PHYSICAL REVIEW ACCELERATORS AND BEAMS **25,** 124601 (2022)

Multivariant spatial and geometric interpolation of field maps through model order reduction

X. Du[*] and L. Groening

GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt D-64291, Germany

(Received 15 February 2022; accepted 17 November 2022; published 7 December 2022)

Complex three-dimensional field maps can be efficiently denoised and compressed in data volume by high order single value decomposition. Extension of the method to inter/extrapolation and determination of basis function representation is reported here. Inter/extrapolation of denoised and compressed maps is applied to spatial coordinates and especially to the parameters defining the geometry of devices creating such maps. It significantly increases the efficiency of the provision of compact and noise-free maps combined with high resolution. The method is illustrated by creating corresponding maps for rf cells of radio-frequency quadrupoles.

DOI: 10.1103/PhysRevAccelBeams.25.124601

I. INTRODUCTION

The first conceptual design of beam dynamics of modern accelerators is done through linear beam optics. It provides basic parameter lists and quantifies the orders of magnitude of an accelerator project. Further steps involve nonlinear effects on the beam. These are from beam self-forces or from external fields imposed by rf cavities, electrostatic optics elements, or by magnets. Such fields determine the achievable beam quality as well as loss scenarios. Precise knowledge of these issues is essential for further planning of the overall project resources as well as the range of applications of the respective machine. This applies especially to circular machines. From first principles, nonlinearities from external fields can be quantified to any desired precision through detailed design and machining of the devices which create them, in combination with high-resolution three-dimensional (3D) field map calculations (sometimes referred to as “transfer map” calculations).

However, the required resources increase with the aimed precision. In practice, both are to be balanced with each other. Field map simulations are prone to numerical noise and require large data volumes. Unfortunately, the impact of noise on the high order field components increases with the order of the latter. A distinction of physical content from noise is difficult if not impossible especially at high orders.

The issue of noise reduction and compression of essential data volume has been addressed recently through high order single value decomposition (HOSVD) of 3D field maps [1] (“high order” is also referred to as “high rank” in related literature; here it does not refer to convergence order). Using the example of 3D electric field maps of a drift tube linac (DTL), the data volume was reduced by 4 orders of magnitude with simultaneous elimination of noise from numerics and/or finite mesh size. Such techniques can be classified as model order reduction.

There is further potential for the drastic reduction of efforts to provide noise-free and precise field maps. First, different classes of beam optical elements create different classes of field maps as for instance the classes of octupole magnets, rf cells of a DTL, a radio-frequency quadrupole (RFQ), an rf antenna, or rf coupling loop. The element (map) of each class is defined by a set of parameters. However, all elements of the same class have many properties that are intrinsic to this special class. Once some few “probe” elements of one class have been fully characterized by a field map, these intrinsic properties of this special class are known. Other elements can be characterized to the same detail with much less efforts since intrinsic properties are not redetermined but readapted. Determination of intrinsic class properties requires the bulk of resources as CPU time and storage space. The latter is not needed to reconfirm intrinsic class properties.

Another contribution to reduce efforts for provision of compact and noise-free field maps is through further augmenting the efficiency of interpolation. Particle-in-cell (PIC) simulations include huge amounts of trilinear and tricubic interpolations. Their efficiency dominates the simulation speed. Most PIC codes use conventional interpolation methods, i.e., picking a few surrounding mesh nodes from linearized data. Some codes may involve

---

[*] x.du@gsi.de

special mathematical algorithms to perform interpolations without revealing them. Finally, efficiency can be augmented by high precision approximation of the maps through analytic expressions.

Codes for the geometry design of RFQs or DTLs often include algorithms that generate denoised field maps from geometric parameters. The most known ones are PARMTEQ [2] and LIDOS [3]. These codes gradually vary these parameters such that they match the actual beam energy and emittance at the respective rf cell. A complete cavity is accordingly modeled by the sequence of all cells. RFQ design codes increasingly base on modern coding methods allowing for more detailed physics simulations. The bunch of macroparticles is tracked through the cells and statistics of the beam (size, energy, emittance, particle loss, etc.) is recorded for later analysis. This process is iterated until an acceptable solution has been found. Accordingly, the geometric parameters of the RFQ electrodes are determined to produce proper field maps. The design of other types of elements follows similar processes. During overall optimization of such processes, fast availability of high quality field maps for arbitrary cell parameters is required.

In the case of RFQs, the on-axis potential can be expanded as infinite Fourier-Bessel series in cylindrical coordinates. Historically, the first RFQs were generated using the two-term potential to calculate particle trajectories and these RFQs worked well. However, performances partially did not fully meet expectations w.r.t. beam transmission and output beam quality. A major contributor was the truncation of the field equations after two terms. Developers added six more terms (and later two more for off-axis trajectories). The eight-term potential field description is widely used and trusted for the case of sinusoidal vane modulation [4].

Solving the Laplace equation takes into account the real geometry of the vanes, which is determined by several parameters. However, this method delivers inaccurate descriptions of the fields near the boundary surfaces. Surface-based fitting methods have been studied as solutions to obtain accurate transfer maps and are successfully applied to magnets [5–8] and rf cells [9]. The error close to boundaries and effects from numerical noise can be overcome by fitting field data on a boundary surface far from the beam axis and continuing inwards using Maxwell's equations. The coefficients of expansion can be determined by the projection of the numerical solution. Codes use a number of predefined lookup tables for predefined geometries. Coefficient values for any rf cell are calculated by interpolation [10].

Fitting of the cell geometry by the least-square method results into an accuracy of several percent, from which in turn the accuracy of beam dynamics simulations is suffering [11,12]. The accuracy is especially poor outside the cylinder of $\sqrt{x^2 + y^2} < a$, which may still be populated by considerable fractions of the beam. One alternative is to simulate numerically the fields at each iteration, being time consuming and needing a lot of memory to store the mesh for several cells.

Similar needs for dealing with high order, aligned, and prestacked data occur in other research fields such as plasma modeling or seismic modeling for instance. HOSVD has been developed as a mathematical tool to perform operations with high order tensors that represent these data. It is applied in computer graphics [13], machine learning [14], scientific computing [15], and signal processing [16]. The technique uses multilinear generalization of matrix SVD to high order tensors, and it provides for adequate preservation of the data essentials through the representation of a tensor through appropriate bases [17].

This paper is organized as follows: in the next section, HOSVD is applied to a set of complex RFQ potential maps together with illustrating the proposed multivariant interpolation and fitting algorithm. The latter is applied to decomposed and compressed data. The third section is on systematic benchmarking of these results to field maps obtained from direct simulations w.r.t. accuracy and efficiency. The paper closes with summarizing conclusions. Mathematical details on the method of inter/extrapolation and fitting are given within the appendixes.

II. DEMONSTRATION OF THE METHOD WITH COMPLEX RFQ POTENTIAL MAP

This section illustrates the complete process of HOSVD and subsequent algorithms using an explicit example. Notations being used throughout the paper are borrowed to a large extent from tensor algebra and will be introduced briefly at first use. The first subsection introduces a set of complex potential maps as dummy data. The second subsection applies HOSVD, compression, and noise reduction (trimming) to the data, applying a process described in detail in [1] including the discussion of the related consequences. Afterward, multivariate interpolation and fitting algorithms are presented which use decomposed and trimmed data. The basic features of interpolation and extrapolation are presented in Appendix A.

A. Field map data of RFQ cells

This section introduces a set of complex potential maps of an RFQ. It serves as preparation of benchmarking of algorithms for inter/extrapolation with those for obtaining an approximate basis function expression of the map. The algorithms are also applicable to other types of transfer maps as quadrupole magnets and DTLs, as long as their characteristics are determined by a finite number of geometric parameters. The RFQ cell is chosen since its structure is more complex w.r.t. a DTL cell. Additionally, the field outside the aperture needs partially to be considered within the calculations.

RFQ cavities are commonly used at hadron linear accelerators. Various geometries of electrodes have been developed to meet a wide range of beam dynamics specifications. Beam transmission through an RFQ is calculated with dedicated simulation codes and precise knowledge of the 3D potential distribution between the four vanes is essential. The quadrupolar electric field varies as $\cos(\omega t)$ and this time dependence is not further considered in this paper.

Initially, the electric potential inside of an RFQ cell is simulated using CST-MWS [18]. In exploiting symmetry, just one-quarter of the RFQ cell needs to be simulated. Since the aperture is much smaller than the rf wavelength, electrostatic approximation is applied. Although the cell length and other geometry parameters change gradually along the RFQ, the field analysis is done assuming identical cells along a fully periodic cavity. The potential on the vane tips is set to ±1 V.

As shown in Fig. 1, the data volume is represented by the blue box with a being fixed to 1.0 mm as minimum aperture radius of the cell. The simulated volume is one quarter within $x = [0, a]$, $y = [0, a]$, and $z = [0, l]$. Here x, y, and z refer to the horizontal, vertical, and longitudinal coordinates, respectively, and l to the rf cell length. In general, this volume includes the bulk of beam particles, and usually, there is a difference of a few percent in transmission efficiency depending on whether particles outside that volume are being considered lost [19].

The simulated field map is projected onto a regular rectangular grid using appropriate interpolation. During simulation and projection, unavoidable discretization errors occur from the finite resolution of the mesh grids. Those errors will be treated later through HOSVD. For each cell, the geometry is completely determined by the three parameters m, r, and l. The field properties have been calculated for many rf cells within a 3D array of m, r, and l. Six sweep steps have been applied for each parameter, thus $6 \times 6 \times 6$ rf cells have been simulated in total. To each rf cell $201 \times 41 \times 41$ (long. × hor. × ver.), mesh nodes have been assigned. The entire potential map set is stored as a sixth-order tensor $A \in \mathbb{R}^{6\times6\times6\times201\times41\times41}$.

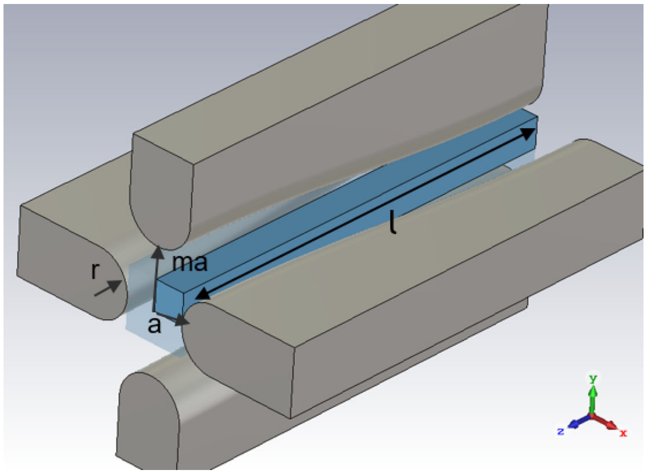


FIG. 1. A typical RFQ cell comprises one-half of an oscillation of the sinusoidal vane tip. a: minimum distance from the axis, ma: maximum distance from the axis, m: modulation factor, l: rf cell length, r: vane tip radius.

$\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ specifies the dimension of A and I_n is the dimension of the n-mode vector in A. $A[i_1, i_2, i_3, i_4, i_5, i_6]$ is the scalar potential value being stored at data point $(i_1, i_2, i_3, i_4, i_5, i_6)$, thus $i_n \le I_n$. Accordingly, a matrix $U \in \mathbb{R}^{2\times3}$ of two rows and three columns is a second-order tensor and $U[i, j]$ is the jth element in the ith row. Each row is a two-mode vector with three dimensions $(I_2 = 3)$ and each column is a one-mode vector with two dimensions $(I_1 = 2)$. The concept of an n-mode vector is explicitly explained in [17].

The numbers of entries in A along each mode are m step, r step, l step, z step, y step, and x step. Accordingly, each scalar potential value is uniquely labeled by six numbers as listed in Table I.

B. HOSVD and trimming

Applying the standard HOSVD procedure [17] onto the sixth-order tensor A delivers its decomposed form

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \times_4 U^{(4)} \times_5 U^{(5)} \times_6 U^{(6)}, \qquad (1)$$

where $S \in \mathbb{R}^{6\times6\times6\times201\times41\times41}$ is called the core tensor or coefficient tensor. $U^{(n)} \in \mathbb{R}^{I_n\times I_n}$ includes singular vectors, i.e., a set of horizontal vectors being stacked vertically. $U^{(n)}$ satisfies $U^{(n)-1} = U^{(n)T}$, meaning that these singular vectors are normalized and orthogonal to each other. The operator $\times_n$, refers to the n-mode product, being the key to HOSVD explicitly explained in [17] and briefly introduced in [1]. The explicit expression for each element in A is an expansion of mutually orthogonal first-order tensors

$$A[k_1, k_2, \ldots, k_6] = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_6=1}^{I_6} S[i_1, i_2, \ldots, i_6] \times U^{(1)}[i_1, k_1] U^{(2)}[i_2, k_2] \ldots U^{(6)}[i_N, k_6]. \qquad (2)$$

TABLE I. Mesh grid of the discrete 6D potential. x, y, z: spatial positions, r, l, m: geometry parameters of rf cell vanes.

Label	Parameter	Mesh nodes	Range
m	Modulation factor	$I_1 = 6$	1.5–2.5
r	Vane tip radius	$I_2 = 6$	0.6–1.6 mm
l	Cell length	$I_3 = 6$	5–20 mm
z	z mesh nodes	$I_4 = 201$	0–l
y	y mesh nodes	$I_5 = 41$	0–3 mm
x	x mesh nodes	$I_6 = 41$	0–3 mm

The method also delivers singular values for each singular vector thus reflecting the vectors' weights. $\lambda^{(n)} \in \mathbb{R}^{I_n}$ denotes the array of singular values for $U^{(n)}$. The singular vectors in $U^{(n)}$ are sorted according to their singular values, such that the most relevant singular vectors can be identified for further use.

For the example of an RFQ cell, the highest singular values for each n-mode obtained after decomposition are

$$\begin{aligned}
\lambda^{(1)} &= (46.180, 1.2870, 0.0006, 0.0002, \ldots) \\
\lambda^{(2)} &= (47.411, 0.0554, 0.0002, 0.0001, \ldots) \\
\lambda^{(3)} &= (47.425, 0.0371, 0.0002, 0.0001, \ldots) \\
\lambda^{(4)} &= (1273.8, 316.10, 0.1255, 0.0896, \ldots) \\
\lambda^{(5)} &= (291.95, 32.394, 0.0135, 0.0001, \ldots) \\
\lambda^{(6)} &= (291.95, 32.394, 0.0135, 0.0001, \ldots). \qquad (3)
\end{aligned}$$

First, denoising and compression are applied to the decomposed tensor A of Eq. (1), hence elimination of redundancy, unphysical fluctuations, and noise. The algorithm neglects low-valued singular vectors (singular vectors corresponding to low singular values) by preserving just a few high-valued singular vectors. This process, being one case of model order reduction, is called trimming of S and it is explained in detail in [1], where it has been successfully applied to a set of electric 3D field maps of DTL cells. The decomposed tensor described by Eq. (1) can be trimmed to

$$A \approx s \times_1 u^{(1)} \times_2 u^{(2)} \times_3 u^{(3)} \times_4 u^{(4)} \times_5 u^{(5)} \times_6 u^{(6)}, \qquad (4)$$

where $u^{(n)} \in \mathbb{R}^{j_n \times I_n}$ and $j_n < I_n$ is the chosen threshold for the n mode. $s \in \mathbb{R}^{j_1 \times j_2 \times \ldots \times j_6}$ is the preserved smaller core tensor. Dropping low single value data will reduce the data volume from $\prod_{n=1}^{6} I_n$ to $\prod_{n=1}^{6} j_n$ and increase the data quality through reduction of redundancy, unphysical fluctuations, and noise. The values of j_n depend on the threshold. Section II of [1] describes the method of proper threshold determination. In most cases, less than 10% of the singular vectors are needed to preserve the physical content of the data, which in turn implies that for an Nth-order tensor of about $(0.1)^N$ of the data is preserved and others are considered as redundancy, unphysical noise, or high frequency terms.

In general, it is not straightforward to distinguish between the latter, and literature discussing this issue is sparse. Their only common feature is representation by low-valued singular vectors. Redundancy has low singular values as the variation of data distributions hardly manifests along the new axis defined by these singular vectors. Noise delivers low singular values due to its randomness. Finally, high frequency terms of the data intrinsically result in low singular values. Theoretical analysis is missing for the time being and in practical applications, these three quantities are usually discarded together. The numerical experiment reported in [1] confirms that for a typical field map, properly trimmed data have much less deviation from perfect analytical solution w.r.t. untrimmed data.

A good qualitative indicator for a noise-dominated singular vector is its smoothness. Figure 2 plots singular vectors $\lambda_1^{(4)}$ to $\lambda_5^{(4)}$ of $U^{(4)}$ as functions of the z step. The first four are very smooth while the fifth features obvious random fluctuations. Its singular value is more than 6 orders of magnitude smaller than the singular value of $\lambda_1^{(4)}$ and 2 orders of magnitude smaller than the one of $\lambda_4^{(4)}$. Strong fluctuations together with very small relative singular values identify noise-dominated singular vectors. Instead, smooth singular vectors with small relative singular values identify vectors corresponding to physical higher order terms.

The thresholds C_n have been chosen such that the singular values of the omitted singular vectors are at least 4 orders of magnitude lower than the highest one, assuring that the physical data content is preserved sufficiently well. Accordingly, $j_1 = j_2 = j_3 = 2, j_4 = j_5 = j_6 = 3,$ is trimmed to a much smaller tensor $s \in \mathbb{R}^{2\times2\times2\times3\times3\times3}$ by deleting all elements whose entries are out of range $1 \leq i_n \leq j_n$. $U^{(n)}$ is trimmed to $u^{(n)}$ by deleting all rows below j_n^{th}, thus $u^{(1)} \in \mathbb{R}^{2\times6}$, $u^{(2)} \in \mathbb{R}^{2\times6}$, $u^{(3)} \in \mathbb{R}^{2\times6}$, $u^{(4)} \in \mathbb{R}^{3\times201}$, $u^{(5)} \in \mathbb{R}^{3\times41}$, and $u^{(6)} \in \mathbb{R}^{3\times41}$. As an illustration, Fig. 3 shows the distribution of $U^{(2)}$ and how it is trimmed to $u^{(2)}$.

Binary storage of the original data A takes 240 MB of storage volume. Instead, storage of the compressed and denoised data (s and $u^{(n)}$) requires just 10 kB. Furthermore, the algorithm significantly reduces the amount of noise and preserves a smooth distribution with less than 1% of deviation from the original data. Hence, compression and noise reduction have been well achieved. Benchmarking is skipped for this instance since the method already has been successfully benchmarked in [1]. The computing process

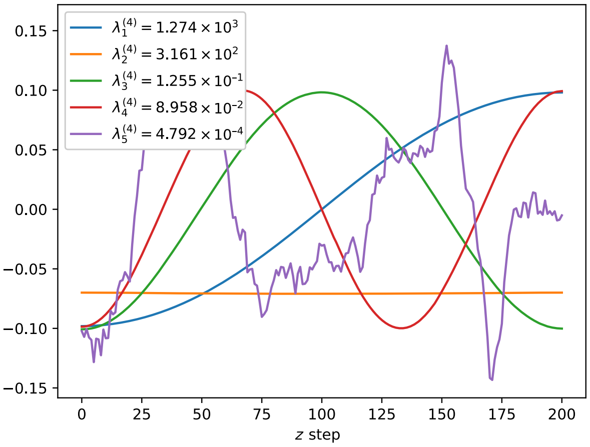


FIG. 2. Singular vectors $\lambda_1^{(4)}$ to $\lambda_5^{(4)}$ of $U^{(4)}$ as functions of the z step.

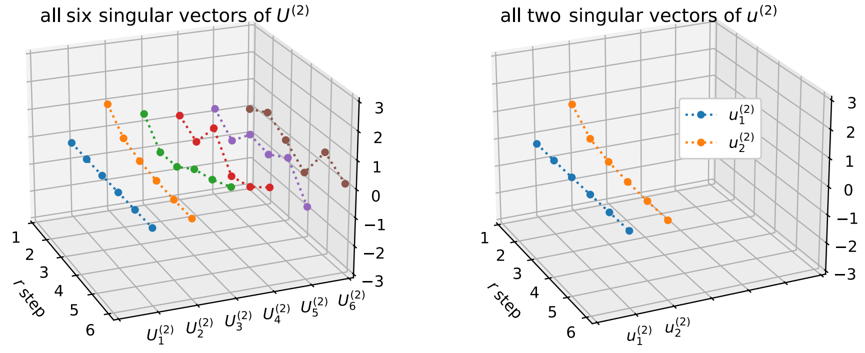


FIG. 3. Example for a plot of $U^{(2)}$ (left) and $u^{(2)}$ (right) as functions of the r step. The vertical axis displays the real values of singular vector elements. According to Eq. (3), $U_3^{(2)} \sim U_6^{(2)}$ correspond to singular values that are at least 4 orders of magnitude lower than the largest singular value occurring in the two-mode decomposition.

has been coded with PYTHON and its source code is shared on GitHub [20]. An extension of the algorithm will be introduced in the following section.

C. Further operations on singular vectors: Inter/extrapolation and curve fitting

Field maps, whether discrete or analytical, are used to evaluate the field strength at the location of the macroparticle. Evaluation of field values at arbitrary coordinates involves multivariate approximations. In general, approximations can be provided through interpolation or from fitting functions. The fitting function can be expressed as a product of tensors being sums of products of univariate functions [21]. Fortunately, Eq. (4) already presents the data in form of a tensor product. The high-valued singular vectors being preserved are essentially smooth and degenerate 1D distributions. Their smoothness is inherited from the intrinsic properties of the original data.

The good properties of preserved singular vectors in $u^{(n)}$ suggest performing 1D interpolation on them or determination of 1D fitting functions for them. The expectation is that once $u^{(n)}$ is inter/extrapolated for all n-modes, multivariate interpolation can be achieved. Multivariate fitting will then be achieved accordingly. Figure 4 represents an illustration using $u^{(2)}$ as an example. Both 1D inter/extrapolation and 1D curve fitting are well studied, thus the 1D module in PYTHON is applied for establishing methods to determine potential values at any 6D coordinate.

1. Interpolation

This section illustrates multivariate inter/extrapolation using the explicit example treated in the previous section. The data remain decomposed and the inter/extrapolation strategy is applied to this decomposed and trimmed data. Considering a point coordinate $(m_0, r_0, l_0, z_0, y_0, x_0)$, the scalar potential value from multivariate interpolation on the point is written as

$$f(m_0, r_0, l_0, z_0, y_0, x_0) \approx s \times_1 \hat{u}^{(1)} \times_2 \hat{u}^{(2)} \times_3 \hat{u}^{(3)} \times_4 \hat{u}^{(4)} \times_5 \hat{u}^{(5)} \times_6 \hat{u}^{(6)}, \quad (5)$$

where $\hat{u}^{(n)} \in \mathbb{R}^{j_n \times 1}$ is the result of several (j_n) 1D interpolations on the base set $u^{(n)}$ for the nth parameter. The left plot of Fig. 4 visualizes this process. The 1D interpolation method is not strictly defined and one may choose linear, polynomial, or spline functions. Equation (5) decomposes

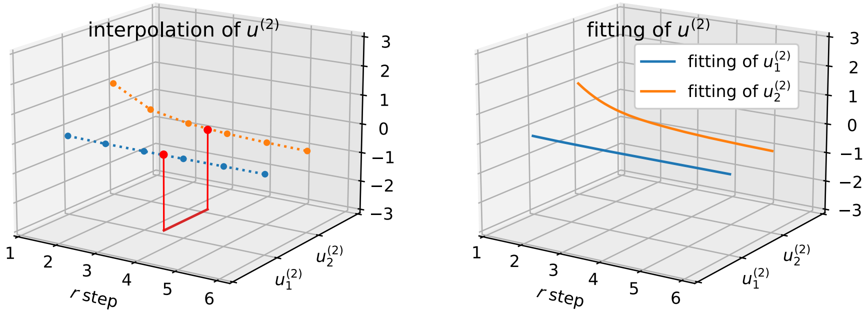


FIG. 4. Interpolation and fitting of preserved singular vectors of $u^{(2)}$. Left: interpolation to r step $= 3.5$ or $r = 1.1$ mm. Right: replacement of singular vectors by smooth and best-fitting functions.

any multivariate inter/extrapolation problem into several $(\sum_{n=1}^{N} j_n)$ 1D inter/extrapolations.

The interpolation procedure may be applied to x, y, and z to perform the spatial interpolation. However, applying the algorithm to geometric parameters instead will deliver the particular 3D potential map at any value of m, r, and l within the range of the arrays. For realistic cell-by-cell PIC simulations, for each iteration, just one 3D field map of one single cell with specific geometric parameters is required. Accordingly, the interpolation process is divided into two steps.

The first step interpolates the geometric parameters m, r, and l to obtain a lower rank core tensor $s' \in \mathbb{R}^{3\times3\times3}$

$$s' = s \times_1 \hat{u}^{(1)} \times_2 \hat{u}^{(2)} \times_3 \hat{u}^{(3)}. \tag{6}$$

The 3D potential map $F \in \mathbb{R}^{201\times41\times41}$ for a cell with parameters m_0, r_0, and l_0 can be easily obtained by

$$F \approx s' \times_1 u^{(4)} \times_2 u^{(5)} \times_3 u^{(6)}. \tag{7}$$

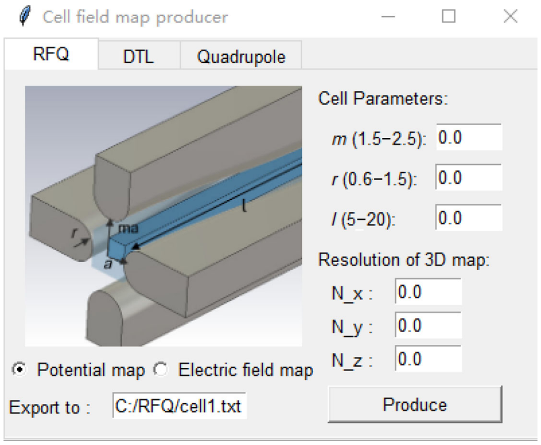
Equation (7) allows for generating high quality RFQ field maps from just 10 kB of data with a few lines of code. Field maps for other beam optics elements can be treated the same way.

It should be emphasized that the resolution of the output field file can be adjusted arbitrarily by just resampling $u^{(4)}$, $u^{(5)}$, and $u^{(6)}$. The method can be implemented into PIC simulation codes either as a module, or be developed into an independent, extremely light-weighted field map generation program. We have realized such a program with PYTHON as an example and Fig. 5 illustrates the interface. The source code for the algorithm and the program have been added to GitHub [20].

Within the second step, a large amount of spatial interpolations is performed by PIC simulations for the sequence of specified coordinates, where macroparticles are located. It is proposed to keep F in the decomposed form and to perform 3D spatial interpolation as

$$f(z_0, y_0, x_0) \approx s' \times_1 \hat{u}^{(4)} \times_2 \hat{u}^{(5)} \times_3 \hat{u}^{(6)}. \tag{8}$$

Multivariate inter/extrapolation has been well studied since it comprises a highly relevant mathematical application [22,23]. However, to the best of our knowledge, the presented method has not been widely used in PIC simulations as an alternative to the traditional methods that have been used for decades. The method's accuracy and efficiency will be evaluated in the following section.



Cell Parameters:

m (1.5–2.5): 0.0

r (0.6–1.5): 0.0

l (5–20): 0.0

Resolution of 3D map:

N_x : 0.0

N_y : 0.0

N_z : 0.0

● Potential map ○ Electric field map

Export to : C:/RFQ/cell1.txt

FIG. 5. Illustration of the executable program producing field maps for typical accelerator elements.

2. *Fitting*

The ultimate solution for inter/extrapolation is finding an expression that fits best the multivariate data points. As mentioned in the Introduction, efforts have been made to find basis representations for some beam optics elements such as multipole magnets and RFQ cells. The following describes the construction of basis function representations of the field, using the decomposed and compressed numerical field map.

Given that $u^{(n)}$ contains different but smooth singular vectors, it can be fit through some expansion of a few terms without any inconveniences from overfitting. The issue to be addressed is what kind of basis and how many terms are required. Let $P^{(n)}$ denote the t_n terms basis functions chosen for the fitting of $u^{(n)}$, then those high value singular vectors can be fitted by

$$u^{(n)} \approx K^{(n)} \hat{P}^{(n)}, \tag{9}$$

where $\hat{P}^{(n)} = P^{(n)}(X^{(n)}) \in \mathbb{R}^{t_n \times I_n}$ are t_n terms basis functions projected onto uniform mesh nodes at defined locations $X^{(n)} \in \mathbb{R}^{I_n}$. The relation $t_n \geq j_n$ has to be met since all singular vectors in $u^{(n)}$ are orthogonal. The coefficients $K^{(n)} \in \mathbb{R}^{j_n \times t_n}$ can be calculated with the least square method and the detailed procedure is presented in Appendix B. By plugging Eq. (9) into Eq. (4), the complete coefficient tensor $\bar{S} \in \mathbb{R}^{t_1\times t_2\times t_3\times t_4\times t_5\times t_6}$ reads

$$\bar{S} = s \times_1 K^{(1)} \times_2 K^{(2)} \times_3 K^{(3)} \times_4 K^{(4)} \times_5 K^{(5)} \times_6 K^{(6)}. \tag{10}$$

Afterward, the basis function representation of the fitting $\bar{A}(m, r, l, z, y, x)$ can be found as a tensor product by simply replacing $\hat{P}^{(n)}$ in Eq. (9) with their basis function version $P^{(n)}$

$$\bar{A} = \bar{S} \times_1 P^{(1)} \times_2 P^{(2)} \times_3 P^{(3)} \times_4 P^{(4)} \times_5 P^{(5)} \times_6 P^{(6)}. \tag{11}$$

$\bar{A}$ is naturally stacked as a tensor product being convenient for all types of linear operations. By continuously increasing the mesh nodes of $\hat{P}^{(n)}$, such that $I_n \to \infty$ for all n

TABLE II. Transformation matrix for each set of basis.

$K^{(1)}$	-4.08×10^{-1}	-6.33×10^{-2}	1.26×10^{-2}	-1.70×10^{-3}
	-5.22×10^{-2}	5.77×10^{-1}	-1.11×10^{-1}	1.57×10^{-2}
$K^{(2)}$	-4.10×10^{0}	-1.96×10^{-2}	1.34×10^{-2}	-6.05×10^{-3}
	-4.27×10^{-1}	-4.53×10^{-1}	3.02×10^{-1}	-1.35×10^{-1}
$K^{(3)}$	-4.08×10^{-1}	-3.21×10^{-2}	4.94×10^{-3}	-6.11×10^{-4}
	-1.43×10^{-2}	5.99×10^{-1}	-4.00×10^{-2}	6.06×10^{-3}
$K^{(4)}$	3.07×10^{-7}	1.21×10^{-1}	-2.95×10^{-7}	-2.47×10^{-2}
	-7.05×10^{-2}	5.33×10^{-7}	7.48×10^{-4}	-1.94×10^{-8}
	4.14×10^{-4}	0.98×10^{-5}	-1.50×10^{-1}	5.25×10^{-4}
$K^{(5)}$ or $K^{(6)}$	-1.56×10^{-1}	3.55×10^{-4}	-7.52×10^{-4}	-5.84×10^{-4}
	-1.98×10^{-3}	2.56×10^{-1}	8.27×10^{-2}	-1.50×10^{-3}
	7.71×10^{-3}	7.10×10^{-2}	-2.90×10^{-1}	-1.68×10^{-1}

modes, the discrete basis will turn into basis function representation as well as the recovered tensor.

There is no special requirement for the choice of $P^{(n)}$. Depending on the specific data, the fitting accuracy may improve or the required terms may be reduced by choosing a suitable basis set of functions according to its characteristics. For example, the Fourier series is suitable for data with periodic features. If polynomials are chosen for all $P^{(n)}$, the mathematical interpretation is identical to tensor product splines, box splines [24], or multivariate splines [25] in terms of multivariate polynomial spline approximation. The fitted multivariate function in this paper applies to the full data range, instead of limited regions, being determined by individual local polynomials.

The algorithm is applied to the decomposed and compressed data being produced at the end of Sec. II B. In that case, $P^{(n)}$ are Legendre polynomials which can be expressed as basis transformation from Maclaurin series

$$\begin{bmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ P_3(x) \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & \vdots \\ -1/2 & 0 & 3/2 & 0 & \\ 0 & -3/2 & 0 & 5/2 & \\ & \cdots & & & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \\ \vdots \end{bmatrix}, \tag{12}$$

with different terms, where $P_i^{(n)}$ corresponds to remapping of $P_{i-1}(x)$ from range $-1 < x < 1$ to the range of the nth variable. The chosen terms for this example are $t_1 = t_3 = 3$, $t_2 = t_4 = t_5 = t_6 = 4$, although a different basis could have been chosen for each dimension. For example, along any RFQ cell structure, the potential distribution along the z axis is periodic, thus a Fourier expansion should be the most promising to be used as $P^{(4)}$. On the other hand, for other cases, the use of Fourier series representations might impose artificial periodicity. The coefficient matrix calculated from Eqs. (B1) and (12) is listed in Table II.

t_n is chosen such that 1D fitting of $u^{(n)}$ can be calculated already in advance up to the desired accuracy. In order to achieve multivariant fitting with an accuracy of better than 1%, the omitted columns within $\lambda^{(n)} K^{(n)}$ are at least 3 orders of magnitude lower than the respective largest number, e.g., the gray values within Table II are omitted, meaning $t_1 = 3$ and $t_3 = 3$. For the other dimensions $t_2 = t_4 = t_5 = t_6 = 4$ and $K^{(5)} = K^{(6)}$, caused by the symmetry relating x and y, i.e., rotation of the RFQ cell by 90° around the z axis followed by swapping x and y preserves the cell geometry.

It is emphasized that obtaining 1% of accuracy requires significantly more terms of Legendre polynomials w.r.t. the number of singular vectors for all n modes. Singular vectors are revealed from the original data and form an orthogonal set of base vectors, unlike the terms of the Legendre polynomial. Each vector k models the most dominant contribution to the residual deviation left by the summed models of vectors $i < k$. Hence, they form a much better base for the data being considered.

Since all $K^{(n)}$ are known, the potential maps can be expressed by Eq. (11) through the new coefficient tensor $\bar{S} \in \mathbb{R}^{3\times4\times3\times4\times4\times4}$. s together with $K^{(n)}$ comprise 272 float numbers being the coefficients of the tensor product of six sets of Legendre polynomials. A function of all six variants that models the potential is given at once. The method is independent of the cell type and can be applied to any class of maps mentioned in the Introduction. For the special case of RFQs, trapezoidal modulation has been widely used recently to improve the shunt impedance, though its analytical potential formula remains to be determined [26]. Given that the method described here is not limited to Fourier-Bessel series and terms, the method is also applicable to RFQ cells with trapezoidal modulation.

With chosen t_n, each of the smooth singular vectors can be fitted by Legendre polynomials of up to four terms. For better understanding, the final expression for the used data set can be also expressed through a Maclaurin series

$$\bar{A}(m,r,l,z,y,x) = \sum_{i=0}^{2}\sum_{j=0}^{3}\sum_{k=0}^{2}\sum_{o=0}^{3}\sum_{p=0}^{3}\sum_{q=0}^{3} C_{ijkopq} m^i l^j r^k z^o y^p x^q. \quad (13)$$

The coefficients C_{ijkopq} can be calculated from $\bar{S}$ by the transformation matrix between the Legendre polynomial and the Maclaurin series, which is stated in Eq. (12).

The peak electric surface field is a crucial rf parameter to be well considered during the design process of any rf cavity and especially of RFQs. Each simulation of single cells provides this scalar together with the 3D potential map. The database can be established as a third-order tensor $E_{\mathrm{peak}} \in \mathbb{R}^{6\times6\times6}$. For further use, it can be fitted by a function $\bar{E}_{\mathrm{peak}}(m,r,l)$, which may provide the highest surface field together with the 3D potential value according to the parameters m, r, l. This convenient capability of the method will enormously increase the efficiency of the design process. Unlike well-studied surface-based fitting methods, the present algorithm does not enforce the fitted map to meet the Laplace equation. Original field maps from finite element method (FEM) simulations intrinsically possess this property, such that the fitting function will satisfy the Laplace equation sufficiently well.

The algorithm paves an alternative path to find a basis function description of the field. Hence, this approach is quite distinguished from the methods mentioned in the Introduction. A considerable advantage rising from Eq. (11) is that performing any linear operation to the base $P^{(n)}$ is equal to performing it to the distribution $\bar{A}$. Commonly used linear operators include inter/extrapolation, integration, and differentiation.

For instance, the partial differential of a multivariate distribution with decomposed form can be directly transformed into its gradients with minor effort. This advantage is based on an important property of linear operators: if a function is expressed as a tensor product of base functions, then the linear operator applied to these functions is equivalent to applying it to its base functions. Here for instance, the electric field components $E_x/E_y/E_z$ are easily obtained by replacing $P^{(6)}/P^{(5)}/P^{(4)}$ with their derivatives. This is straightforward through representation of the potential map $\bar{A}$ by Eq. (11). Choosing smooth basis functions has the advantage of differentiability at the expense that all basis functions need to be evaluated, which, however, for the reduced model has little impact since the number of basis functions is rather small. Section III B compares efficiencies of interpolation and basis function evaluation.

III. BENCHMARKING OF INTER/EXTRAPOLATION AND FITTING ALGORITHM

Benchmarking of the algorithm has been done w.r.t. two important figures of merit, namely accuracy and efficiency. Unfortunately, they are coupled to each other, and generally, the optimization of one is at the expense of the other.

Accuracy of fitting can be increased by including additional terms, thus reducing efficiency drastically since an increased amount of computing power is consumed. For a given method as PIC, it is up to the programmer to define the proper balance. The new algorithm shall improve this trade-off, i.e., while keeping the efficiency, the accuracy can be improved or vice versa.

A. Accuracy

In the following, the map of the cell with parameters $m = 2$, $r = 1.1$, $l = 14$ is simulated for comparison with the map generated from parameter inter/extrapolation [Eq. (5) with coordinates $(2, 1.1, 14, [0, l], [0, 2], [0, 2])$]. These parameters refer to a cell in the middle of the RFQ. The volume of the original potential is expanded to $x = [0, 2]$ and $y = [0, 2]$ accordingly. This simulation will serve as a benchmark of accuracy in the following steps.

All 1D interpolations and extrapolations call the same functions imported from the PYTHON module "scipy.interpolate.interp1d." The quadratic method is used, being a common spline technique to interpolate functions. Figure 6 displays one slice of the simulated 3D potential within the specified cell, the inter/extrapolated map, and their relative difference. The results demonstrate that within the relevant volume enclosing the beam bulk, the accuracy is within the requirements, which for rf cells is less than 1%. Within the adjacent volume, the deviations remain low. They increase sharply as the distance from the center increases considerably, i.e., beyond the volume populated by relevant amounts of beam particles. It is reiterated that inside the volume being relevant for beam dynamics simulations, the deviations are below 1%. Accordingly, the method delivers sufficiently accurate potential maps.

In the following, three different paths parallel to the beam axis are chosen for evaluation. The resulting comparison between simulated and inter/extrapolated potential distributions is presented in Fig. 7.

All these three paths are located inside of the extrapolated volume. Two of them are partially inserted into the metallic vane tips where the real potential is constant. On the surface of the vane tips, the potential distributions feature a sharp corner indicating infinite large curvature. Such nonsmooth distributions at boundaries shall be removed from the original data since they lower the quality of HOSVD. Such data fractions introduce many nonsmooth (nondifferentiable) singular vectors with low singular

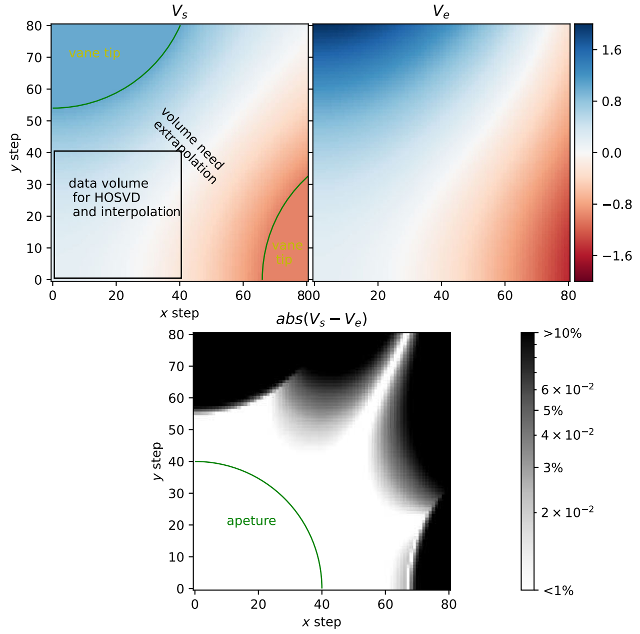


FIG. 6. Upper left: simulated potential map V_s for an RFQ cell with parameters $m = 2$, $r = 1.1$, $l = 14$, $z = 0.4l$, $x = [0, 2]$, $y = [0, 2]$. Upper right: potential map from inter/extrapolation. Lower: relative difference between simulated and inter/extrapolated potential maps.

values, being less distinguishable from noise. Such singular vectors do not occur in smooth distributions.

Interpolation produces field maps of high precision within the beam aperture. Extrapolation successfully fills metallic volumes with unnatural but smooth values, such that transitions from vacuum to metallic volumes are continuous and physical. Since the data are denoised, results from interpolation inside of the volume being relevant for the beam are physically correct. As a result, the proposed algorithm produces accurate potential maps inside the relevant volume; it also extrapolates to adjacent volumes (containing low fractions of the beam) to the same precision. Interpolation of the geometric parameters m, r, and l delivers the same accuracy.

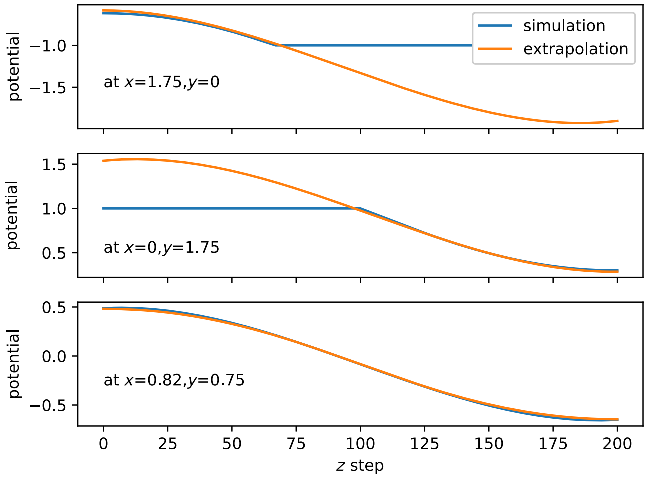


FIG. 7. Potential distribution along three straight paths parallel to the beam axis. In the upper and middle case, the path is partially inside of the metallic vane tip, in which the real potential is constant. However, the extrapolation method fills this volume with a smooth distribution, i.e., it ignores the metallic boundaries and presumes a vacuum instead.

The fitting function obtained in Sec. II C 2 has been evaluated and compared as well with the reference data from simulations. As an illustration, Fig. 8 shows the 1D fitting of $u^{(4)}$.

There are three high-valued singular vectors preserved in $u^{(4)}$, drawn with dotted lines in Fig. 8, and their corresponding basis function expressions are drawn with solid lines. The coefficients for the basis functions are $K^{(4)}$ from Table II. Figure 8 demonstrates that the function expressions for $u_1^{(4)}$ and $u_2^{(4)}$ fit the z dependence to very good precision. There remains some deviation concerning $u_3^{(4)}$, which can be eliminated by introducing one more term to the polynomial. However, this will not be of any considerable benefit since the singular value of $u_3^{(4)}$ is much lower than those of $u_1^{(4)}$ and $u_2^{(4)}$.

Figure 9 shows the comparison between the simulated potential map V_s and the values obtained from the expansion $\bar{A}$. It uses a 2D map with parameters $m = 2$, $r = 1.1$, $l = 14$, $z = 0.4$, $x = [0, 2]$, and $y = [0, 2]$.

The algorithm serves to find a multivariate formula modeling the data to very good precision, hence demonstrating its high efficiency. The results demonstrate that the obtained basis function expression can represent simulations with high accuracy, i.e., with deviations of less than 1% within the beam aperture volume and its close surrounding. It is possible to further reduce the residuals by including more terms. However, these residuals are already

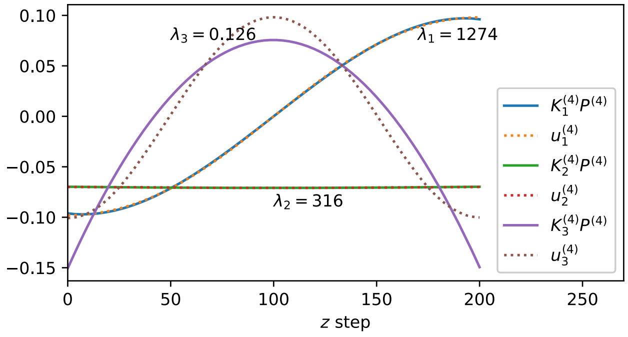


FIG. 8. Fitting of $u^{(4)}$ through basis function expressions using four terms of Legendre polynomials. The fittings of other bases $u^{(l)}$ are as good as of $u_1^{(4)}$ and $u_2^{(4)}$. Fitting of $u_3^{(4)}$ is worse but its low singular value makes this loss of accuracy effectively negligible. λ_1, λ_2, and λ_3 are the first three elements of $\lambda^{(4)}$ in Eq. (3).

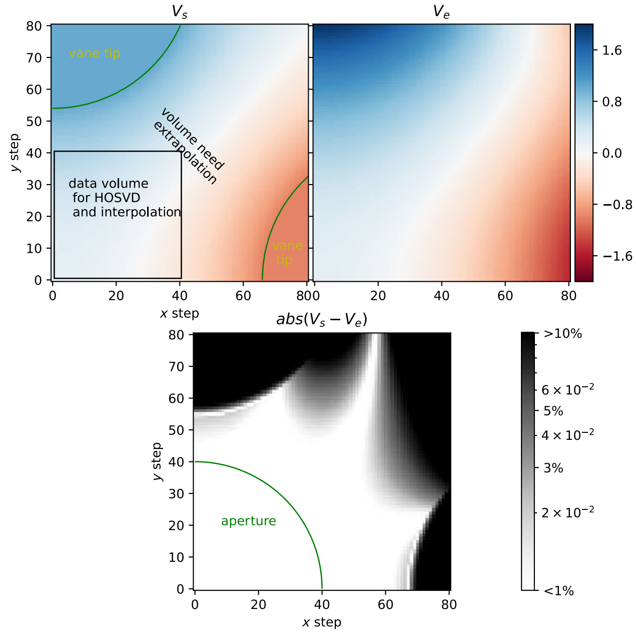


FIG. 9. Upper left: simulated potential map V_s for an RFQ cell with parameters $m = 2$, $r = 1.1$, $l = 14$, $z = 0.4$, $x = [0, 2]$, and $y = [0, 2]$. Upper right: basis function potential map V_a. Lower: relative difference between basis function potential and the simulated potential map.

sufficiently small for beam dynamic simulations. It is re-emphasized that the geometric parameters are also included in the expression, as the algorithm conceptually does not distinguish between geometric parameters and spatial coordinates.

B. Efficiency

Rigorous discussion of efficiency requires expert coding skills. Accordingly, just a preliminary evaluation shall be presented here. The PYTHON code shared in GitHub pays attention to the clarity of logic but it still lacks code efficiency w.r.t. treatment of tensor transpose operations. The algorithm should be implemented through an efficient low-level language, properly chosen for the specific application. Tests have been performed using a desktop computer with an Intel Core W Processor 2155 @ with a speed of 3.3 GHz and running PYTHON with one single processor. The runtime for the decomposition of the sixth-order potential map was 96 s and the runtime for tensor recovery was just 4 s. As mentioned at the beginning of the section, runtime for estimation at a given range of spatial coordinates is crucial for the efficiency of any simulation code. A set of dummy data for tensors of flexible order and each n mode with 50 dimensions has been established. After HOSVD, each n mode has been trimmed from 50 to just 3 dimensions.

The widely used PYTHON mathematical module “scipy.interpolate.RegularGridInterpolator” is taken for reference. Unfortunately, this module just supports “linear” and “nearest” interpolation for tensors. Another module called “scipy.interpolate.interpn” supports higher degree interpolation methods for 1D and 2D cases only, however.

In the first step, a numerical test of the module “scipy.interpolate.interp1d” has been done, which is used for interpolation and fitting of singular vectors. It assumed a 1D degenerate distribution of 50 data points fitted by three terms of polynomial/Fourier series. It revealed that to obtain a value at a specific location, the evaluation of the polynomial/Fourier series is 20/4 times faster than the interpolation of the degenerate distribution. It must be emphasized that more complex expressions are usually slower in comparison.

Afterward, 1000 random n-dimensional coordinates have been generated for interpolation of n-dimensional data. The scipy-module is capable to treat all coordinates at once. Being a well-coded module, it divides tasks into multiple subtasks and processes them simultaneously. A loop of 1000 steps has been used for both, the algorithm presented here and the scipy-module, such that they can be compared under the same conditions. Table III lists runtimes for the 1000 data points interpolation of a tensor of nth order.

Traditional linear interpolation requires 2^n n-dimensional points to be taken as surroundings. The present algorithm requires $2 \times (3 \times n)$ 2D points instead. Thus, the algorithm is advantageous for high orders of interpolated tensors, hence especially for interpolation methods using orders higher than linear. For instance, with third-order methods applied to 3D data, the new algorithm takes $3 \times 3 \times 8$ surrounding 2D points, whereas the conventional method takes 8^3 3D data points.

The algorithm works very well in treating data on aligned mesh grids, thus the 50^5 dummy data have been resampled to $50^2 \times 300^3$ by interpolation. With the reduced model, the required time is 0.13 s. This duration does not increase much as the degree of interpolation is augmented from linear to cubic. Though for the scipy-module, the duration is 19 s, even if the module is well optimized and tasks are processed in parallel. It is concluded that in such cases the algorithm’s benefits are overwhelming.

TABLE III. CPU runtime (seconds) for 1000 random spatial interpolations. Third column: time for the interpolation algorithm proposed here, fourth column: time for evaluation of multivariant polynomial with three terms per variant, fifth column: time for interpolation performed by the SciPy module.

Tensor order	Data volume	Interpolation “cubic”	Polynomial three terms	SciPy module “linear”
2	50^2	0.10	0.05	0.08
3	50^3	0.16	0.08	0.16
4	50^4	0.22	0.11	0.36
5	50^5	0.28	0.13	0.80

Finally, the order for interpolation can be selected by just setting this order for 1D interpolation of singular vectors. In contrast, traditional methods require rewriting the entire code for implementation.

Surface-based fitting methods employ field values on 2D surfaces to determine an appropriate fitting function, albeit 3D degenerate data are available. Instead, the new algorithm uses the full 6D data to obtain a fitting function and in consequence, it is significantly more robust w.r.t. noise.

The presented algorithm is quite distinguished from 2D surface-based approaches. Existing methods start from expansions following the Laplace equation, while the new one paves a path to finding fitting functions of any type. These fitting functions and the data intrinsically fulfill the Laplace equation at a sufficient level since they originate from FEM simulations.

The free choice of coordinate systems and basis functions comes with many benefits, e.g., here the Cartesian coordinate system has been chosen together with a fourth-order polynomial, being very beneficial to the efficiency. The basis function is no longer restricted to lengthy circular cylinder harmonic (multipole) expansions whose evaluation is much slower than polynomials of the same term. Additionally, this kind of basis sometimes introduces boundary conditions that are not conducive to extrapolation. Finally, geometric parameters are also included in the fitting function, thus further parameter interpolation through look-up is obsolete.

IV. CONCLUSION

For the first time, multivariate inter/extrapolation and fitting through model order reduction have been applied to model particle accelerators, here the e.m. fields of rf cavities. It allows for the extension of the concept of inter/extrapolation of spatial coordinates to geometric parameters determining the boundaries of the map. Additionally, basis function expressions can be extracted which match simulated data very well. These expressions are free from unphysical noise and preserve the intrinsic field properties of a given class of field maps. Inter/extrapolation and extraction of basis function formulas based on data previously compressed by HOSVD, which has been reported earlier [1]. Compared to traditional methods introduced in [11], the new method significantly improves the quality of field description and provides many other practical advantages. The related routine needs just a few hundred lines of (PYTHON) code and an auxiliary file with some tens of KB to store the compressed field.

Future efforts, being beyond the scope of this paper, shall be put on the same data embedded into cylindrical coordinates for scenarios of appropriate symmetry. This aims especially at the treatment of maps from RFQs for which design codes use coordinate projection on cylindrical harmonic base functions.

ACKNOWLEDGMENTS

The author expresses his great appreciation to Dr. Yaoshuo Yuan (IHEP) for fruitful discussions during the development of this work.

APPENDIX A: CONCEPT OF INTERPOLATION AND EXTRAPOLATION

Within the numerical analysis, interpolation is a type of estimation, i.e., a method of constructing new data points within the range of a discrete set of known data points. Extrapolation instead is a type of estimation on new data points beyond this range, i.e., into the trial space. Any new data point is constructed from its relation to existing data points. It is similar to interpolation, using assumptions for this relation. Extrapolation is subject to enhanced uncertainty and the risk of producing meaningless results. Figure 10 shows a simple example of inter/extrapolation of arbitrary 1D data with noise.

Accordingly, multivariate inter/extrapolation is inter/extrapolation of multidimensional functions of more than one variable. In case the variates are spatial coordinates, it is called spatial interpolation.

APPENDIX B: CALCULATION OF COEFFICIENTS FOR BASIS REPRESENTATION

With the relation of Eq. (9), the coefficients $K^{(n)}$ can be calculated with the least square method as

$$K^{(n)} = u^{(n)} \hat{P}^{(n)T} (\hat{P}^{(n)} \hat{P}^{(n)T})^{-1}, \tag{B1}$$

where the number of terms t_n should be much less than I_n, such that the equation for $K^{(n)}$ is overdetermined. The explicit expression of elements in $\hat{P}^{(n)} \hat{P}^{(n)T}$ is

$$(\hat{P}^{(n)} \hat{P}^{(n)T})[a, b] = \hat{P}_a^{(n)} \hat{P}_b^{(n)T} = \sum_{i=1}^{I_n} \hat{P}_a^{(n)}[i] \hat{P}_b^{(n)}[i]. \tag{B2}$$

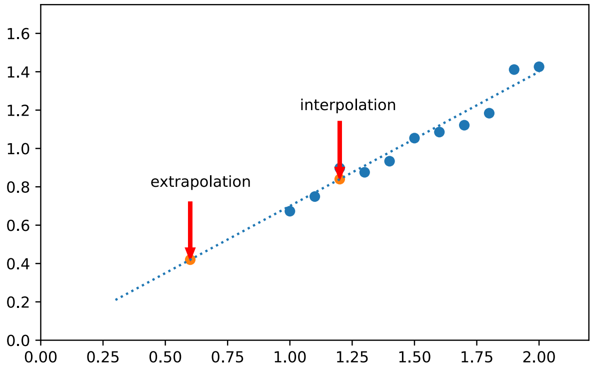


FIG. 10. Example for one-dimensional interpolation and extrapolation.

For each dimension, a square matrix $D^{(n)} \in \mathbb{R}^{t_n \times t_n}$ is defined with elements to be calculated from

$$D^{(n)}[a,b] = \int_{-1}^{1} P_a^{(n)} P_b^{(n)}. \tag{B3}$$

$D^{(n)}$ is independent of the original data, which is a feature originating from the chosen basis $P^{(n)}$. It is known once $P^{(n)}$ has been chosen. Its elements, specified by entries $D^{(n)}[a,b]$, are the inner products of the ath term and the bth term of $\hat{P}^{(n)}$, i.e., $D^{(n)}$ is the identity matrix if $P^{(n)}$ forms an orthonormal basis.

Approximating the sum through an integral as

$$\sum_{i=1}^{I_n} \hat{P}_a^{(n)}[i] \hat{P}_b^{(n)}[i] \approx I_n^2 \int_{-1}^{1} P_a^{(n)} P_b^{(n)}, \tag{B4}$$

one obtains

$$\hat{P}^{(n)} \hat{P}^{(n)T} \approx D^{(n)}. \tag{B5}$$

Plugging Eq. (B5) into Eq. (B1), the solution for $K^{(n)}$ is

$$K^{(n)} = u^{(n)} \hat{P}^{(n)T} D^{(n)-1} \frac{1}{I_n^2}. \tag{B6}$$

After $K^{(n)}$ has been determined for all dimensions and plugging Eq. (9) into Eq. (4), the data can be expressed as

$$\begin{aligned} A &\approx s \times_1 K^{(1)} \hat{P}^{(1)} \times_2 K^{(2)} \hat{P}^{(2)} \ldots \times_6 K^{(6)} \hat{P}^{(6)} \\ &= \bar{S} \times_1 \hat{P}^{(1)} \times_2 \hat{P}^{(2)} \ldots \times_6 \hat{P}^{(6)}, \end{aligned} \tag{B7}$$

where the new coefficient tensor $\bar{S} \in \mathbb{R}^{j_1 \times j_2 \ldots \times j_6}$ is expressed as

$$\bar{S} = s \times_1 K^{(1)} \times_2 K^{(2)} \ldots \times_N K^{(6)}. \tag{B8}$$

Afterward, the multivariate fitting function $\bar{A}(m,r,l,z,y,x)$ can be found as a tensor product expressed by Eq. (11) by simply replacing $\hat{P}^{(n)}$ in Eq. (B7) by their basis function version $P^{(n)}$.

[1] X. Du and L. Groening, Compression and noise reduction of field maps, Phys. Rev. Accel. Beams **21**, 084601 (2018).

[2] K. R. Crandall, PARMTEQ, a beam dynamics code for the RFQ linear accelerator, AIP Conf. Proc. **177**, 22 (1988).

[3] http://www.ghga.com/accelsoft/lidosrfq.html.

[4] K. R. Crandall, Radio-frequency quadrupole vane-tip geometries, Los Alamos Report No. LA-9695-MS, 1983.

[5] M. Venturini and A. J. Dragt, Accurate computation of transfer maps from magnetic field data, Nucl. Instrum. Methods Phys. Res., Sect. A **427**, 387 (1999).

[6] C. E. Mitchell and A. J. Dragt, Accurate transfer maps for realistic beam-line elements: Straight elements, Phys. Rev. ST Accel. Beams **13**, 064001 (2010).

[7] R. M. G. M. Trines, S. J. L. van Eijndhoven, J. I. M. Botman, T. J. Schep, and H. L. Hagedoorn, Modeling the fields of magneto-optical devices, including fringe field effects and higher order multipole contributions, with application to charged particle optics, Phys. Rev. ST Accel. Beams **4**, 062401 (2001).

[8] Y. Giboudot and A. Wolski, Use of transfer maps for modeling beam dynamics in a nonscaling fixed-field alternating-gradient accelerator, Phys. Rev. ST Accel. Beams **15**, 044001 (2012).

[9] D. T. Abell, Numerical computation of high-order transfer maps for rf cavities, Phys. Rev. ST Accel. Beams **9**, 052001 (2006).

[10] R. Tiede, G. Clemente, H. Podlech, U. Ratzinger, A. Sauer, and S. Minaev, LORASR code development, in *Proceedings of the 10th European Particle Accelerator Conference, EPAC06, Edinburgh, Scotland, 2006* (EPS-AG, Edinburgh, Scotland, 2006) pp. 2194–2196, paper WEPCH118.

[11] R. R. Ferdinand, R. Duperrier, J. M. Lagniel, P. Mattei, and S. Nath, Field description in an RFQ and its effect on beam dynamics, in *Proceedings of the 19th International Linear Accelerators Conference, Chicago, IL, 1998*, (NTIS, Springfield, VA, 1998), p. 115.

[12] K. R. Crandall, Effects of vane-tip geometry on the electric fields in radio-frequency quadrupole linacs (1983), https://www.osti.gov/biblio/6004768.

[13] M. Vasilescu and D. Terzopoulos, Tensortextures: Multilinear image-based rendering, ACM Trans. Graphics, **23**, 336 (2004).

[14] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, Tensor decompositions for learning latent variable models, arXiv:1210.7559.

[15] W. Austin, G. Ballard, and T. G. Kolda, Parallel tensor compression for large-scale scientific data, in *Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS, Chicago, IL* (IEEE, New York, 2016), 10.1109/ipdps.2016.67.

[16] N. Kreimer and M. D. Sacchi, A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation, Geophysics **77**, V113 (2012).

[17] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. **21**, 1253 (2000).

[18] https://www.3ds.com/zh/products-services/simulia/products/cst-studio-suite/.

[19] G. Xialing and L. Zihua, Joint of two dynamics design codes for high current RFQ, At. Energy Sci. Technol. **37**, 208 (2003).

[20] https://github.com/duxngsi/Field_maps_HOSVD-compress-fitting.

[21] W. A. Light and E. W. Cheney, in *Approximation Theory in Tensor Product Spaces*, 5th ed., edited by E. Beutler, M. A.

Lichtman, B. W. Coller, and T. S. Kipps (McGraw-Hill, New York, 1994), Vol. 2, Chap. 7, pp. 654–662.

[22] K. Jetter, M. Buhmann, W. Haussmann, R. Schaback, and J. Stoeckler, *Topics in Multivariate Approximation and Interpolation* (Elsevier Science, New York, 2005).

[23] P. J. Olver, On multivariate interpolation, Stud. Appl. Math. **116**, 201 (2006).

[24] C. R. de Boor, On the evaluation of box splines, Numerical Algorithms **5**, 5 (1993).

[25] M. J. Lai, Multivariate splines and their applications, in *Encyclopedia of Complexity and Systems Science*, edited by R. Meyers (Springer, New York, NY, 2009), ISBN 978-0-387-75888-6, 10.1007/978-0-387-30440-3_345.

[26] B. Mustapha, A. A. Kolomiets, and P. N. Ostroumov, Full three-dimensional approach to the design and simulation of a radio-frequency quadrupole, Phys. Rev. ST Accel. Beams **16**, 120101 (2013).