cdot 10^7$
Operations	$1.8 \cdot 10^{19}$	$1.0 \cdot 10^{10}$	$3.2 \cdot 10^8$

The two flavours of MGPF

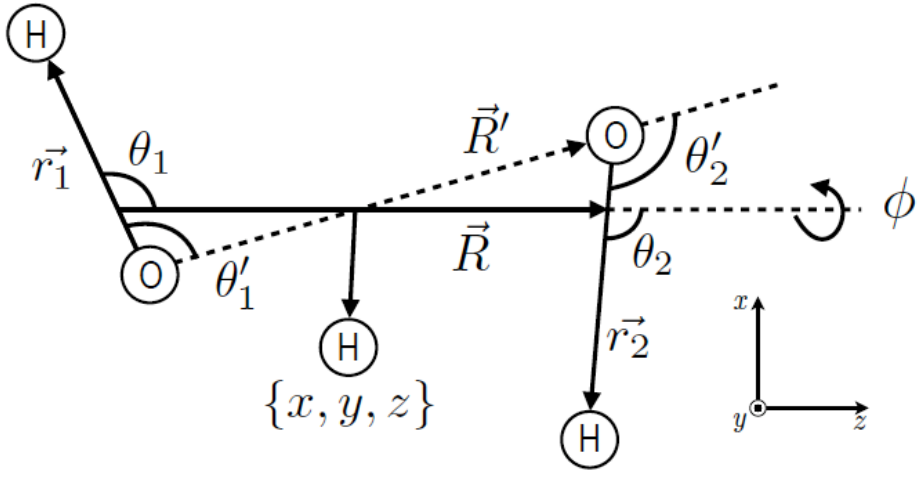
Two approaches to MGPF

- Bottom-up, make a nice guess of a *small* coarse grid:
  - ...tedious, choose a coarse grid, optimize it, (start over and check),
  - ...usually, this will not be as accurate (depends on our needs),
  - ...but, nevertheless, is fast!

- Top-down, use a *very* large coarse grid and trim the resulting expansion:
  - ...more expensive,
  - ...but more accurate!

Content

Choice of the coordinates



Combination scheme: $[r_1, r_2]$, $[x, y, \phi]$, $[u_1, u_2]$, $[R, z_{\text{red}}]$
Number of SPFs: [11/55/25/18].

rms-errors

Rms-errors in cm^{-1} for a series of Jacobi td-MGPF expansions.

				$k_BT(cm^{-1})$		
λ_{thrs}^{red}	**SPP**	**Terms**	$\Lambda_{red}^{1/2}$	**400**	**4000**	**10000**
8520	[5/8/8/c]	320	234.84	59.587	102.925	123.416
852	[10/14/10/c]	1400	85.81	30.403	42.088	50.354
85.2	[13/24/16/c]	4992	29.96	9.576	14.220	17.504
8.52	[18/37/21/c]	13986	11.11	2.889	5.371	6.546
0.85	[25/58/27/c]	40716	4.63	1.287	2.102	2.617

Rms-errors in cm^{-1} for a series of Valence td-MGPF expansions.

				$k_BT(cm^{-1})$		
λ_{thrs}^{red}	**SPP**	**Terms**	$\Lambda_{red}^{1/2}$	**400**	**4000**	**10000**
12150	[5/7/7/c]	245	260.50	58.675	106.644	128.209
1215	[8/12/9/c]	864	120.59	29.437	53.548	64.068
121.5	[12/21/15/c]	3780	39.35	9.151	17.506	21.171
12.15	[18/35/20/c]	12600	14.12	3.128	6.615	7.947
1.215	[25/57/25/c]	35625	5.47	1.429	2.750	3.345

Total fine-grid size: 1.79×10^{10} (Jac.) or 1.12×10^{10} (Val.)

Ground State Energy (ZPE)

	Ground state energies (cm^{-1})				
	Maximum natural population				
Coord.	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Jacobi	6594.53	6604.07	6600.05	6601.60	6602.48
Valence	6597.46	6600.95	6600.33	6602.14	6602.50
Terms J	320	1,400	4,992	13,986	40,716
Terms V	245	864	3,780	12,600	35,625

Excitation Energies, Fundamentals

	MCTDH		**DMC**		**Lanczos**		**VSCF/CI**	
Mode	**even**	**odd**	**even**	**odd**	**even**	**odd**	**even**	**odd**
GS	0.00	18.13	0	14	0.0	13.3	0	22
ϕ	131.71	217.88	131	224	132.5	214.9	132	215
$u_1 + u_2$	439.90	480.45	479	517	460.6	490.2	465	528
R	485.07	503.34	505	521	499.2	519.0	515	540
$u_1 - u_2$	573.07	583.82	588	602	598.9	603.9	576	606
z	691.69	721.40	644	665	759.9	758.8	741	785

DMC: McCoy, JCP **123** 064317 (2005)
Lanczos: Yu, JCP **125** 204306 (2006)
VSCF/CI: Bowman, Carter, JCP **123** 064317 (2005)

Content

Summary and Outlook

- MGPF can bring a high dimensional PES (9D and hopefully 12D) into product form (here: Tucker format).
- It does this efficiently and accurately.
- However, the expansion may consist of (too) many terms.
- There are several possible strategies to reduce the number of terms:
  - Introduce weights in MGPF.
  - Transform the MGPF potential tensor to a CANDECOMP format: $V^{\text{app}}_{i_1,\ldots,i_f} = \sum_{r=1}^{R} C_r \, v^{(1)}_{i_1,r} \ldots v^{(f)}_{i_f,r}$
  - Transform the MGPF potential tensor to multilayer form.

Acknowledgments

- Daniel Pelaez
- Keyvan Sadri
- Joel Bowman, for sending us the PES routine.

Thank you!

http://mctdh.uni-hd.de/