



HAL
open science

On the sum-of-products to product-of-sums transformation between analytical low-rank approximations in finite basis representation

Ramón Panadés-Barrueta, Natasa Nadoveza, Fabien Gatti, Daniel Peláez

► **To cite this version:**

Ramón Panadés-Barrueta, Natasa Nadoveza, Fabien Gatti, Daniel Peláez. On the sum-of-products to product-of-sums transformation between analytical low-rank approximations in finite basis representation. The European Physical Journal. Special Topics, 2023, 232 (12), pp.1897-1904. 10.1140/epjs/s11734-023-00928-z . hal-04304825

HAL Id: hal-04304825

<https://hal.science/hal-04304825v1>

Submitted on 25 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the sum-of-products to product-of-sums transformation between analytical low-rank approximations in finite basis representation

Ramón L. Panadés-Barrueta^{†,*}, Natasa Nadoveza[‡], Fabien Gatti[‡] and Daniel Peláez^{*,‡}

[†]*Faculty of Chemistry and Food Chemistry, Technische Universität Dresden, 01069 Dresden, Germany*

[‡]*Université Paris-Saclay, CNRS, Institut des Sciences Moléculaires d'Orsay, 91405, Orsay, France.*

E-mail: ramon_lorenzo.panades-barrueta@tu-dresden.de;
daniel.pelaez-ruiz@universite-paris-saclay.fr

Abstract

In this work we analyse and compare different possible strategies for the transformations among low-rank (i.e. few number of terms) tensor approximations. The motivation behind this is to achieve compact yet accurate representations of potential-like operators (scalar fields) in symbolic or analytical form. We do this analysis from a formal and from a numerical perspective. Specifically, we concentrate on Tucker and Canonic Polyadic ansätze. We introduce the sum-of-product finite basis representations (SOP-FBR) for both. Here, the factor matrices (aka single-particle functions) are approximated through a set of auxiliary basis functions, specific to the system. In this way, analytical, grid-independent, low-rank expressions can be obtained. We illustrate how finite-precision arithmetic hinders transformations among all these forms. The solution to this issue seems to adapt current algorithms to high-precision arithmetic at the expense of an increase in CPU times.

Introduction

Accurate and compact low-rank tensor expressions are essential to achieve high-performance in many different mathematical, engineering or physical applications. Low-rank refers to the fact that these type of mathematical expressions provide a good global approximation to a set of reference data structured in the form of a multidimensional array. This is also of particular relevance in Deep Learning where over-parametrized models (weight matrices) can be totally (or partially) replaced by such type of approximations.¹ Specifically, the most significant advances in the field of quantum dynamical simulations have been linked to the introduction of novel *ansätze* for the efficient representation of multidimensional objects such as the wave function (WF). The interested reader is referred to one of our recent publications in which a comprehensive summary on the topic is presented.² Grid-based methods require quantities to be expressed on a grid. In other words, they are represented as multidimensional arrays of data aka tensors (V_{i_1, \dots, i_f}). In the case of molecular systems, our main interest, these

tensors are typically of $f = 3N - 6(5)$ dimensions (D). Hereafter, we shall consider an initial f -dimensional tensor $V \in \mathbb{R}^{N_1 \times \dots \times N_f}$, consisting of $\prod_{\kappa}^f N_{\kappa}$ data-points. This number being large enough not to fit in memory. In such case, a tensor decomposition becomes mandatory. Arguably, in the molecular context the most widely employed tensor decomposition methods rely on grid-based sum-of-products *ansätze* such as Tucker and the canonical polyadic (CP) form, respectively Eqs.1 and 2.

$$V_I^{\text{Tucker}} = \sum_J^{[m]} C_J \Omega_{IJ} \quad (1)$$

$$V_I^{\text{CP}} = \sum_{r=1}^R \lambda_r \Omega_{I_r} \quad (2)$$

where $I \equiv (i_1 \dots i_f)$ and $J \equiv (j_1 \dots j_f)$, $[m]$ represents each of the maximum terms in the sums, and Ω are Hartree products of single-particle potentials (SPP, aka factor matrices). It should be noted that for the κ -th degree of freedom $i_{\kappa} = 1, \dots, N_{\kappa}$ and $j_{\kappa} = 1, \dots, m_{\kappa}$, with $m \ll N$. Though formally similar, there are major differences between them. For instance the CP SPPs are not orthogonal (though normalized) whereas the Tucker ones are orthonormal, both on grid and on basis space.³ This extra degree of flexibility in the former leads to very compact CP expansions when compared to Tucker ones. This is evident by the nature of data-structure associated to the expansion coefficients (aka core tensor): a tensor in the case of Tucker (C_J , with $\prod_{\kappa=1}^f m_{\kappa}$ elements, with $m \ll N$) and a vector in the case of CP ($\{\lambda_r\}_{r=1}^R$). Despite the favourable ratio in number of data between the Tucker expression and the original tensor, an exponential scaling in the number of core elements is inherent to any Tucker expression. This makes a priori CP approaches preferable. In what follows, we will mainly refer to grid-based quantum dynamical problems. More specifically within the context of multiconfigurational approaches such as the Multiconfiguration Time-Dependent Hartree (MCTDH) or the Variational Multiconfigurational Gaussian (vMCG) within their most common implementations: Heidelberg MCTDH⁴ and Quantics,⁵ respectively. However, it should be clear that the methods and transformations described here are by no

means restricted to this field and/or its applications. They are also relevant in engineering⁶ as well as in model reduction for Deep Learning.¹

MCTDH, in particular, makes extensive use of these two forms for the compressed representation of grid-based quantities (such as the potential operator). Accordingly, major and sustained efforts have been devoted within this community in order to achieve such low-rank expressions from unstructured initial tensors (e.g. scalar field expressed on a grid).⁷⁻¹¹ It should be realised that these algorithms *refit* the data-points arranged on a grid (e.g. potential on a grid) into a collection of sets of structured numbers (vectors, matrices, tensors), the so-called low-rank approximation which can be appropriately operated to retrieve an approximation to the targeted values. A second alternative is the use of analytical expressions of the potential directly in such a product form, as is the case of a commonly employed benchmark system, HONO.¹² This approach has obvious advantages over grid-based representations. First, it is in principle possible to directly implement such expressions symbolically, as is the case in MCTDH with the so-called `operator` file which encodes the system Hamiltonian. This requires the expansion functions being *simple* expressions fulfilling the SOP condition. And second, within a given set of boundaries, these forms are independent of the grid. These, among other reasons,² led us to suggest the use of a set of basis to analytically represent the SPPs. In this way, the original algebraic refitting problem is turned into an optimization one.² This idea was first implemented using a Tucker form:

$$V(q_1, \dots, q_f) = \sum_{j_1}^{m_1} \cdots \sum_{j_f}^{m_f} C_{j_1, \dots, j_f} \prod_{\kappa} \left[\sum_{\mu}^{t_{\kappa}} c_{\mu, j_{\kappa}}^{(\kappa)} T_{\mu}(q_{\kappa}) \right] \quad (3)$$

where, for simplicity, we have assumed the same number of auxiliary functions per SPP within a given degree of freedom (DOF). More recently, we have also introduced the CP counterpart of SOP-FBR (CP-FBR) and presented its practical implementation.¹³ The actual procedure differs substantially from that of SOP-FBR owing to the intrinsic difficulties

associated to the CP form.¹⁴ In the Theory section, we discuss our current algorithm.¹³ It should be emphasized that the FBR form enables rebuilding a potential tensor on grids of any density within the provided boundaries.

In what follows, we will assume as starting point a SOP-FBR expression obtained from a fit using either: (i) a set of reference geometries ($\vec{\mathbf{R}} \equiv (q_1, \dots, q_f) \equiv (i_1, \dots, i_f)$) and associated energies ($V(i_1, \dots, i_f)$) in tensor form; or (ii) a *judiciously* chosen list of reference points and associated energies ($\{\vec{\mathbf{R}}_i, \mathbf{E}_i\}_{i=1}^N$). Concerning this last case, we have recently shown that it is possible to generate such a list in an automated manner using the network of stationary points (minima and first-order saddle points) of a PES as reference.¹⁵ We employed the AutoMekin software package for this.¹⁶ This software can be used also for non-covalent interacting systems consisting of n-fragments.¹⁷

As we will discuss later, despite its simultaneous SOP and analytical nature, Eq.3 is not suitable for the direct implementation of an MCTDH `operator` file. This file constitutes the actual definition of the system Hamiltonian and for that a set of *simple* symbolic expressions (*e.g.* powers, exponential and trigonometric functions) are used. There are two reasons preventing SOP-FBR form being suitable to this: (i) a typical SOP-FBR consists of too many terms (m^f) and (ii) the SPPs do not have a *simple* form since they are themselves a sum of basis functions.

In this work, we show that starting from common low-rank SOP FBR approximations such as Tucker or CP, it is formally possible to obtain a product-of-sums (POS) expression potentially compatible for a direct implementation into such an `operator` file. First, we present the formally exact algebraic transformations needed for this, using as starting points both SOP-FBR and CP-FBR. Finally, we address the numerical implementation of the transformation and show that numerical bottlenecks plague the whole process, thus hindering

the direct achievement of our initial goal.

Theory

In this section, we present the description of the theoretical methods employed in this work. We start by succinctly describing our recent CP-FBR approach, specifically the approximation thereof which we have used. Then we thoroughly discuss the transformation from SOP to POS.

The Canonical Polyadic Finite Basis Representation (CP-FBR)

As any canonic polyadic form, CP-FBR form is much more compact than Tucker based SOP-FBR. Indeed, one *replaces* the core tensor with a vector of a size R (rank of the decomposition):

$$V(q_1, \dots, q_f) = \sum_{r=1}^R \lambda_r \prod_{\kappa=1}^f \sum_{\mu=1}^{t_\kappa} B_{\mu r}^{(\kappa)} T_\mu(q_\kappa) \quad (4)$$

It should be noted that, for the sake of legibility, we have changed our initial notation of the expansion coefficients from c to B (*vide infra*). Note in passing that the basis functions (T) can be different for different degrees of freedom.² To begin with, we would like to stress the common features as well as to point out the differences between the two aforementioned ansätze. We will also briefly discuss the sensitivity of the X-FBR form with respect to variations on the auxiliary basis functions. In both cases, one can achieve the suitable SOP form from the initial multidimensional function through tensor decomposition methods, which, depending on the algorithm, require either a full tensor or a list of geometries sampled from it. As mentioned in the Introduction, in case of SOP-FBR, it has been shown that the algebraic process of the decomposition can be replaced by an optimization one.² The advantage of this approach is that the geometries are not restricted to the grid definition anymore. Owing to the large degree of flexibility (number of coefficients) of the SOP-FBR

form, one could use a set of approximated functions as SPPs and then simply tweak the core tensor coefficients in order to minimize some objective function. Conversely, due to different nature of the parameters in CP form, the direct optimization is not as efficient as it is in the Tucker form. Indeed, the information of the potential is now contained in the factor matrices (SPPs) and the weights are merely the products of their norms. Consequently, if one would use approximated functions as SPPs in a CP-FBR form, the optimization would require the optimization of the fitting basis, rather than the weights. Unfortunately, the RMSE as an objective function being highly sensitive to variations in the expansion coefficients makes the optimization process very difficult to converge.¹⁴ This can be easily illustrated. For this, we considered an initial CP-ALS decomposition of rank $R = 15$. Then, we applied numerical noise to the factors in the form of random variations at each discrete point within $\pm 0.5\%$ to obtain what we will further refer to as *noisy factors*, which will serve as approximated SPPs. After fitting the so-obtained noisy factors with Chebyshev polynomials of order 10, we optimized the fitting coefficients using Powell algorithm. Initial RMSE of 188.20 cm^{-1} decreased to 70.44 cm^{-1} after approximately $7.35 \cdot 10^6$ iterations. In the Figure 1 we show the changes of RMSE values in cm^{-1} (in logarithmic scale for better visibility) during the optimization, on the left for the first 400 and on the right for the last 400 iterations. One can clearly see that the values are reaching 10^{10} , indicating the high sensitivity of the objective function to the optimization parameters

As a consequence, CP-FBR requires an appropriate initial guess CP decomposition. We have observed that, fortunately, this decomposition can be performed on a small subset (coarse grid) of the grid needed for dynamics. The obtained factors are then fitted using an appropriate basis set (potential dependent), in our case Chebyshev polynomials. The optimal number of basis functions (or highest order in the case of polynomials) for each degree of freedom can be determined so that it minimizes an objective function, *e.g.* RMSE, on a given validation set. This set can include any list of representative points which are not mapped on the coarse grid. In this way, one can assess interpolation power of the obtained analytical

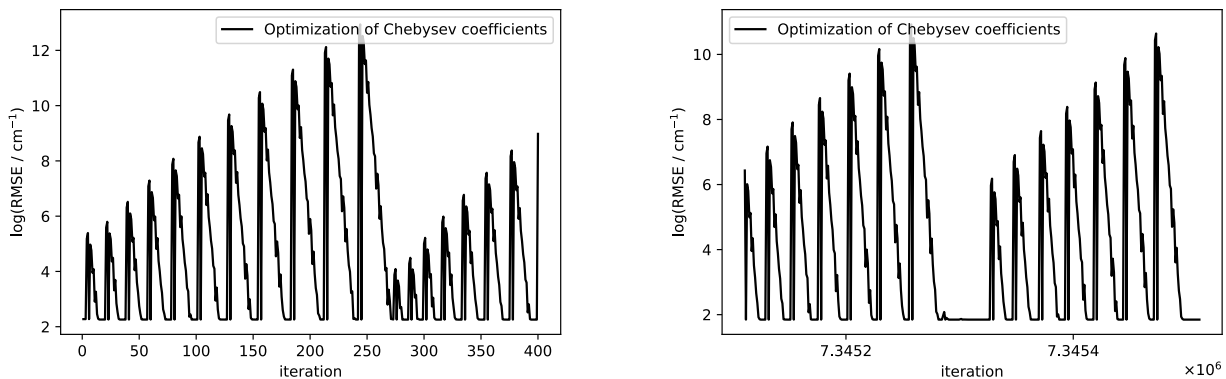


Figure 1: Optimization of the CP-FBR form obtained by fitting noisy factors using fitting polynomial of order 10. Initial RMSE of CP-FBR was 188.20 cm^{-1} and after 734511 iterations (using Powell algorithm) it decreased to 70.44 cm^{-1} . The values of RMSE are given in cm^{-1} in logarithmic scale, so that the span of values is better visible

CP-FBR form. Typically, we use a validation grid which is slightly bigger than the coarse one. It is our experience that the so-prepared CP-FBR form preserves well the RMSE upon interpolation to grids of any density. In our current example, however, we simplified this scheme. Since our goal is to compare to SOP-FBR in the context of the sum-of-products to product-of-sums transformation, we directly generated a series of CP-FBR expansions using (and optimizing) an increasing number of fitting polynomials up to a maximum of 10 which, for the sake of comparison, coincides with the SOP-FBR case.

The transformation from sum-of-products to product-of-sums

We will start our analysis with a grid-based Tucker form:

$$V_{i_1, \dots, i_f} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \prod_{\kappa=1}^f v_{i_\kappa j_\kappa}^{(\kappa)} \quad (5)$$

As discussed above, this expression can be readily turned into an analytical expression through the use of a suitable set of basis functions adapted to the problem in hand. As a rule of thumb, topologically similar potentials share the same type of basis functions. Specifically, we have observed that bound degrees of freedom (stretching, valence angles) are

well represented through series of Chebyshev polynomials. These will be our default choice in this work. More specifically, these orthogonal polynomial series are interesting because (i) every single polynomial can be represented as a linear combination of orthogonal monomials (ii) orthogonal polynomials are convenient for curve fitting because they can reduce the multicollinearity of the variances of the parameters in the fit. As a result, we can transform Equation 5 as follows:

$$V(q_1, \dots, q_f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \prod_{\kappa=1}^f v_{j_\kappa}^{(\kappa)}(q_\kappa) \quad (6)$$

We introduce a series of Chebyshev polynomials of first kind $T_\mu(q_\kappa)$ of order t_{j_κ} as well as a set of expansion coefficients $B_{\mu j_\kappa}^{(\kappa)}$ for the basis functions:

$$v_{j_\kappa}^{(\kappa)}(q_\kappa) = \sum_{\mu_{j_\kappa}=1}^{t_{j_\kappa}} B_{\mu_{j_\kappa} j_\kappa}^{(\kappa)} T_{\mu_{j_\kappa}}(q_\kappa) \quad (7)$$

For practical reasons, in the previous equation we have changed the usual notation of the expansion coefficients from $c_{\mu, j_\kappa}^{(\kappa)}$ to $B_{\mu_{j_\kappa} j_\kappa}^{(\kappa)}$. As indicated, we will consider a simplification that has no impact whatsoever in our conclusions, we will consider the same number of auxiliary basis sets for every degree of freedom (DOF). Hence $t_{j_\kappa} = t_\kappa \forall j_\kappa$. In this case, equation 6 can be further simplified using the relationship between product of sums (POS) and sums of products (SOP):

$$V(q_1, \dots, q_f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} \prod_{\kappa=1}^f B_{\mu_\kappa j_\kappa}^{(\kappa)} T_{\mu_\kappa}(q_\kappa) \quad (8)$$

Applying the associative property of multiplication to the product in Equation 8, it is possible to write it in a more convenient way:

$$V(q_1, \dots, q_f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} \prod_{\kappa=1}^f B_{\mu_\kappa j_\kappa}^{(\kappa)} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (9)$$

Then, by commutativity and associativity of the sums, one can rewrite Equation 9 as:

$$V(q_1, \dots, q_f) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} C_{j_1 \dots j_f} \prod_{\kappa=1}^f B_{\mu_\kappa j_\kappa}^{(\kappa)} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (10)$$

or perhaps, more conveniently, as:

$$V(q_1, \dots, q_f) = \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \prod_{\kappa=1}^f B_{\mu_\kappa j_\kappa}^{(\kappa)} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (11)$$

where we have just re-ordered the sums. Now, taking a closer look to the inner part of expression Equation 11, one can notice that it has the familiar Tucker structure. One can then define the Tucker decomposition of a new tensor $A_{\mu_1 \dots \mu_f}$ into the same initial core tensor $C_{j_1 \dots j_f}$ and the factor matrices $B_{\mu_\kappa j_\kappa}^{(\kappa)}$.

$$A_{\mu_1 \dots \mu_f} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_f=1}^{m_f} C_{j_1 \dots j_f} \prod_{\kappa=1}^f B_{\mu_\kappa j_\kappa}^{(\kappa)} \quad (12)$$

This tensor can be considered as the representation of the original core tensor C_J in our basis set. Hence, in a sense, this object encapsulates all the relevant information about the PES. Introducing this new quantity, we arrive at our sought simplified POS-FBR expression:

$$V(q_1, \dots, q_f) = \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} A_{\mu_1 \dots \mu_f} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (13)$$

This expression deserves some comments. First, the POS-FBR is, obviously, analytical, and presents SPPs which are less complex than the SOP-FBR ones. This unlocks one of our goals, the writing of the operator file. Equation 13 is of the form required by the MCTDH operator file (numbers are simple examples):

...

A(mu1, ..., muf) | cheb0 | cheb1 | cheb2

...

where the first column represent elements of the \mathcal{A} tensor and the label **cheb**(μ_κ) represent the element $T_{\mu_\kappa}(q_\kappa)$. The latter have been implemented in the Heidelberg MCTDH software package source code by the authors. However, POS-FBR still presents a core tensor that grows exponentially, as any Tucker, thus it still leads to far too many Hamiltonian terms. To achieve a more compact expression, one would naïvely expect to be able to perform some kind of diagonalisation on the core tensor and, by truncation, arriving at much fewer terms. This is, generally, not possible. However, one can achieve a kind of *pseudo-diagonalisation* of the core together with rank reduction (from $\prod_\kappa^f \mu_\kappa$ to R_A) through the use of a CP-ALS algorithm on the A-tensor. It should be indicated that this idea was already used by Schröder in his grid-based `natpot2cpd` routine of the MCTDH software package.

$$V(q_1, \dots, q_f) = \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} \sum_{r_A=1}^{R_A} \lambda_{r_A} \prod_{\kappa=1}^f \phi_{\mu_\kappa r_A}^{(\kappa)} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (14)$$

where $\{\phi\}$ are the factor matrices of the tensor A. Note that the CP decomposition has been performed on the A-tensor, hence the extra A subindices. On the other hand, one can readily see that if starting from a CP-FBR form:

$$V(q_1, \dots, q_f) = \sum_{r=1}^R \lambda_r \prod_{\kappa=1}^f \sum_{\mu_\kappa=1}^{t_\kappa} B_{\mu_\kappa r}^{(\kappa)} T_{\mu_\kappa}(q_\kappa) \quad (15)$$

one can also perform the same POS transformation and get:

$$V(q_1, \dots, q_f) = \sum_{\mu_1=1}^{t_1} \cdots \sum_{\mu_f=1}^{t_f} \sum_{r=1}^R \lambda_r \prod_{\kappa=1}^f B_{\mu_\kappa r}^{(\kappa)} \prod_{\kappa=1}^f T_{\mu_\kappa}(q_\kappa) \quad (16)$$

where:

$$A_{\mu_1 \cdots \mu_f}^{CP} = \sum_{r=1}^R \lambda_r \prod_{\kappa=1}^f B_{\mu_\kappa r}^{(\kappa)} \quad (17)$$

It is interesting to realize that though formally identical, Eqs. 14 and 16 carry a subtly

different meaning. In the former, the rank (R_A) corresponds to that of the A-tensor, the representation of the original Tucker core tensor in our FBR basis. On the other hand, the rank of the latter (R) is directly the rank of the potential tensor (or a representative subset of it, such as a coarser grid, as in the case of CP-FBR¹³).

The proposed SOP-POS transformation is prone to numerical instabilities due to the limitations of floating point arithmetic. Indeed, the coefficients included in the original SOP expansion cover a wide numerical interval and adding numbers of very different magnitudes (or subtracting numbers of similar magnitudes) can lead to severe losses of precision due to round-off errors. Of course, smart sum algorithms (e.g. the Kahan algorithm) instead of *naïve* sequential sums can improve on this, but they are not always readily available. Examples of this phenomenon can be found in the literature.¹⁸⁻²⁰ A particularly suited case can be found in a paper by Jiang *et al.*,²¹ in which the authors study the errors arising from the representation of the polynomial

$$p(x) = (x - 0.75)^7(x - 1)^{10} \tag{18}$$

as a series of Chebyshev polynomials of the first kind (the exact coefficients are given in Appendix A of their paper).²¹ The formally exact transformation generates a series of 17-order Chebyshev polynomials. To clearly illustrate the loss of precision, we have reproduced their aforementioned example in Figure 2.

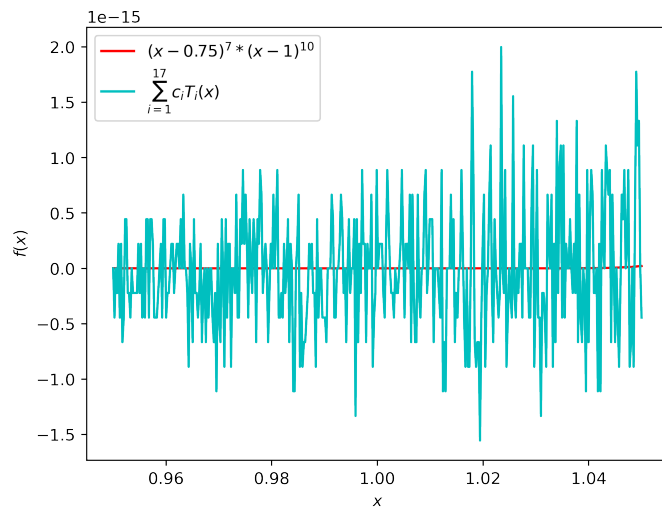


Figure 2: Loss of precision when representing a polynomial of degree 17 in a vicinity of its stationary point by an equivalent (exact) Chebyshev series.

As it can be observed, the original expression $p(x)$ is a smooth trace whereas the *formally* exact Chebyshev expansion is highly oscillating. These issues will rapidly grow with dimensionality, with the size of the approximated polynomial and the concomitant number of operations to be performed with them. It should be noted that in this example, the largest coefficient (in absolute value) is $\sim 10^3$ as in our own expansions (see discussion around Table 1).

Results and discussion

In this section, we shall briefly present and discuss the actual implementation and numerical resolution of the aforementioned transformations. We will use a simple 3D model potential, the Polyanski–Jensen–Tennyson $H_2^{16}O$ PES known as PJT2.²²

From SOP-FBR to POS-FBR (and back again)

As discussed in the Theory section, to start our analysis we will use as reference our analytical SOP-FBR PES for the H_2O molecule. In Ref.² we present a detailed description of its

computation. Here we will briefly sketch its main features. The optimisation used a grid of [15, 15, 15] reference structures generated from a sampling of normal distributions of coordinates (mode ordering is $[r_1, r_2, \theta]$) centered at the water molecule equilibrium structure.² Moreover, a set of initial SPPs was obtained from an initial Tucker decomposition using an inexpensive reparametrized semiempirical potential (MOPAC software²³) expressed on the full grid.² The resulting model had a total of 290 parameters:

$$V(q_1, q_2, q_3) = \sum_{j_1=1}^5 \sum_{j_2=1}^5 \sum_{j_3=1}^5 C_{j_1 j_2 j_3} \prod_{\kappa=1}^3 \sum_{\mu_{j\kappa}=1}^{11} B_{\mu_{j\kappa} j_\kappa}^{(\kappa)} T_{\mu_{j\kappa}}(q_\kappa) \quad (19)$$

with $(q_1, q_2, q_3) \equiv (r_1, r_2, \theta)$. Given the size of the parameter space, a full direct optimization was possible (see original paper for details). The RMSE of the resulting SOP-FBR PES was of 0.686 cm^{-1} on the full grid.

The transformation from SOP-FBR to POS-FBR (see Eq.13) involves the n-mode product between our SOP-FBR optimized core tensor (C_J) and the matrices of the optimal Chebyshev expansion coefficients (\mathbf{B}) (see Eq.19):

$$A = C \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \times_3 \mathbf{B}^{(3)} \quad (20)$$

As an example, the individual product for the first DOF reads:

$$(C \times_1 \mathbf{B}^{(\kappa)})_{\mu_1, \dots, j_f} = \sum_{j_1=1}^{m_1} C_{j_1, \dots, j_f} B_{\mu_1 j_1}^{(1)} \quad (21)$$

After doing these operations, the resulting POS-FBR expansion yields a RMSE of 2940580.20 cm^{-1} on the grid. Given that the transformation between these two expressions is formally exact, one would expect a RMSE value closer to the original SOP-FBR one or merely 0.686 cm^{-1} (and not such an *uncanny* value). Moreover, after 87301 iterations of the Powell algorithm the RMSE was still of $252874.53 \text{ cm}^{-1}$. Please note that global optimizers might lead to such range of values and then arrive at meaningful values (see for instance a numerical

example in Figure 1). In the following, we will try to disentangle the reasons behind this erroneous value. It should be highlighted that all of these calculations have been carried out using *extended precision doubles*, (`numpy.longdouble`)¹ in combination with an improved summation algorithm²⁴ implemented in the Python `fsum` function ².

The first thing to note is that the quality (RMSE) of an expansion is very sensitive to the precision of Chebyshev polynomial coefficients. For instance, by simply adding a small noise ($7 \cdot 10^{-9}$) to the original SOP parameters, without any further transformation, an increased RMSE value of 10817.13 cm^{-1} is obtained. This numerical instability is propagated through the successive series of contractions, thus leading to a degradation of the quality of the obtained RMSE. In fact, this is very likely related to the loss of precision arising from the tensor products, as the numbers involved in these calculations might differ by several orders of magnitude. This well-known dramatic effect can be illustrated for the case of polynomials of high degree, like the ones generated after the full contraction.

The SOP-POS transformation from CP-FBR

In view of the previous results, we will now take advantage of the compactness of the CP-FBR expression. First it should be noted that the size of the A-tensor depends solely on the number of fitting functions so that in what follows we shall consider a CP A-tensor of size identical to the one in SOP. There are, however, two main differences regarding its computation in the CP case: (i) the range of values of the so-called CP weights (λ_r) is significantly smaller than the one from the Tucker core tensor; and (ii) the number of operations to obtain it is much smaller in the case of CP, given that summation of products of Chebyshev coefficients is done over R terms instead of the $\prod_{\kappa=1}^f m_{\kappa}$ terms in Tucker. Both of them are responsible for the much less pronounced loss of the precision in CP form when transforming the initial expression into the POS one. Indeed, in the case of CP-FBR, the

¹(see <https://numpy.org/doc/stable/user/basics.types.html#extended-precision>)

²(<https://docs.python.org/3/library/math.html>)

initial RMSE of 0.96 cm^{-1} after transformation leads to an RMSE of 21480.5 cm^{-1} , 2 orders of magnitude less than that of Tucker POS. Nevertheless, the resulting RMSE is still not acceptable. It could be further improved by reducing the number of fitting polynomials, and this effect can be observed in Figure 3, where we represented in logarithmic scale the ratio of RMSE after transformation and original one, as a function of order of fitting polynomials (Chebyshev order, t_κ). This clearly shows that the RMSE is better preserved during the SOP-POS transformation for lower order of fitting polynomials. But the drawback is that such lower number might not be sufficient to achieve good quality of the original CP-FBR form. To illustrate this, we show in Table 1 the variation of the RMSE with the order of the fitting polynomial. As it can be seen, the best balance between initial quality of the CP-FBR and the possibility to preserve it during the transformation was obtained for $t_\kappa = 8$, where RMSE is 7.34 cm^{-1} before and 8.98 cm^{-1} after. To further illustrate the effect of precision loss associated, we have compared the POS transformations using the same number of basis and, hence, the concomitant number of operations but in floating point number representations of decreasing precision, from `longdouble` to `double`. Table 1 presents these values. As it can be observed, for the lower t_κ the effect is negligible, but upon increase it becomes much more pronounced.

Table 1: Comparison of RMSE (in cm^{-1}) of CP-FBR and POS-FBR for different order of fitting polynomial and different precision.

Chebyshev order	5	6	7	8	9	10
CP-FBR	143.96	42.01	15.80	7.34	3.14	0.94
POS-FBR(<code>longdouble</code>)	143.96	42.01	15.80	8.98	22.88	21480.57
POS-FBR(<code>double</code>)	143.96	42.01	15.81	8.02	60.92	39182.27

Conclusions

Compact, low-rank, representations of (high-dimensional) operators are essential for efficient grid-based simulations. This also holds for the efficient evaluation of Deep Learning models.

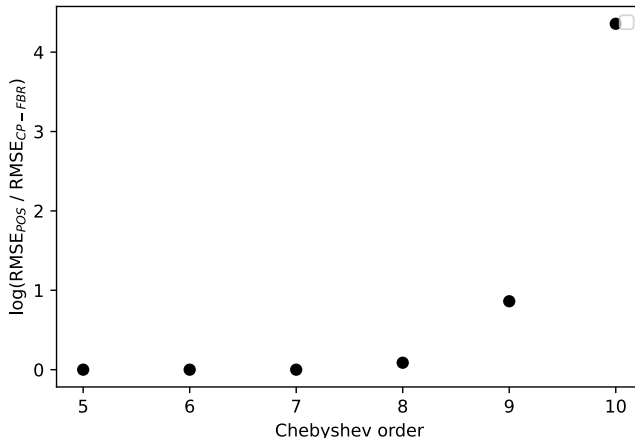


Figure 3: Logarithm of ratio of RMSE values of CP-FBR and its corresponding POS-FBR as a function of order of fitting polynomials used to obtain CP-FBR. The figure illustrates that with increased number of operations the rms is significantly less preserved during the transformation

Achieving a low-rank expression through tensor decomposition (refit) is limited either by the size of the tensor (Tucker case) or by the actual algebraic process (CP case). Refitting processes (aka tensor decomposition) lead to expressions which are bound to a specific grid. This is problematic. For instance, one may need to change the distribution of grid points (DVR) or their density in the course of the simulation (for instance in order to numerically converge the results). Hence the interest in the possibility of achieving analytical (symbolic) low-rank expressions. In this work, we have explored the different possibilities within the context of Tucker and CP forms. We have shown that numerical issues (finite-precision arithmetic) prevent the transformation of SOP- and CP-FBR forms into the seemingly more convenient POS-FBR. The latter allows the direct generation of an analytical, and more precisely, symbolic expression for the Hamiltonian operator, the so-called `operator` file in MCTDH. The POS-FBR form would result in a sum-of-products of simple polynomials (e.g. Chebyshev). Unfortunately, the SOP-POS transformation involves a large number of operations leading to a major loss of precision. This, ultimately, renders the whole process impractical. The underlying reason is that polynomial evaluation, more precisely the underlying arithmetic, is particularly susceptible to such loss of precision. The problem gets aggravated for poly-

nomials of high degree, which, provided the range of our input values, would be around 10. Finally, we would like to point out that, in the case of MCTDH, it is indeed feasible to write a CP-FBR expression in the form of a sum of (user defined) 1D-functions owing to its low-rank character. This is not the case for Tucker-FBR unless the core elements are pruned.² In any case, one could do this without transformation to POS.

Availability of data

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgements

NN is very grateful for its Ph.D. funding from the University Paris-Saclay. The authors are very thankful to J.-Y. Bazzara and A. Borissov (ISMO) for their computer support.

References

- (1) Liu, X.; Parhi, K. K. Tensor Decomposition for Model Reduction in Neural Networks: A Review. *IEEE Circuits and Systems Magazine* **2023**,
- (2) Panadés-Barrueta, R. L.; Peláez, D. Low-rank sum-of-products finite-basis-representation (SOP-FBR) of potential energy surfaces. *J. Chem. Phys.* **2020**, *153*, 234110.
- (3) Jäckle, A.; Meyer, H.-D. Time-dependent calculation of reactive flux employing complex absorbing potentials: General aspects and application within MCTDH. *J. Chem. Phys.* **1996**, *105*, 6778.

- (4) Worth, G. A.; Beck, M. H.; Jäckle, A.; Meyer, H.-D. The MCTDH Package, H.-D. Meyer, Version 8.4.12. See <http://mctdh.uni-hd.de/>, 2016.
- (5) Worth, G. A.; Giri, K.; Richings, G. W.; Burghardt, I.; Beck, M. H.; Jäckle, A.; Meyer, H.-D. The QUANTICS Package, Version 1.1, See: <https://www2.chem.ucl.ac.uk/worthgrp/quantics/doc/quantics/citation.html/>, 2015.
- (6) Mamonov, A. V.; Olshanskii, M. A. Interpolatory tensorial reduced order models for parametric dynamical systems. *Computer Methods in Applied Mechanics and Engineering* **2022**, *397*, 115122.
- (7) Jäckle, A.; Meyer, H.-D. Product representation of potential energy surfaces. *J. Chem. Phys.* **1996**, *104*, 7974.
- (8) Peláez, D.; Meyer, H.-D. The multigrid POTFIT (MGPF) method: Grid representations of potentials for quantum dynamics of large systems. *J. Chem. Phys.* **2013**, *138*, 014108.
- (9) Otto, F. Multi-Layer Potfit: An accurate potential representation for efficient high-dimensional quantum dynamics. *J. Chem. Phys.* **2014**, *140*, 014106.
- (10) Schröder, M.; Meyer, H.-D. Transforming high-dimensional potential energy surfaces into sum-of-products form using Monte Carlo methods. *J. Chem. Phys.* **2017**, *147*, 064105.
- (11) Schröder, M. Transforming high-dimensional potential energy surfaces into a canonical polyadic decomposition using Monte Carlo methods. *J. Chem. Phys.* **2020**, *152*, 024108.
- (12) Richter, F.; Hochlaf, M.; Rosmus, P.; Gatti, F.; Meyer, H.-D. A study of mode-selective trans-cis isomerisation in HONO using ab initio methodology. *J. Chem. Phys.* **2004**, *120*, 1306–1317.

- (13) Nadoveza, N.; Panadés-Barrueta, R.; Shi, L.; Gatti, F.; Peláez, D. Analytical high-dimensional operators in Canonical Polyadic Finite Basis Representation (CP-FBR). *J. Chem. Phys.* **2023**, *158*, 114109.
- (14) Kolda, T. G.; Bader, B. W. Tensor Decompositions and Applications. *SIAM Review* **2009**, *51*, 455–500.
- (15) Panadés-Barrueta, R. L.; Martínez-Núñez, E.; Peláez, D. Specific Reaction Parameter Multigrid POTFIT (SRP-MGPF): Automatic Generation of Sum-of-Products Form Potential Energy Surfaces for Quantum Dynamical Calculations. *Front. Chem.* **2019**, *7*, 576.
- (16) Martínez-Núñez, E.; Barnes, G. L.; Glowacki, D.; Kopec, S.; Peláez, D.; Rodríguez, A.; Rodríguez-Fernández, R.; Shannon, R. J.; J. J. P. Stewart, P. T.; Vázquez, S. A. AutoMeKin2021: An open-source program for automated reaction discovery. *J. Comput. Chem.* **2021**, 1–13.
- (17) Kopec, S.; Martínez-Núñez, E.; Soto, J.; Peláez, D. vdW-TSSCDS - An automated and global procedure for the computation of stationary points on intermolecular potential energy surfaces. *Int. J. Quant. Chem.* **2019**, e26008.
- (18) Graillat, S.; Langlois, P.; Louvet, N. Algorithms for accurate, validated and fast polynomial evaluation. *Japan Journal of Industrial and Applied Mathematics* **2009**, *26*, 191–214.
- (19) Bailey, D. H.; Barrio, R.; Borwein, J. M. High-precision computation: Mathematical physics and dynamics. *Applied Mathematics and Computation* **2012**, *218*, 10106–10121.
- (20) Jiang, H.; Li, S.; Cheng, L.; Su, F. Accurate evaluation of a polynomial and its derivative in Bernstein form. *Computers & Mathematics with Applications* **2010**, *60*, 744–755.

- (21) Jiang, H.; Barrio, R.; Li, H.; Liao, X.; Cheng, L.; Su, F. Accurate evaluation of a polynomial in Chebyshev form. *Applied Mathematics and Computation* **2011**, *217*, 9702–9716.
- (22) Polyansky, O. L.; Jensen, P.; Tennyson, J. The potential energy surface of H₂ ¹⁶O. *J. Chem. Phys.* **1996**, *105*, 6490–6497, PES, water.
- (23) Stewart, J. J. P. MOPAC2016, Stewart Computational Chemistry, Colorado Springs, CO, USA. [HTTP://OpenMOPAC.net](http://OpenMOPAC.net) **2016**,
- (24) Richard Shewchuk, J. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete & Computational Geometry* **1997**, *18*, 305–363.