The Multiconfiguration Time-Dependent Hartree (MCTDH) Method and its Multi-Layer (ML-MCTDH) Extension

Hans-Dieter Meyer

Theoretische Chemie
Universität Heidelberg

Quantum Days, Bilbao, July 13/14, 2015
1 Multiconfiguration time-dependent Hartree, MCTDH
2 Multi-Layer MCTDH
3 MCTDH and ML-MCTDH viewed as tensor decomposition methods
4 Compact representations of the PES
5 Highlights and Conclusions
1. Multiconfiguration time-dependent Hartree, MCTDH

2. Multi-Layer MCTDH

3. MCTDH and ML-MCTDH viewed as tensor decomposition methods

4. Compact representations of the PES

5. Highlights and Conclusions
The *ansatz* for the MCTDH wavefunction reads

\[ \Psi(q_1, \cdots, q_f, t) = \sum_{j_1=1}^{n_1} \cdots \sum_{j_f=1}^{n_f} A_{j_1, \cdots, j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_{\kappa}}^{(\kappa)}(q_{\kappa}, t) \]

\[ = \sum_{J} A_{J} \Phi_{J} \]
The *ansatz* for the MCTDH wavefunction reads

\[
\psi(q_1, \cdots, q_f, t) = \sum_{j_1=1}^{n_1} \cdots \sum_{j_f=1}^{n_f} A_{j_1, \cdots, j_f}(t) \prod_{\kappa=1}^{f} \varphi_{j_{\kappa}}^{(\kappa)}(q_{\kappa}, t)
\]

\[
= \sum_J A_J \Phi_J
\]

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:

\[
i \dot{A}_J = \sum_L \langle \Phi_J | \hat{H} | \Phi_L \rangle A_L
\]

\[
i \dot{\varphi}^{(\kappa)}_j = (1 - P^{(\kappa)}) \sum_{k,l} \rho^{(\kappa)}_{j,k} \langle \hat{H}^{(\kappa)} \rangle_{k,l} \varphi^{(\kappa)}_l
\]
Time-independent Schrödinger equation

Applying a variational principle leads to an eigenvalue problem for the coefficients

\[ \sum_{L} \langle \Phi_{J} | \hat{H} | \Phi_{L} \rangle A_{L} = E A_{J} \]

and a relaxation procedure for the single-particle functions

\[ \frac{\partial}{\partial \tau} \varphi_{j}^{(\kappa)} := - \left( 1 - P^{(\kappa)} \right) \sum_{k,l=1}^{n_{\kappa}} \left( \rho^{(\kappa)} \right)_{jk}^{-1} \langle \hat{H} \rangle_{kl}^{(\kappa)} \varphi_{l}^{(\kappa)} \to 0. \]

- Equations must be fulfilled simultaneously
- Start with a guess wavefunction
- Solve iteratively until self-consistency ("Improved relaxation")
MCTDH with Mode Combination

\[
\left( q_1, q_2, q_3, q_4, q_5, q_6, \ldots, q_{f-1}, q_f \right)
\]

\[
Q_1 \quad Q_2 \quad Q_3 \quad \ldots \quad Q_p
\]
MCTDH with Mode Combination

\[
\left( q_1, q_2, q_3, q_4, q_5, q_6, \ldots, q_{f-1}, q_f \right)
\]

\[
Q_1 \quad Q_2 \quad Q_3 \quad \ldots \quad Q_p
\]

MCTDH wavefunction

\[
\Psi(q_1, \ldots, q_f, t) \equiv \Psi(Q_1, \ldots, Q_p, t)
\]

\[
= \sum_{j_1}^{n_1} \cdots \sum_{j_p}^{n_p} A_{j_1, \ldots, j_p}(t) \prod_{\kappa=1}^{p} \varphi_{j_{\kappa}}^{(\kappa)}(Q_{\kappa}, t)
\]

Single-particle functions:

\[
\varphi_{j_{\kappa}}^{(\kappa)}(Q_{\kappa}, t) = \sum_{l_1=1}^{N_{1,\kappa}} \cdots \sum_{l_d=1}^{N_{d,\kappa}} c_{j_{\kappa}}^{(\kappa)}(l_1, \ldots, l_d)(t) \chi_{j_{\kappa}}^{(\kappa)}(q_1, \ldots, q_d, \kappa)
\]

Exponential Scaling:

Standard:

\[
N_f, MCTDH:
\]

combined:

\[
\left( \tilde{n}_1 / d \right)_f
\]
MCTDH with Mode Combination

\[
(q_1, q_2, q_3, q_4, q_5, q_6, \cdots, q_{f-1}, q_f)
\]

\[
Q_1 \quad Q_2 \quad Q_3 \quad \cdots \quad Q_p
\]

MCTDH wavefunction

\[
\Psi(q_1, \cdots, q_f, t) \equiv \Psi(Q_1, \cdots, Q_p, t)
\]

\[
= \sum_{j_1}^{n_1} \cdots \sum_{j_p}^{n_p} A_{j_1, \cdots, j_p}(t) \prod_{\kappa=1}^{p} \varphi_{j_\kappa}^{(\kappa)}(Q_\kappa, t)
\]

Single-particle functions:

\[
\varphi_{j_\kappa}^{(\kappa)}(Q_\kappa, t) = \sum_{l_1=1}^{N_{1,\kappa}} \cdots \sum_{l_d=1}^{N_{d,\kappa}} c_{j_\kappa l_1 \cdots l_d}^{(\kappa)}(t) \chi_{l_1}^{(\kappa)}(q_{1,\kappa}) \cdots \chi_{l_d}^{(\kappa)}(q_{d,\kappa})
\]
MCTDH with Mode Combination

\[
(q_1, q_2, q_3, q_4, q_5, q_6, \cdots, q_{f-1}, q_f)
\]

\[
Q_1 \quad Q_2 \quad Q_3 \quad \cdots \quad Q_p
\]

MCTDH wavefunction

\[
\Psi(q_1, \cdots, q_f, t) \equiv \Psi(Q_1, \cdots, Q_p, t)
\]

\[
= \sum_{j_1}^{n_1} \cdots \sum_{j_p}^{n_p} A_{j_1, \cdots, j_p}(t) \prod_{\kappa=1}^{p} \varphi_{j_{\kappa}}^{(\kappa)}(Q_{\kappa}, t)
\]

Single-particle functions:

\[
\varphi_{j_{\kappa}}^{(\kappa)}(Q_{\kappa}, t) = \sum_{l_1=1}^{N_{1,\kappa}} \cdots \sum_{l_d=1}^{N_{d,\kappa}} c_{j_{\kappa}l_1 \cdots l_d}^{(\kappa)}(t) \chi_{l_1}^{(\kappa)}(q_{1,\kappa}) \cdots \chi_{l_d}^{(\kappa)}(q_{d,\kappa})
\]

Exponential Scaling:

Standard: \( N^f \), \hspace{1em} MCTDH: \( n^f \), \hspace{1em} combined: \( (\tilde{n}^{1/d})^f \)
1 Multiconfiguration time-dependent Hartree, MCTDH

2 Multi-Layer MCTDH

3 MCTDH and ML-MCTDH viewed as tensor decomposition methods

4 Compact representations of the PES

5 Highlights and Conclusions
Multi-Layer MCTDH

- Mode-combination has proved to be very helpful

- But mode-combination orders larger than 3 or 4 make the propagation of the SPFs infeasible

- Use MCTDH to propagate the SPFs of an underlying MCTDH calculation


ML-MCTDH expansion of wavefunction

$$
\psi(Q_1^1, \ldots, Q_p^1) = \sum_{j_1=1}^{n_1^1} \cdots \sum_{j_p=1}^{n_p^1} A_{1;j_1,\ldots,j_p}^1 \prod_{\kappa_1=1}^{p} \varphi_{j_{\kappa_1}}^{(1;\kappa_1)}(Q_{\kappa_1}^1)
$$
ML-MCTDH expansion of wavefunction

\[ \Psi(Q_1, \ldots, Q_p) = \sum_{j_1=1}^{n_1^1} \cdots \sum_{j_p=1}^{n_p^1} A_{1;j_1,\ldots,j_p}^1 \prod_{\kappa_1=1}^{p} \varphi_{j_{\kappa_1}}^{(1;\kappa_1)}(Q_{\kappa_1}) \]

\[ \varphi_{m}^{(1;\kappa_1)}(Q_{\kappa_1}) = \sum_{j_1=1}^{n_1^2} \cdots \sum_{j_{p_{\kappa_1}}}^{n_{p_{\kappa_1}}} A_{m;j_1,\ldots,j_{p_{\kappa_1}}}^{2;\kappa_1} \prod_{\kappa_2=1}^{p_{\kappa_1}} \varphi_{j_{\kappa_2}}^{(2;\kappa_1,\kappa_2)}(Q_{\kappa_2}) \]
ML-MCTDH expansion of wavefunction

\[
\Psi(Q_1^1, \ldots, Q_p^1) = \sum_{j_1=1}^{n_1^1} \cdots \sum_{j_p=1}^{n_p^1} A_{1; j_1, \ldots, j_p}^1 \prod_{\kappa_1=1}^p \varphi_{j_{\kappa_1}}^{(1; \kappa_1)}(Q_{\kappa_1})
\]

\[
\varphi_{m}^{(1; \kappa_1)}(Q_{\kappa_1}) = \sum_{j_1=1}^{n_1^2} \cdots \sum_{j_{\kappa_1}^m}^{n_{p_{\kappa_1}}^2} A_{m; j_1, \ldots, j_{\kappa_1}}^2 \prod_{\kappa_2=1}^{p_{\kappa_1}} \varphi_{j_{\kappa_2}}^{(2; \kappa_1, \kappa_2)}(Q_{\kappa_2})
\]

\[
\varphi_{m}^{(2; \kappa_1, \kappa_2)}(Q_{\kappa_2}) = \sum_{j=1}^{N_{\alpha}} A_{m; j}^3 \chi_{j}^{(\alpha)}(q_{\alpha})
\]

\[
Q_{\kappa_\ell}^{\ell; \kappa_1, \ldots, \kappa_{\ell-1}} = \{ Q_{1\kappa_1}, \ldots, Q_{p_{\kappa_\ell}}^{\ell+1; \kappa_1, \ldots, \kappa_{\ell} \kappa_\ell} \}
\]
Standard Method and MCTDH trees

Standard Method

MCTDH
MCTDH and ML-MCTDH trees

MCTDH combined

ML-MCTDH
ML-MCTDH tree for naphthalene (48D)
PE-spectrum of naphthalene (48D)  Q. Meng

Gas-phase photoelectron spectrum

Theoretical spectrum

48D ML-MCTDH
Problems studied with the Heidelberg ML-MCTDH package

- Henon-Heiles: 6D, 18D, 1458D
- Pyrazine: 24D, 2E
- Difluorobenzene cation: 30D, 5E
- Naphtalene cation: 48D, 6E
- Antracene cations: 66D, 6E
- Formaldehyde Oxide: 9D, 5E
Conclusions
ML-MCTDH

- ML-MCTDH is capable to treat very large systems with hundreds of degrees of freedom.
- ML-MCTDH is very suitable for studying system/bath problems.
- ML-MCTDH is most useful when using model Hamiltonians. However, model Hamiltonians like the VC-Hamiltonian can be very helpful to investigate real chemical systems.
- ML-MCTDH is very fast in a low accuracy mode but may become costly if a high accuracy is asked for.
1. Multiconfiguration time-dependent Hartree, MCTDH

2. Multi-Layer MCTDH

3. MCTDH and ML-MCTDH viewed as tensor decomposition methods

4. Compact representations of the PES

5. Highlights and Conclusions
Expansion of coefficients

Standard Method

\[ \psi(q_1, \cdots, q_f) = \sum_{i_1}^{N_1} \cdots \sum_{i_f}^{N_f} \psi_{i_1, \cdots, i_f} \chi_{i_1}^{(1)}(q_1) \cdots \chi_{i_f}^{(f)}(q_f) \]
Expansion of coefficients

**Standard Method**

\[
\Psi(q_1, \ldots, q_f) = \sum_{i_1}^{N_1} \cdots \sum_{i_f}^{N_f} \psi_{i_1, \ldots, i_f} \chi_{i_1}^{(1)}(q_1) \cdots \chi_{i_f}^{(f)}(q_f)
\]

**MCTDH**

\[
\psi_{i_1, \ldots, i_f} = \sum_{j_1, \ldots, j_f} A_{j_1, \ldots, j_f} c_{j_1, i_1}^{(1)} \cdots c_{j_f, i_f}^{(f)}
\]
Expansion of coefficients

Standard Method

$$\Psi(q_1, \cdots, q_f) = \sum_{i_1}^{N_1} \cdots \sum_{i_f}^{N_f} \Psi_{i_1, \cdots, i_f} \chi^{(1)}_{i_1}(q_1) \cdots \chi^{(f)}_{i_f}(q_f)$$

MCTDH

$$\Psi_{i_1, \cdots, i_f} = \sum_{j_1, \cdots, j_f} A_{j_1, \cdots, j_f} c_{j_1, i_1}^{(1)} \cdots c_{j_f, i_f}^{(f)}$$

MCTDH combined

$$\Psi_{i_1, \cdots, i_f} = \sum_{j_1, \cdots, j_p} A_{j_1, \cdots, j_p} c_{j_1, i_1}^{(1)} \cdots c_{j_d, i_d}^{(p)} \cdots c_{j_p, i_p}^{(p)}$$
Expansion of coefficients

Standard Method

\[
\psi(q_1, \cdots, q_f) = \sum_{i_1}^{N_1} \cdots \sum_{i_f}^{N_f} \psi_{i_1, \cdots, i_f} \chi^{(1)}_{i_1}(q_1) \cdots \chi^{(f)}_{i_f}(q_f)
\]

MCTDH

\[
\psi_{i_1, \cdots, i_f} = \sum_{j_1, \cdots, j_f} A_{j_1, \cdots, j_f} c^{(1)}_{j_1, i_1} \cdots c^{(f)}_{j_f, i_f}
\]

MCTDH combined

\[
\psi_{i_1, \cdots, i_f} = \sum_{j_1, \cdots, j_p} A_{j_1, \cdots, j_p} c^{(1)}_{j_1, i_1 \cdots i_d} \cdots c^{(p)}_{j_p, i_\cdots i_f}
\]

MCTDH is a decomposition of the wave-function tensor into a (time-dependent) Tucker form!
Expansion of coefficients in ML-MCTDH form

ML-MCTDH (one extra layer)

\[
\Psi_{i_1, \ldots, i_f} = \sum_{j_1, \ldots, j_p} A^{(1)}_{j_1, \ldots, j_p} \left( \sum_{k_1, \ldots, k_{p_1}} A^{(2;1)}_{j_1, k_1, \ldots, k_{p_1}} A^{(3;1,1)}_{k_1, i_1} \cdots A^{(3;1,p_1)}_{k_{p_1}, i_{p_1}} \right) \times \cdots \times \left( \sum_{k_1, \ldots, k_{p_{\kappa_1}}} A^{(2;\kappa_1)}_{j_{\kappa_1}, k_1, \ldots, k_{p_{\kappa_1}}} A^{(3;\kappa_1,1)}_{k_1, i_{\alpha}} \cdots A^{(3;\kappa_1,p_{\kappa_1})}_{k_{p_{\kappa_1}}, i_f} \right) \times \cdots \times \left( \sum_{k_1, \ldots, k_{p_{\kappa_{p}}}} A^{(2;p)}_{j_p, k_1, \ldots, k_{p_{p}}} A^{(3;p,1)}_{k_1, i_{\alpha}} \cdots A^{(3;p,p_{p})}_{k_{p_{p}}, i_f} \right)
\]

Hierarchical Tucker format
Other decomposition methods

CANDECOMP, CP

\[ \Psi(q_1, \cdots, q_f) = \sum_r a_r \varphi_r^{(1)}(q_1) \cdots \varphi_r^{(f)}(q_f) \]

\[ \Psi_{i_1, \cdots, i_f} = \sum_r a_r c_{r, i_1}^{(1)} \cdots c_{r, i_f}^{(f)} \]
Other decomposition methods

CANDECOMP, CP

\[ \Psi(q_1, \cdots, q_f) = \sum_r a_r \varphi_r^{(1)}(q_1) \cdots \varphi_r^{(f)}(q_f) \]

\[ \Psi_{i_1, \cdots, i_f} = \sum_r a_r c_{r, i_1}^{(1)} \cdots c_{r, i_f}^{(f)} \]

Tensor Train (TT) format. Similar to matrix product states. TT can be viewed as a simplified, restricted form of the Hierachical Tucker format (i.e. ML-MCTDH).
1. Multiconfiguration time-dependent Hartree, MCTDH

2. Multi-Layer MCTDH

3. MCTDH and ML-MCTDH viewed as tensor decomposition methods

4. Compact representations of the PES

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

To this end we require the Hamiltonian to be in product form

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

where $\hat{h}_{r}^{(\kappa)}$ operates on the $\kappa$-th particle only.
Product representation of the Hamiltonian

The computation of the Hamiltonian matrix $\langle \Phi_J \mid \hat{H} \mid \Phi_L \rangle$ and the mean-fields $\langle \hat{H} \rangle^{(\kappa)}_{k,l}$ requires the evaluation of multi-dimensional integrals. It is essential that these integrals are done fast. To this end we require the Hamiltonian to be in product form

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

where $\hat{h}^{(\kappa)}_r$ operates on the $\kappa$-th particle only.

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

$$\langle \Phi_J \mid \hat{H} \mid \Phi_L \rangle = \sum_{r=1}^{s} c_r \langle \varphi^{(1)}_{j_1} \mid \hat{h}^{(1)}_r \mid \varphi^{(1)}_{l_1} \rangle \ldots \langle \varphi^{(p)}_{j_p} \mid \hat{h}^{(p)}_r \mid \varphi^{(p)}_{l_p} \rangle$$
The most direct way to the product form is an expansion in a product basis. Hence we approximate some given potential $V$ by

$$V^{\text{app}}(Q_1, \ldots, Q_p) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_p=1}^{m_p} C_{j_1 \ldots j_p} v_{j_1}^{(1)}(Q_1) \cdots v_{j_p}^{(p)}(Q_p)$$
The most direct way to the product form is an expansion in a product basis. Hence we approximate some given potential $V$ by

$$V^{\text{app}}(Q_1, \ldots, Q_p) = \sum_{j_1=1}^{m_1} \ldots \sum_{j_p=1}^{m_p} C_{j_1 \ldots j_p} v_{j_1}^{(1)}(Q_1) \ldots v_{j_p}^{(p)}(Q_p)$$

working with grids:

$$V^{\text{app}}_{i_1 \ldots i_p} = \sum_{j_1=1}^{m_1} \ldots \sum_{j_p=1}^{m_p} C_{j_1 \ldots j_p} v_{i_1 j_1}^{(1)} \ldots v_{i_p j_p}^{(p)}$$
The most direct way to the product form is an expansion in a product basis. Hence we approximate some given potential $V$ by

$$V^{\text{app}}(Q_1, \ldots, Q_p) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_p=1}^{m_p} C_{j_1 \ldots j_p} v^{(1)}_{j_1}(Q_1) \cdots v^{(p)}_{j_p}(Q_p)$$

working with grids:

$$V^{\text{app}}_{i_1 \ldots i_p} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_p=1}^{m_p} C_{j_1 \ldots j_p} v^{(1)}_{i_1 j_1} \cdots v^{(p)}_{i_p j_p}$$

Tucker format!
The coefficients are given by overlap

\[ C_{j_1\ldots j_p} = \sum_{i_1=1}^{N_1} \ldots \sum_{i_p=1}^{N_p} v_{i_1 j_1}^{(1)} \ldots v_{i_p j_p}^{(p)} V_{i_1 \ldots i_p} \]
The coefficients are given by overlap

\[ C_{j_1 \ldots j_p} = \sum_{i_1=1}^{N_1} \ldots \sum_{i_p=1}^{N_p} v_{i_1 j_1}^{(1)} \ldots v_{i_p j_p}^{(p)} V_{i_1 \ldots i_p} \]

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

\[ Q_{kk'}^{(\kappa)} = \sum_{l}^{\kappa} V_{l_1 \ldots l_{\kappa-1} k \kappa+1 \ldots i_p} V_{l_1 \ldots l_{\kappa-1} k' \kappa+1 \ldots i_p} \]
The coefficients are given by overlap

\[ C_{j_1...j_p} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_p=1}^{N_p} v_{i_1j_1}^{(1)} \cdots v_{i_pj_p}^{(p)} V_{i_1...i_p} \]

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

\[ Q^{(\kappa)}_{\kappa k'} = \sum_{i}^{\kappa} V_{i_1...i_{\kappa-1}k_{\kappa+1}...i_p} V_{i_1...i_{\kappa-1}k'_{\kappa+1}...i_p} \]

POTFIT is feasible for at most $10^9$ grid points (7 DOF, say).
The coefficients are given by overlap

\[
C_{j_1 \ldots j_p} = \sum_{i_1=1}^{N_1} \cdots \sum_{i_p=1}^{N_p} v_{i_1 j_1}^{(1)} \cdots v_{i_p j_p}^{(p)} V_{i_1 \ldots i_p}
\]

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

\[
Q^{(\kappa)}_{kk'} = \sum_{I}^{\kappa} V_{i_1 \ldots i_{\kappa-1} k_{\kappa+1} \ldots i_p} V_{i_1 \ldots i_{\kappa-1} k'_{\kappa+1} \ldots i_p}
\]

POTFIT is feasible for at most $10^9$ grid points (7 DOF, say).

Multi-grid Potfit (MGPF) and Monte Carlo Potfit (MCPF)

**MGPF**

- Chose a fine \( (N_\kappa) \) and a coarse \( (n_\kappa) \) product grid. The coarse grid should be part of the fine grid.
- Perform a full (i.e. exact) POTFIT on the coarse grid.
- Interpolate the SPPs to the fine grid \( \tilde{\kappa} = \text{fine-grid} \):

\[
\tilde{v}^{(\kappa)} = \tilde{\rho}^{(\kappa)} \rho^{(\kappa)^{-1}} v^{(\kappa)}
\]

- To be accurate, the determination of the coefficients requires now the inversion of a huge matrix.
- A Boltzmann weighting is easy to include.
**Multi-grid Potfit (MGPF) and Monte Carlo Potfit (MCPF)**

**MGPF**
- Chose a fine ($N_\kappa$) and a coarse ($n_\kappa$) product grid. The coarse grid should be part of the fine grid.
- Perform a full (i.e. exact) POTFIT on the coarse grid.
- Interpolate the SPPs to the fine grid ($\tilde{v} = \text{fine-grid}$):

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

**MCPF**
- Perform all "integrations" over the grid by Monte Carlo.
- To be accurate, the determination of the coefficients requires now the inversion of a huge matrix.
- A Boltzmann weighting is easy to include.
Potfit and its MG and MC variants express the potential tensor in a Tucker format. But MCTDH does not require this structure, a CANDECOMP is sufficient. As the latter can be more compact, we want to further decrease the size of the potential representation by reducing the Tucker format generated by MG- or MC-Potfit to a CANDECOMP. But how to do that?

Potfit and its MG and MC variants express the potential tensor in a Tucker format. But MCTDH does not require this structure, a CANDECOMP is sufficient. As the latter can be more compact, we want to further decrease the size of the potential representation by reducing the Tucker format generated by MG- or MC-Potfit to a CANDECOMP. But how to do that?

As there is ML-MCTDH, one may think of ML-POTFIT. This will lead to a more compact representation, but not to a faster evaluation, because MCTDH cannot make use of the hierarchical Tucker format structure.

However, ML-MCTDH can do!

Hierarchical representation of a multidimensional function

\[ \tilde{V}(\mathbf{q}) = V^{(0)} + \sum_{\alpha=1}^{f} V^{(1)}_{\alpha}(q_{\alpha}) + \sum_{\alpha<\beta}^{f} V^{(2)}_{\alpha\beta}(q_{\alpha}, q_{\beta}) + \sum_{\alpha<\beta<\gamma}^{f} V^{(3)}_{\alpha\beta\gamma}(q_{\alpha}, q_{\beta}, q_{\gamma}) \cdots \]
High dimensional model representation, HDMR

Hierarchical representation of a multidimensional function

\[ \tilde{V}(q) = V^{(0)} + \sum_{\alpha=1}^{f} V^{(1)}_{\alpha}(q_{\alpha}) + \sum_{\alpha<\beta} V^{(2)}_{\alpha\beta}(q_{\alpha}, q_{\beta}) + \sum_{\alpha<\beta<\gamma} V^{(3)}_{\alpha\beta\gamma}(q_{\alpha}, q_{\beta}, q_{\gamma}) \cdots \]

The component functions (clusters) are determined as:

\[ V^{(0)} = V(a) \]
\[ V^{(1)}_{\alpha}(q_{\alpha}) = V(q_{\alpha}; a^{\alpha}) - V^{(0)} \]
\[ V^{(2)}_{\alpha\beta}(q_{\alpha}, q_{\beta}) = V(q_{\alpha}, q_{\beta}; a^{\alpha\beta}) - V^{(1)}_{\alpha}(q_{\alpha}) - V^{(1)}_{\beta}(q_{\beta}) - V^{(0)} \]

Unfortunately, the number of clusters increases strongly with order. Possible improvements:
- Perform the cluster expansion in combined modes
- One may use more than one reference point
- One may use a reference path rather than a reference point
High dimensional model representation, HDMR

Hierarchical representation of a multidimensional function

\[ \tilde{V}(q) = V^{(0)} + \sum_{\alpha=1}^{f} V^{(1)}_{\alpha}(q_\alpha) + \sum_{\alpha<\beta}^{f} V^{(2)}_{\alpha\beta}(q_\alpha, q_\beta) + \sum_{\alpha<\beta<\gamma}^{f} V^{(3)}_{\alpha\beta\gamma}(q_\alpha, q_\beta, q_\gamma) \cdots \]

The component functions (clusters) are determined as:

\[ V^{(0)} = V(a) \]
\[ V^{(1)}_{\alpha}(q_\alpha) = V(q_\alpha; a^\alpha) - V^{(0)} \]
\[ V^{(2)}_{\alpha\beta}(q_\alpha, q_\beta) = V(q_\alpha, q_\beta; a^{\alpha\beta}) - V^{(1)}_{\alpha}(q_\alpha) - V^{(1)}_{\beta}(q_\beta) - V^{(0)} \]

Unfortunately, the number of clusters increases strongly with order.

Possible improvements:
- Perform the cluster expansion in combined modes
- One may use more than one reference point
- One may use a reference path rather than a reference point
Tunneling splitting in malonaldehyde

9 Atoms, 21 degrees of freedom

1. Multiconfiguration time-dependent Hartree, MCTDH

2. Multi-Layer MCTDH

3. MCTDH and ML-MCTDH viewed as tensor decomposition methods

4. Compact representations of the PES

5. Highlights and Conclusions
Highlights and Breakthroughs

1990, Very first MCTDH publication, Meyer, Manthe, Cederbaum
1999, pyrazine, 24D, 2E, Raab, Worth, Meyer, Cederbaum
2003, Dissipative quantum dynamics, 61D, Nest, Meyer
2005, Vibronic spectrum of C$_5$H$_4^+$, 21D 5E, Markmann et al
2007, IR spectrum of H$_5$O$_2^+$, (15D) Vendrell et al
2008, Tunneling dynamics of bosons, Zöllner et al
2009, Isotopologues of H$_5$O$_2^+$, (15D) Vendrell et al
2011, 2014, Tunnelling splittings in malonaldehyde, 21D, Schröder, Meyer
2013, Vibronic dynamics of naphthalene (48D,6E) and anthracene (66D,6E) cations, Meng, Meyer
Conclusions

MCTDH, realistic problems with 5 to 9 atoms

- Search for good coordinates.
- Deriving the KEO can be cumbersome, but it is a solved problem.
- Finding a compact representation for the PES is a major problem for molecules with 5 or more atoms. The PES representation is often the source of largest errors. Work on improving PES-representations is in progress.
- Finally, the MCTDH calculation as such may take a considerable amount of CPU-time, but MCTDH is stable and we can check its accuracy.
People, who made the Heidelberg MCTDH package

- Graham Worth, Birmingham (MCTDH, pyrazine)
- Fabien Gatti, Montpellier (Kinetic energy operators)
- Oriol Vendrell, Hamburg (ML-MCTDH, Zundel-cation)
- Michael Brill (Parallelization of MCTDH)
- Andreas Raab (Density operator propagation)
- Markus Schröder, Heidelberg (Malonaldehyde, MC-Potfit)
- Frank Otto, Hong Kong (ML-MCTDH, ML-Potfit)
- Daniel Pelaez-Ruiz, Lille (MG-Potfit, H$_3$O$_2^-$)
- Qingyong Meng, Dalian (ML calculations with VCH)

M. Beck, A. Jäckle, M.-C. Heitz, S. Wefing, S. Sukiasyan, Ch. Cattarius, P. S. Thomas, K. Sadri and others.
Thank you!

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