



Radial basis function interpolation of fields resulting from nonlinear simulations

Boukje M. de Gooijer¹ · Jos Havinga¹ · Hubert J. M. Geijselaers¹ · Anton H. van den Boogaard¹

Received: 8 April 2022 / Accepted: 22 December 2022 / Published online: 25 January 2023
© The Author(s) 2023

Abstract

Three approaches for construction of a surrogate model of a result field consisting of multiple physical quantities are presented. The first approach uses direct interpolation of the result space on the input space. In the second and third approaches a Singular Value Decomposition is used to reduce the model size. In the reduced order surrogate models, the amplitudes corresponding to the different basis vectors are interpolated. A quality measure that takes into account different physical parts of the result field is defined. As the quality measure is very cheap to evaluate, it can be used to efficiently optimize hyperparameters of all surrogate models. Based on the quality measure, a criterion is proposed to choose the number of basis vectors for the reduced order models. The performance of the surrogate models resulting from the three different approaches is compared using the quality measure based on a validation set. It is found that the novel criterion can effectively be used to select the number of basis vectors. The choice of construction method significantly influences the quality of the surrogate model.

Keywords Metamodel · Surrogate model · Multiphysical field · Proper orthogonal decomposition · Truncation criterion

1 Introduction

Computational fluid dynamics (CFD) and finite-element (FE) analyses are powerful tools to solve engineering problems for which no analytical solution can be found. However, for nonlinear problems with many degrees of freedom these methods can be computationally costly. Consequently the applicability of such models for direct use in inverse analyses, (robust) optimization or control algorithms is limited.

For these purposes a surrogate model can be constructed [16, 25]. Surrogate models are cheap to evaluate approximation models that mimic the output of expensive models. The CFD and FE models can be considered as a black box mapping from an input space to an output space. The relations between inputs and outputs can be approximated by fitting a function of input data to output data that are obtained from evaluations of the considered FE or CFD model [8].

In many cases, surrogate modelling (also referred to as metamodelling) is used to map a multidimensional input space to a scalar output space. To construct surrogate models, the expensive model is evaluated on predefined sample points in the input parameter space. The sample points and the results of the expensive model together are referred to as the training set.

Many different metamodelling techniques exist [16, 18, 25, 38]. In our work, metamodelling of metal forming processes are studied. These processes are a suitable benchmark because they have strong nonlinearities involved. For example in the work of Wei et al. [36] the wrinkling tendency in a forming process of a car sheet part is expressed as a scalar function of process parameters. In their work the response of an FE-model is replaced with a response surface model to optimize the process. In the work of Wiebenga et al. [37] an

Jos Havinga, Hubert J. M. Geijselaers, Anton H. van den Boogaard contributed equally to this work.

✉ Boukje M. de Gooijer
b.m.degooijer@utwente.nl

Jos Havinga
jos.havinga@utwente.nl

Hubert J. M. Geijselaers
h.j.m.geijselaers@utwente.nl

Anton H. van den Boogaard
a.h.vandenboogaard@utwente.nl

¹ Chair of Nonlinear Solid Mechanics, University of Twente, Drienerlolaan 5, Enschede, Overijssel 7522 NB, The Netherlands

industrial V-bending process is optimized. In the V-bending process the flange shape is defined by the main angle, a scalar, which is modelled using a kriging metamodel. Other examples in which metamodels replace metal-forming process simulations use polynomial regression or response surface methodology (RSM) [4, 14, 15, 27, 30, 33, 36], support vector regression (SVR) [31, 34, 35], Multivariate Adaptive Regression Splines (MARS) [3, 21], kriging ([6, 32], radial basis functions (RBF) [29] and neural networks [20, 28]. Some of the aforementioned metamodeling techniques, such as kriging and RBF, interpolate the existing data. Other techniques, such as RSM and SVR, are regression methods. In this work we will focus on interpolating metamodeling techniques and we will refer to the output that is being interpolated as the interpolant.

In several applications, it is required to model a result array instead of a scalar. Then the interpolant consists of more than one variable and the output field may even contain different physical quantities at the same time. When the training set for full field surrogate models is large, reduction techniques can be applied. A commonly used reduction technique is the Proper Orthogonal Decomposition. Proper Orthogonal Decomposition (POD) is an umbrella term that includes Principal Component Analysis (PCA), Karhunen–Loeve Expansion (KLE) and Singular Value Decomposition (SVD) [19]. The goal of the decomposition is to obtain a low-dimensional representation of the output field. This is done by seeking the predominant modes in the output data of the training set and use these as a new basis.

POD is usually combined with RBF interpolation. For example, in Hamdaoui et al. [11], a POD-based surrogate is used to describe the displacement field of the stamping of an axisymmetric cup to model the major and minor strain in a Forming Limit Diagram. In Dang et al. [6] the displacement field is described by a POD-based surrogate model to optimize the shape of a metal product after springback.

Most metamodels have hyperparameters associated with them that influence the fitting quality of the metamodels in the output space. Different criteria can be used to choose these parameters in order to optimize the fitting quality of a metamodel. We will refer to the criterion that is used for optimizing the hyperparameters as the fitting criterion. In this work, we will focus on RBF as an interpolation method. To quantify the quality of the interpolation, different fitting criteria can be determined, such as the likelihood criterion for kriging interpolation [17], or based on the Leave-One-Out (LOO) cross-validation values for RBF interpolation [24].

The key question in this work is how to efficiently create an accurate reproduction of a result array consisting of multiple physical fields using RBF interpolation. Three different approaches for construction of such surrogate model are presented. The main difference between the approaches

is the interpolant. The first approach is based on a direct interpolation of the result array on the input space. The second and third approach will involve model reduction. In these approaches a reduced basis is determined using singular value decomposition (SVD). The left singular vectors will be the basis vectors in the truncated basis. By projecting the result arrays onto the basis vectors, the amplitudes of the result arrays in the new basis can be found. These amplitudes corresponding to different basis vectors are interpolated in two different ways, by means of array and scalar interpolation. For array interpolation, the interpolant will be the array that collects the amplitudes corresponding to all basis vectors in the basis. For scalar interpolation, the interpolant will be a scalar that describes the amplitude corresponding to a single basis vector. In the case of array interpolation all amplitudes are interpolated simultaneously, whereas in the case of scalar interpolation, the amplitudes of the separate basis vectors are interpolated independently. In this work, the hyperparameters corresponding to the different approaches for surrogate model construction will be optimized. To do so, a new quality measure that takes into account the physical parts is used as a fitting criterion.

When a reduced order model is constructed, the number of basis vectors to be retained in the truncated basis has to be determined. While inclusion of more basis vectors generally increases its accuracy, it can also add more noise, especially in the higher order basis vectors [26]. A new criterion for choosing the number of basis vectors to be included in the truncated basis of the reduced order models is proposed.

This paper is organized as follows: in Sect. 2, three approaches for surrogate model construction are described. Thereafter, Sect. 3 describes the quality measure and the different sources of error that can be distinguished. Furthermore, it is described how the quality measure and sources of error can be used in a fitting criterion, as well as in a criterion to choose the number of basis vectors in the truncated basis. A demonstrator process is introduced in Sect. 4. In Sect. 5, the performance of the three approaches is compared.

2 Constructing different surrogate models

Three surrogate models of a result field consisting of different physical parts are constructed. The described approaches are generic and can be applied to any result field. As an example the output of a single FE simulation consisting of M variables is considered. The variables can be stored in an $M \times 1$ result vector \mathbf{y} . In this case, it is assumed that the output is fully described using three physical fields, being displacement u , equivalent plastic strain ε and stress σ . The different physical parts are partitioned as subarrays in the result array:

$$\mathbf{y} = \left\{ \begin{matrix} \{\mathbf{y}_u\} \\ \{\mathbf{y}_\varepsilon\} \\ \{\mathbf{y}_\sigma\} \end{matrix} \right\}, \tag{1}$$

in which \mathbf{y}_u contains M_u displacements, \mathbf{y}_ε contains M_ε equivalent plastic strains and \mathbf{y}_σ contains M_σ stress components.

To construct the different surrogate models, a training set needs to be obtained. The training set is obtained by sampling the input space and evaluating the black-box model, e.g. a FE simulation, on the sample points \mathbf{x} .

The training set consists of the set of inputs \mathbf{X} and the corresponding result arrays that are collected in the matrix \mathbf{Y} :

$$\mathbf{X} = \left[\begin{matrix} \dots & \{\mathbf{x}_i\} & \dots \\ N_{\text{dim}} \times N_{\text{exp}} \end{matrix} \right] \xrightarrow{\text{FEA}} \mathbf{Y} = \left[\begin{matrix} \dots & \{\mathbf{y}_i\} & \dots \\ M \times N_{\text{exp}} \end{matrix} \right], \tag{2}$$

in which N_{dim} is the dimensionality of the input space and N_{exp} is the number of sample points or experiments in the training set. We will refer to the matrix \mathbf{Y} , in which the result arrays are collected, as the snapshot matrix.

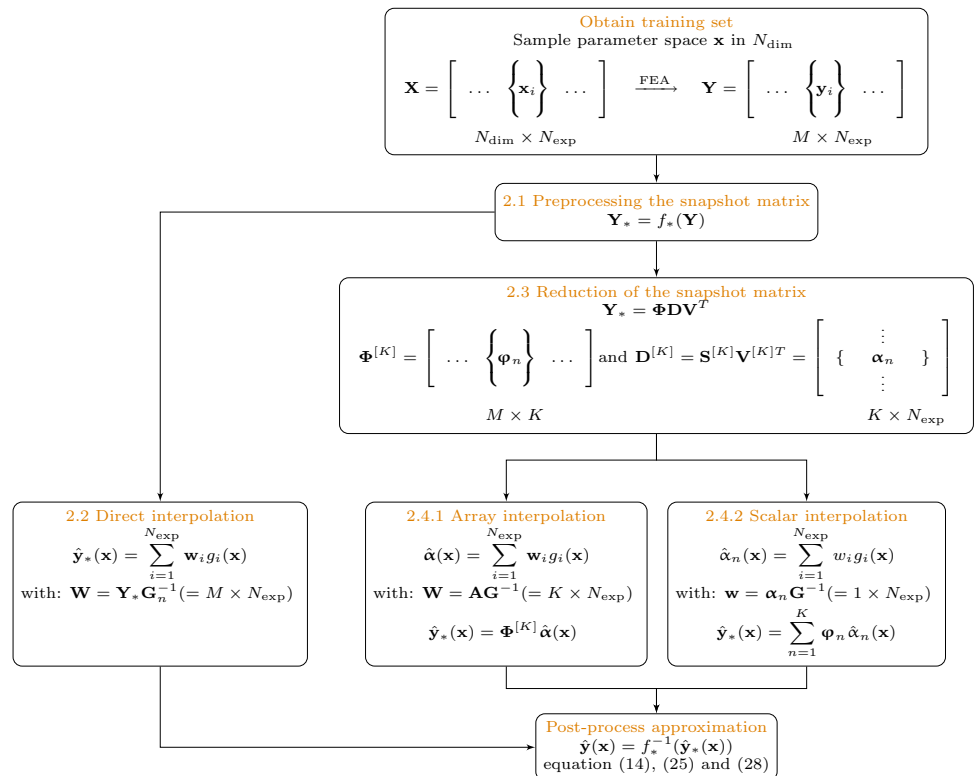
Three different surrogate models will be constructed. An overview of the construction procedures for the different surrogate models is presented in Fig. 1. The first surrogate model is obtained by directly interpolating the result array

on the input space, as will be described in Sect. 2.2. This method is referred to as ‘Direct interpolation’. In this method the interpolant will be directly related to the result array. The full result space that is available based on the training set will be used for interpolation.

The second and third surrogate models are reduced order models, in which singular value decomposition (SVD) is used to reduce the output space. In an SVD the predominant modes in the output data are sought to form a new orthogonal basis. As the first vectors in this basis describe the most variation, the basis can be truncated to K basis vectors. The output fields are projected onto this truncated basis to find the amplitudes. These amplitudes will be interpolated to find a continuous surrogate model. To construct the second surrogate model, the amplitudes corresponding to all basis vectors are interpolated on the input space as one array as described in Sect. 2.4.1. Hence, the interpolant is a K -dimensional array and there will be only one set of optimized hyperparameters for all the amplitudes corresponding to different basis vectors. This method is referred to as ‘SVD, array interpolation’.

For the construction of the third surrogate model, the amplitude function is interpolated for each basis vector separately as will be described in Sect. 2.4.2. Hence, the interpolants are K scalars. The hyperparameters are optimized for the amplitude function for each basis direction. This method is referred to as ‘SVD, scalar interpolation’ and is depicted with the rightmost path in Fig. 1.

Fig. 1 Flowchart of the construction of three different surrogate models



The reduced order models are cheaper to evaluate compared to the surrogate model that is based on ‘Direct interpolation’. However, as no reduction is applied, it is expected that the ‘Direct interpolation’ model will be most accurate. The reduced order model that is based on scalar interpolation of the amplitudes has more freedom to optimize the model parameters. Although the flexibility in the fitting procedure increases the risk of overfitting [23], SVD combined with scalar interpolation is expected to result in a more accurate surrogate model than SVD combined with array interpolation.

2.1 Preprocessing the snapshot matrix

To obtain more accurate surrogate models, the output data in the training set will be preprocessed. The function $f_*(\cdot)$ transforms the snapshot matrix into the preprocessed snapshot matrix \mathbf{Y}_* , where the sub- and superscript respectively denote the applied preprocessing. The function that reverses the applied preprocessing is called the post-processing function and is denoted with $f_*^{-1}(\cdot)$.

In all three approaches the mean is subtracted from the snapshot matrix. The mean of each row in the snapshot matrix is calculated using

$$\bar{\mathbf{y}} = \frac{1}{N_{\text{exp}}} \mathbf{Y} \mathbf{1}_{N_{\text{exp}}}, \quad (3)$$

in which $\mathbf{1}_{N_{\text{exp}}}$ is an $N_{\text{exp}} \times 1$ vector of ones, following the notation as used by Pronzato [22]. The zero centered snapshot matrix will be

$$\mathbf{Y}_0 = f_0(\mathbf{Y}) = \mathbf{Y} - \bar{\mathbf{y}} \mathbf{1}_{N_{\text{exp}}}^T. \quad (4)$$

When the output data consist of different physical parts, e.g., the displacement, strain and stress as shown in Eq. (1), the data of the physical parts can be stored in one snapshot matrix or in separate snapshot matrices. These snapshot matrices can be reduced by means of a Proper Orthogonal Decomposition. It has been shown in earlier work that decomposing the different physical parts in one matrix improves the overall accuracy of the surrogate model [7].

Typically, the strain components are in the order of 10^{-1} , while the stress components are in the order of 10^1 – 10^2 MPa (or 10^7 – 10^8 Pa). When the different physical quantities are of different order, scaling must be applied to improve the decomposition. Without scaling, the decomposition will be dominated by the component with the highest numerical values [7]. Guéntot [10] proposed to scale the snapshot matrix with the range, mean or standard deviation. In this work it is chosen to scale each physical part by its range of observed values in the data set.

For example a scaling constant s_u of the displacement field is calculated as

$$s_u = \left(\max \left(\mathbf{Y}_u - \bar{\mathbf{y}}_u \mathbf{1}_{N_{\text{exp}}}^T \right) - \min \left(\mathbf{Y}_u - \bar{\mathbf{y}}_u \mathbf{1}_{N_{\text{exp}}}^T \right) \right)^{-1}. \quad (5)$$

Now we can define a scaling array $\mathbf{s} = \{s_m\}$ as

$$s_m = \begin{cases} s_u & m = 1 \dots M_u \\ s_\varepsilon & m = M_u + (1 \dots M_\varepsilon) \\ s_\sigma & m = M_u + M_\varepsilon + (1 \dots M_\sigma) \end{cases}. \quad (6)$$

The preprocessed snapshot matrix takes the following form:

$$\mathbf{Y}_{\text{scaled}} = f_{\text{scaled}}(\mathbf{Y}) = \text{diag}(\mathbf{s}) \left[\mathbf{Y} - \bar{\mathbf{y}} \mathbf{1}_{N_{\text{exp}}}^T \right], \quad (7)$$

2.2 Direct interpolation

The first method to construct a surrogate model directly interpolates the result array on the input parameter space. To construct a continuous surrogate model of the result array, the data in the zero centered snapshot matrix from Eq. (4) are interpolated. The corresponding interpolant is an array.

To perform the interpolation, radial basis functions (RBF) will be used. In RBF interpolation, a basis function is placed at the location of each data point in the input parameter space. We will derive the radial basis interpolation of arbitrary interpolant $\hat{\mathbf{f}}(\mathbf{x})$ which can be described with the output array $M \times 1$.

The RBF interpolation of interpolant $\hat{\mathbf{f}}(\mathbf{x})$ will be the sum of the weighted radial basis functions of each sample point:

$$\hat{\mathbf{f}}(\mathbf{x}) = \sum_{i=1}^{N_{\text{exp}}} \mathbf{w}_i g_i(\mathbf{x}) = \mathbf{W} \mathbf{g}(\mathbf{x}), \quad (8)$$

$M \times 1 \quad M \times 1 \quad 1 \times 1 \quad M \times N_{\text{exp}} \quad N_{\text{exp}} \times 1$

in which $g_i(\mathbf{x})$ is a radial basis function that depends on the Euclidean distance $\|\cdot\|$ between an arbitrary point \mathbf{x} and sample point \mathbf{x}_i in the training set, \mathbf{w}_i is the array that collects the M weights for the radial basis function $g_i(\mathbf{x})$ and \mathbf{W} is the weight matrix that collects all M weights corresponding to the N_{exp} radial basis functions. The weights for all basis functions can be solved using the interpolation requirement:

$$\hat{\mathbf{f}}(\mathbf{x}_j) = \mathbf{f}_j. \quad (9)$$

Leading to the following linear system of equations:

$$\mathbf{W} \mathbf{G} = \mathbf{F}, \quad (10)$$

$M \times N_{\text{exp}} \quad N_{\text{exp}} \times N_{\text{exp}} \quad M \times N_{\text{exp}}$

in which $G_{ij} = g_i(\mathbf{x}_j)$ and \mathbf{F} is the matrix with output training data of the interpolant.

Different choices can be made for the basis function $g_i(\mathbf{x})$. In many studies it has been shown that the performance of the multiquadric RBF for scalar interpolation is generally good [9, 16]. In a comparative study performed by Hamim [12] on the application of RBF in POD-based surrogate models, it was shown that multiquadric RBFs perform best. The predictive accuracy of the interpolation is improved by application of global scaling parameters θ [13]. The global scaling parameters scale the parameter space in each dimension.

The multiquadric RBF with global scaling has the following form:

$$g_i(\mathbf{x};\theta) = \sqrt{1 + (\|\text{diag}(\theta)(\mathbf{x} - \mathbf{x}_i)\|)^2}. \tag{11}$$

The global scaling parameters θ are the hyperparameters, that will be optimized based on a scalar error measure as will be described in Sect. 3.

To construct the surrogate model based on direct interpolation, the zero centered snapshot matrix is used as the matrix with output training data of the interpolant $\mathbf{F} = \mathbf{Y}_0$. The approximated result array using direct interpolation will be:

$$\hat{\mathbf{y}}(\mathbf{x}) = \bar{\mathbf{y}} + \mathbf{W}\mathbf{g}(\mathbf{x};\theta) \quad \text{in which: } \mathbf{W} = (\mathbf{Y} - \bar{\mathbf{y}}\mathbf{1}_{N_{\text{exp}}}^T)\mathbf{G}^{-1}. \tag{12}$$

Scaling of the snapshot matrix has no influence on the approximation of the result vector when direct interpolation is used to construct the surrogate model.

2.3 Reduction of the snapshot matrix

When the number of variables M in the result array of which a surrogate model will be constructed is large, storage issues may arise. In that case reduction techniques can be applied to construct the surrogate model. Another benefit of using reduction methods in the construction of surrogate models is that noise in the output field is reduced [26].

In this work, singular value decomposition (SVD) is used to find the proper orthogonal basis vectors [1]. The SVD of the preprocessed snapshot matrix takes the following form:

$$\mathbf{Y}_{\text{scaled}} = \mathbf{\Phi}\mathbf{D}\mathbf{V}^T = \left[\dots \left\{ \begin{matrix} \boldsymbol{\varphi}_n \end{matrix} \right\} \dots \right] \left[\begin{matrix} \ddots & & 0 \\ & d_n & \\ 0 & & \ddots \end{matrix} \right] \left[\begin{matrix} \vdots \\ \{ \mathbf{v}_n^T \} \\ \vdots \end{matrix} \right] \tag{13}$$

$M \times N_{\text{exp}} \quad N_{\text{exp}} \times N_{\text{exp}} \quad N_{\text{exp}} \times N_{\text{exp}}$

The preprocessed snapshot matrix is decomposed into three matrices: $\mathbf{\Phi}$ that contains the left singular vectors $\boldsymbol{\varphi}_n$ as its columns, \mathbf{D} that contains the singular values d_n on its diagonal and \mathbf{V} that contains the right singular vectors \mathbf{v}_n . The subscript n denotes the n -th direction in the basis. The left

singular vector matrix spans an orthogonal coordinate system in the result space and will be used as a basis for the data.

As the singular values are sorted by size from largest to smallest, the most information will be captured by the first singular vectors. The basis can therefore be truncated, such that it contains only the first K basis vectors. The truncated basis with K basis vectors is defined as

$$\mathbf{\Phi}^{[K]} = \left[\left\{ \begin{matrix} \boldsymbol{\varphi}_1 \end{matrix} \right\} \dots \left\{ \begin{matrix} \boldsymbol{\varphi}_K \end{matrix} \right\} \right]. \tag{14}$$

$M \times K$

Because the mean has been subtracted from the snapshot matrix, the maximum number of available basis vectors will be $N_{\text{exp}} - 1$. The projections of the result vector on the basis vectors are referred to as amplitudes. The amplitudes are found by multiplying the singular values with the right singular vectors:

$$\mathbf{A}^{[K]} = \mathbf{D}^{[K]}\mathbf{V}^{[K]T} = \left[\begin{matrix} \vdots \\ \{ \boldsymbol{\alpha}_n^T \} \\ \vdots \end{matrix} \right]. \tag{15}$$

$K \times N_{\text{exp}}$

The vector $\boldsymbol{\alpha}_n$ collects the amplitudes of the N_{exp} result vectors corresponding to basis vector $\boldsymbol{\varphi}_n$. The K -rank approximation of the preprocessed snapshot matrix $\mathbf{Y}^{[K]}$ can now be written as

$$\mathbf{Y}^{[K]} = \mathbf{\Phi}^{[K]}\mathbf{A}^{[K]}. \tag{16}$$

By rewriting Eq. (7) and substituting Eq. (16) the K -rank approximation $\mathbf{Y}^{[K]}$ of the snapshot matrix can be found:

$$\mathbf{Y}^{[K]} = \bar{\mathbf{y}}\mathbf{1}_{N_{\text{exp}}}^T + \text{diag}(\mathbf{s})^{-1}(\mathbf{\Phi}^{[K]}\mathbf{A}^{[K]}). \tag{17}$$

The i th column in this matrix is the K -rank approximation $\mathbf{y}_i^{[K]}$ of the result vector of experiment i :

$$\mathbf{y}_i^{[K]} = \bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\sum_{n=1}^K \boldsymbol{\varphi}_n \alpha_{in} \right). \tag{18}$$

Similarly, the result vector of experiment i that is solely approximated with basis vector $\boldsymbol{\varphi}_n$, is defined as

$$\mathbf{y}_i^n = \bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1}(\boldsymbol{\varphi}_n \alpha_{in}), \tag{19}$$

this approximation will be used to optimize the hyperparameters of the surrogate model constructed using ‘SVD, scalar interpolation’ in Sect. 3.4.3.

2.4 Amplitude interpolation

When SVD is applied to construct the surrogate models, the amplitudes \mathbf{A} corresponding to the different basis vectors will be interpolated to form a continuous surrogate model on the input space. This can be done using the same method as described in Sect. 2.2. The amplitudes can be interpolated using two different approaches, array interpolation or scalar interpolation. When array interpolation is used, the interpolant describes multiple amplitudes collected in an array. When scalar interpolation is used, the interpolant is the amplitude corresponding to one basis vector. For each basis vector to be included in the prediction, a separate scalar amplitude interpolation is fitted.

2.4.1 Array interpolation

If the amplitudes are interpolated using array interpolation, all amplitudes corresponding to the K basis vectors will be interpolated at once. There will be only one vector with global scaling parameters $\boldsymbol{\theta}$ for all K basis vectors. The interpolant as described in Eq. (8) will be the array with K interpolated amplitudes, $\hat{\mathbf{f}}(\mathbf{x}) = \hat{\boldsymbol{\alpha}}(\mathbf{x})$. The expression for the interpolant will be

$$\hat{\boldsymbol{\alpha}}^{[K]}(\mathbf{x}) = \mathbf{W}^{[K]} \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) \quad (20)$$

$$K \times 1 \quad K \times N_{\text{exp}} \quad N_{\text{exp}} \times 1$$

The matrix with output training data of the interpolant that is used to calculate the weights is the amplitude matrix $\mathbf{A}^{[K]}$ as given in Eq. (15). Following Eq. (10) the matrix with weights is found using

$$\mathbf{W}^{[K]} = \mathbf{A}^{[K]} \mathbf{G}^{-1} \quad (21)$$

$$K \times N_{\text{exp}} \quad K \times N_{\text{exp}} \quad N_{\text{exp}} \times N_{\text{exp}}$$

The approximated result vector using SVD and array interpolation will be

$$\hat{\mathbf{y}}(\mathbf{x}) = \bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} (\boldsymbol{\Phi}^{[K]} \hat{\boldsymbol{\alpha}}^{[K]}(\mathbf{x})). \quad (22)$$

2.4.2 Scalar interpolation

The other possibility is to interpolate each amplitude per basis direction separately. Then, the interpolant is a scalar function for each basis vector and there will be a vector with global scaling parameters $\boldsymbol{\theta}$ for each basis vector. The reasoning behind optimizing the RBF for each basis vector separately, is that it may be expected that different modes have different dependencies to the input parameters, which potentially enhances the model accuracy when optimizing all RBFs from different modes separately. The interpolant as

described in Eq. (8) will be a scalar with the amplitude corresponding to basis vector n , $\hat{f}(\mathbf{x}) = \hat{\alpha}_n(\mathbf{x})$. The expression for the interpolant will be

$$\hat{\alpha}_n(\mathbf{x}) = \mathbf{w} \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}_n) \quad (23)$$

$$1 \times 1 \quad 1 \times N_{\text{exp}} \quad N_{\text{exp}} \times 1$$

The matrix with output training data of the interpolant that is used to calculate the weights is the amplitude array $\boldsymbol{\alpha}_n^T$, a row in the amplitude matrix given in Eq. (15). Following Eq. (10) the array with weights is found using

$$\mathbf{w} = \boldsymbol{\alpha}_n^T \mathbf{G}^{-1} \quad (24)$$

$$1 \times N_{\text{exp}} \quad 1 \times N_{\text{exp}} \quad N_{\text{exp}} \times N_{\text{exp}}$$

Hence, the approximated result vector using SVD and scalar interpolation is

$$\hat{\mathbf{y}}(\mathbf{x}) = \bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\sum_{n=1}^K \boldsymbol{\Phi}_n \hat{\alpha}_n(\mathbf{x}) \right). \quad (25)$$

3 Optimization of surrogate model parameters

This section describes how to find the optimal values of the global scaling parameters $\boldsymbol{\theta}$ and how to choose the number of basis vectors K for the reduced order models. To do so, first a scalar quality measure that takes into account the different physical parts is introduced in Sect. 3.1. This quality measure will be used as a fitting criterion to find the optimal values of the global scaling parameter and will also be used to define a criterion for the number of basis vectors to be included in the reduced basis. To optimize the global scaling parameters and choose the number of basis vectors, it is important to understand the different sources of error in the surrogate models. The error between a reference solution \mathbf{y}_q and its approximated result is an $M \times 1$ array, that is referred to as the error array $\boldsymbol{\epsilon}_q = \{\epsilon_{mq}\}$.

In Sect. 5.2, the scalar quality measure that is defined in this section will also be used to assess the quality of the obtained surrogate models based on a validation set. Therefore, the reference solution \mathbf{y}_q can either be a result array in the training set, or a result array in a validation set. The resulting error arrays are denoted with $\tilde{\boldsymbol{\epsilon}}_q$ when based on the training set and $\boldsymbol{\epsilon}_q$ when based on the validation set.

Different sources of error can be distinguished that are annotated with an upper prescript. For the model that is based on direct interpolation the total error denoted with δ (Sect. 3.2) is mainly the interpolation error. In Sect. 5.2, it will be show that there is an additional small error due to inherent sparsity of data. For the reduced order models two sources of error are distinguished: the interpolation

error denoted with ι (Sect. 3.4) and the truncation error denoted with κ (Sect. 3.3). The total error is denoted with τ (Sect. 3.2).

For all approaches, a quality measure based on the error due to interpolation will be used to optimize the global scaling parameters θ . In Sect. 3.5 the relation between the interpolation error and the truncation error is used to choose the number K of basis vectors.

3.1 Quality measure considering physical parts

A quality measure is defined based on a reference solution y_q such that it can be used to calculate the error with respect to the training set as well as with respect to a validation set.

The error array contains the differences between the reference solution and the approximated result in components of different physical parts. Similar to the result vector itself, the error array can be partitioned in a part corresponding to the displacement field, $\epsilon_{u,q}$, the equivalent plastic strain field, $\epsilon_{e,q}$, and the stress tensor field, $\epsilon_{\sigma,q}$.

To obtain a scalar measure for the quality of the result field the Fraction of Variance Unexplained (FVU) is used. The fraction of variance unexplained is defined as the Sum of Squared Errors (SSE) divided by the Total Sum of Squares (SST) of the zero centered data. For example the FVU in the displacement field is

$$\begin{aligned}
 FVU_u &= \frac{SSE_u}{SST_u} = \frac{\sum_{q=1}^Q (\|\epsilon_{u,q}\|_2)^2}{\sum_{q=1}^Q (\|y_{u,q} - \bar{y}_u\|_2)^2} \\
 &= \frac{\sum_{q=1}^Q \sum_{m=1}^{M_u} (\epsilon_{u,mq})^2}{\sum_{q=1}^Q \sum_{m=1}^{M_u} (y_{u,mq} - \bar{y}_{u,m})^2}.
 \end{aligned}
 \tag{26}$$

The FVU in the equivalent plastic strain and the stress are calculated equivalently. The total FVU can be defined as

$$FVU = \frac{1}{3} (FVU_u + FVU_e + FVU_\sigma), \tag{27}$$

Due to the normalization in Eq. (26) the three different parts are equally important in the error measure, independent of the dimension of each physical field and of the magnitudes of the values.

Again a distinction is made whether the FVU is based on the training set or the validation set. The FVU based on the training set is denoted with \widetilde{FVU} .

3.2 Total reconstruction error

The total reconstruction error of the direct model ${}^\delta\epsilon$ and of the reduced order models ${}^\tau\epsilon$ are defined as the difference between a reference solution y_p and its interpolated approximation $\hat{y}(x_p)$. The total reconstruction error of a reference solution approximated using the direct interpolation is

$${}^\delta\epsilon_p = y_p - \hat{y}(x_p). \tag{28}$$

For the reduced order models, truncation of the result space will occur by removing the higher modes. The total reconstruction error of the reduced order models therefore depends on the number of basis vectors that are included in the truncated basis. The total reconstruction error of a reference solution in the validation set y_p approximated using K basis vectors will be

$${}^\tau\epsilon_p^{[K]} = y_p - \hat{y}^{[K]}(x_p). \tag{29}$$

Note that this error array describes the error fields in the different physical parts, and can be substituted into Eqs. (26) and (27) to calculate a scalar error measure that describes the quality of the model, which is denoted with ${}^\tau FVU$. For the reduced order models the ${}^\tau FVU$ is a function of the number of basis vectors included in the basis, hence we write ${}^\tau FVU^{[K]}$ to indicate the number of basis vectors included.

3.3 Truncation error

The truncation error of the training set is the difference between the result vector and its K -rank approximation as defined in Eq. (18). The two reduced order models (with array and scalar interpolation) use the same basis in construction, only the interpolation of the amplitudes is different. The truncation error in the training set for both reduced order models can be defined as

$$\begin{aligned}
 {}^\kappa\tilde{\epsilon}_i^{[K]} &= y_i - y_i^{[K]} \\
 &= y_i - \left[\bar{y} + \text{diag}(s)^{-1} \left(\sum_{n=1}^K \phi_n \alpha_{ni} \right) \right].
 \end{aligned}
 \tag{30}$$

As proposed by [2] a result vector from the validation set (which they call a supplementary observation) can be mapped into the truncated basis using a least square projection. The amplitude corresponding to basis direction n for reference solution y_p can be found using

$$\alpha_{np} = \phi_n^T [\text{diag}(s)(y_p - \bar{y})]. \tag{31}$$

The reference solution can be approximated in the truncated basis with K basis vectors as

$$\mathbf{y}_p^{[K]} = \bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} (\Phi^{[K]} \Phi^{[K]T} [\text{diag}(\mathbf{s})(\mathbf{y}_p - \bar{\mathbf{y}})]). \quad (32)$$

We define the truncation error as the difference between the reference solution and its approximation using K basis vectors. The truncation error for reference solution \mathbf{y}_p of the validation set will be

$$\kappa \mathbf{e}_p^{[K]} = \mathbf{y}_p - \mathbf{y}_p^{[K]}, \quad (33)$$

where $\mathbf{y}_p^{[K]}$ is the projection of the reference solution onto the truncated basis.

3.4 Interpolation error

The interpolation error \mathbf{e} is a measure for the error inside the result space spanned by the basis vectors that are included in the model. For a reference solution in the validation set, the interpolation error is defined as the difference between the reference solution approximated using K basis vectors as defined in Eq. (32), and the approximated result vector. The interpolation error of validation sample $\mathbf{y}(\mathbf{x}_p)$ reconstructed using K basis vectors will be

$$\mathbf{e}_p^{[K]} = \mathbf{y}_p^{[K]} - \hat{\mathbf{y}}^{[K]}(\mathbf{x}_p). \quad (34)$$

To assess the interpolation error in the training set, Leave-One-Out cross validation data will be used. In Leave-One-Out (LOO) cross validation one sample point and the corresponding result are removed from the training set and the surrogate model is reconstructed based on the retained $N_{\text{exp}} - 1$ training points. The surrogate model constructed without training point i is designated as $\hat{\mathbf{f}}^{-i}(\mathbf{x})$. By evaluating the obtained surrogate model at the point \mathbf{x}_i and comparing it with the corresponding result field \mathbf{f}_i , the LOO-error is calculated. This procedure is repeated for all N_{exp} training points to perform a cross-validation. The calculation of the LOO-error proposed in [24] can be extended to calculate the LOO-error of a result array. For an arbitrary interpolant $\mathbf{f}(\mathbf{x})$ the error of leaving out training point i is

$$\tilde{\mathbf{e}}_i = \mathbf{f}_i - \hat{\mathbf{f}}^{-i}(\mathbf{x}_i) = \frac{\mathbf{w}_i}{(\mathbf{G}^{-1})_{ii}}, \quad (35)$$

in which $\tilde{\mathbf{e}}_i$ is the array with errors, which has the same size as the interpolant.

The interpolation error based on leaving out point \mathbf{x}_i in one of the surrogate models is defined as

$${}^i\tilde{\mathbf{e}}_i = \mathbf{y}_i - \hat{\mathbf{y}}^{-i}(\mathbf{x}_i). \quad (36)$$

In the following sections, the interpolation errors in the different surrogate models are derived by factoring out the interpolant so that Eq. (35) can be substituted. To calculate the interpolation error in the different surrogate models

based on this LOO-error three assumptions are made. First it is assumed that the mean $\bar{\mathbf{y}}$ does not change when sample i is left out of the training set. Secondly, for the reduced order models, it is assumed that the obtained basis vectors do not change when sample i is left out of the training set. Lastly, it is assumed that the global scaling parameters $\boldsymbol{\theta}$ stay the same during the LOO cross validation.

The FVU and $\text{FVU}^{[K]}$ of the interpolation errors presented in the next sections are used as the scalar measure that will be minimized to determine the global scaling parameters $\boldsymbol{\theta}$.

3.4.1 Model 1: direct interpolation

Using direct interpolation, the interpolation error based on the training set is the total error. The error of leaving out point \mathbf{x}_i in the training set can be found by substituting Eq. (12) into the definition of the interpolation error in Eq. (36) and factoring out the interpolant to be able to substitute Eq. (35). The error of leaving out point \mathbf{x}_i in the surrogate model with direct interpolation gives the following error:

$$\begin{aligned} \delta \tilde{\mathbf{e}}_i &= \mathbf{y}_i - \hat{\mathbf{y}}^{-i}(\mathbf{x}_i) \\ &= [\bar{\mathbf{y}} + \mathbf{y}_{0,i}] - [\bar{\mathbf{y}} + (\hat{\mathbf{y}}_0^{-i}(\mathbf{x}_i))] \\ &= [\mathbf{y}_{0,i} - \hat{\mathbf{y}}_0^{-i}(\mathbf{x}_i)] \\ &= \left[\frac{\mathbf{w}_i}{(\mathbf{G}^{-1})_{ii}} \right], \end{aligned} \quad (37)$$

in which \mathbf{w}_i is the $M \times 1$ column of the matrix with weights corresponding to experiment i . This error array for the different physical parts is substituted into Eqs. (26) and (27) to calculate the total FVU, which is a scalar error measure that describes the quality of the model. The FVU due to interpolation in the training set will be minimized to find the best fitting metamodel parameters. Hence, the N_{dim} components of the global scaling parameter vector $\boldsymbol{\theta}$ are found using

$$\text{argmin}_{\boldsymbol{\theta}} \delta \widetilde{\text{FVU}}. \quad (38)$$

3.4.2 Model 2: SVD and array interpolation

The error of leaving out point \mathbf{x}_i in the training set can again be derived by substituting the obtained approximation into Eq. (36) and factoring out the interpolant. The interpolation error depends on the number of basis vectors K included in the surrogate model. The error is therefore calculated between the K -rank approximation of \mathbf{y}_i as defined in Eq. (18) and the approximation from the surrogate model fitted without sample point i evaluated at sample point i , that we call $\hat{\mathbf{y}}^{-i,[K]}(\mathbf{x}_i)$. By substituting Eqs. (18) and (22) into Eq. (36) the interpolation error of the model based on SVD and array interpolation is found:

$$\begin{aligned}
 \epsilon_i^{[K]} &= \mathbf{y}_i^{[K]} - \hat{\mathbf{y}}^{-i,[K]}(\mathbf{x}_i) \\
 &= \left[\bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\Phi^{[K]} \alpha_i^{[K]} \right) \right] \\
 &\quad - \left[\bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\Phi^{[K]} \hat{\alpha}^{-i,[K]}(\mathbf{x}_i) \right) \right] \\
 &= \text{diag}(\mathbf{s})^{-1} \left(\Phi^{[K]} \left[\alpha_i^{[K]} - \hat{\alpha}^{-i,[K]}(\mathbf{x}_i) \right] \right) \\
 &= \text{diag}(\mathbf{s})^{-1} \left(\Phi^{[K]} \left[\frac{\mathbf{w}_i}{(\mathbf{G}^{-1})_{ii}} \right] \right),
 \end{aligned} \tag{39}$$

in which $\alpha_i^{[K]}$ is the column in the amplitude matrix (Eq. (15)) with K amplitudes corresponding to experiment i . Again, the errors in the different physical parts are substituted into Eqs. (26) and (27) to calculate a scalar error measure that describes the quality of the model.

For the surrogate model based on array interpolation a global scaling parameter vector θ is sought for each truncation of the basis:

$$\theta^{[K]} = \text{argmin}_{\theta} \widetilde{\text{FVU}}^{[K]} \text{ with: } K \in \{1, \dots, N_{\text{exp}} - 1\}. \tag{40}$$

3.4.3 Model 3: SVD and scalar interpolation

The interpolation error for the model based on SVD and scalar interpolation is found by substituting Eq. (19) and (25) into Eq. (36) and factoring out the interpolant. The interpolation error of the surrogate model with SVD and scalar interpolation is

$$\begin{aligned}
 \epsilon_i^n &= \mathbf{y}_i^n - \hat{\mathbf{y}}^{-i,n}(\mathbf{x}_i) \\
 &= \left[\bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\Phi_n \alpha_{ni} \right) \right] \\
 &\quad - \left[\bar{\mathbf{y}} + \text{diag}(\mathbf{s})^{-1} \left(\Phi_n \hat{\alpha}_n^{-i}(\mathbf{x}_i) \right) \right] \\
 &= \text{diag}(\mathbf{s})^{-1} \left(\Phi_n \left[\alpha_{ni} - \hat{\alpha}_n^{-i}(\mathbf{x}_i) \right] \right) \\
 &= \text{diag}(\mathbf{s})^{-1} \left(\Phi_n \left[\frac{w_{n,i}}{(\mathbf{G}^{-1})_{n,ii}} \right] \right).
 \end{aligned} \tag{41}$$

Again this error array can be used to calculate a scalar error measure in the physical domain (Eq. (27)) in order to find the global scaling parameter vector θ_n for each basis direction:

$$\theta_n = \text{argmin}_{\theta} \widetilde{\text{FVU}}^n \text{ with: } n \in \{1, \dots, N_{\text{exp}} - 1\}. \tag{42}$$

Note that for $K = 1$ in Eq. (40) and $n = 1$ in Eq. (42) the amplitude array interpolation and scalar interpolation result in the same global scaling parameters.

3.5 Choosing the number of basis vectors

The number of basis vectors in the truncated basis K can be freely chosen. With smaller K the model is cheaper to evaluate, but less accurate. With larger K the model will

be more accurate, but also more expensive to evaluate and it will require more storage space. The higher modes are dominated by noise and only contribute little to the accuracy of the surrogate model or may even deteriorate the accuracy.

To choose the number of basis vectors to be retained in the truncated basis, the ratio between the summation of K included singular values and the summation of all of them is often used [5, 12]. This ratio, referred to as the cumulative energy, is defined as:

$$C_K = \frac{\sum_{n=1}^K d_n^2}{\sum_{n=1}^{N_{\text{exp}}-1} d_n^2}. \tag{43}$$

The cumulative energy which is considered sufficient can be set and is called the cut-off ratio or threshold. For example, in the work of Hamim [12] the cut-off ratio is set to 99%, whereas in the work of Buljak [5] it is set to 99.999%. It has been shown in earlier work that the cumulative energy is not a proper indicator for the quality of the surrogate model [7].

The goal is to find the right balance between added accuracy and added error. It is therefore proposed to choose the number of basis vectors based on a ratio R_K between the interpolation error $\widetilde{\text{FVU}}^{[K]}$ and the truncation error ${}^K\text{FVU}^{[K]}$:

$$R_K = \frac{\widetilde{\text{FVU}}^{[K]}}{{}^K\text{FVU}^{[K]}}. \tag{44}$$

The interpolation error $\widetilde{\text{FVU}}^{[K]}$ is the component of the error in the subspace that is spanned by the modes 1 to K . Additional basis vectors will not reduce the interpolation error. On the other hand, the truncation error ${}^K\text{FVU}^{[K]}$ is the component of the error that is spanned by the modes $K + 1$ to $N_{\text{exp}} - 1$, and that can therefore be potentially reduced by adding more basis vectors. By choosing a threshold on the ratio R_K , no more basis vectors will be added when the potential improvement by enlarging the basis is less than R_K times the error that is already present in the model.

4 Demonstrator process: bending

To compare the different methods a demonstrator process is introduced in this section. For demonstration purposes we will investigate the bending of a metal flap. A schematic representation of the process can be found in Fig. 2. In the process a sheet metal work piece with initial sheet thickness (x_1) is bent downward to the final punch depth (x_2). Thereafter the punch is released and the workpiece will spring back.

The process is described by two input variables, the sheet thickness (x_1) and the final punch depth (x_2). A point in the design space can be denoted as: $\mathbf{x} = \{x_1, x_2\}^T$.

The process is modelled with a 2D plane strain model using the FE analysis software MSC Marc/Mentat version

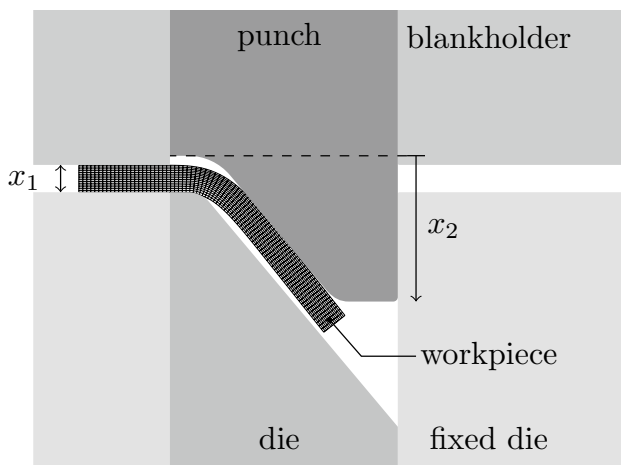


Fig. 2 Schematic representation of the demonstrator process and Finite Element mesh [7]

2016. The sheet metal is modelled with an elastic-plastic isotropic material model with Von Mises yield criterion and a tabular hardening relation between flow stress and equivalent plastic strain. The work piece is meshed using 1200 quadrilateral elements (N_{elem}) and 1296 nodes (N_{nod}). The elements are fully integrated using four integration points per element with constant dilatational strain. The solution of the FE simulation including importing data into MATLAB 2019a takes approximately 45 s.

The output of the FE-model consists of the nodal displacement field, the equivalent plastic strain field, and stress tensor field in the integration points. Hence a result of the FE simulation is collected in an $M \times 1$ result vector:

$$\mathbf{y} = \left\{ \begin{array}{c} \{\mathbf{y}_u\} \\ \{\mathbf{y}_\epsilon\} \\ \{\mathbf{y}_\sigma\} \end{array} \right\}, \quad (45)$$

With two degrees of freedom per node $M_u = 2 \cdot N_{\text{nod}} = 2592$. With four integration points per element $M_\epsilon = 4 \cdot N_{\text{elem}} = 4800$. With four integration points per element and because of the plane strain analysis there are 4 independent components of the stress tensor, $M_\sigma = 4 \cdot 4 \cdot N_{\text{elem}} = 19,200$. The overall size of a result vector is $M = M_u + M_\epsilon + M_\sigma = 26,592$.

4.1 Dimensions and range of input and output

The input and output parameters of the model have different ranges and dimensions. The nominal thickness x_1 is 0.3 mm, the thickness varies between 0.295 mm and 0.305 mm. The punch depth varies between between 1.5 mm and 1.6 mm. These ranges in input result in a variation in output. The displacement field is given in millimeter and varies between

– 1.69 mm and + 0.07 mm. The equivalent plastic strain field is dimensionless and varies between 0 and 0.48. Lastly, the stress tensor values are in MPa and vary between –661 MPa and +566 MPa.

4.2 Constructing surrogate models of the demonstrator

To obtain a training set a star point design is combined with a Latin Hypercube Sample (LHS).

To investigate the influence of the sample size on the surrogate model accuracy, four different sample sizes are used. The number of sample points in the LHS designs are 15, 35, 55 and 75 sample points. To rule out the dependency on the distribution of the sample points in the input space, five different LHS designs per sample size are used. For the demonstrator process with $N_{\text{dim}} = 2$, the starpoint design consists of $N_{\text{star}} = 2 \cdot N_{\text{dim}} + 1 = 5$ sample points. The total number of experiments in one training set will be

$$N_{\text{exp}} = N_{\text{star}} + N_{\text{lhs}} = 2 \cdot N_{\text{dim}} + 1 + N_{\text{lhs}}. \quad (46)$$

Consequently, the used sample sizes are 20, 40, 60 and 80. The designs of the different samplings are optimized by generating 250 different LHS designs and picking the 5 best based on the maximum minimum distance between sample points of the combined star-point and LHS design.

Because each sample size is replicated with five different LHS designs, the total number of FE-simulations is $5 + 5 \cdot (15 + 35 + 55 + 75) = 905$. The results of the FE-simulations will lead to a total of 20 training sets for which the global scaling parameters are optimized based on the interpolation errors as presented in Eqs. (38), (40) and (42) for direct interpolation, SVD+array interpolation and SVD+scalar interpolation respectively.

5 Results using different surrogate models

The Fraction of Variance Unexplained (FVU) as described in Sect. 3.1 is used to compare the different surrogate models after the hyperparameters have been optimized. First the models are analyzed based on the training set using the Leave-One-Out errors in Sect. 5.1. To obtain a validation set the input space is sampled with more sample points and the corresponding simulations are performed. The results based on the validation set are presented in Sect. 5.2. In Sect. 5.3 the selection of the number of basis vectors, based on the criterion in Eq. (44) is reviewed.

5.1 Results based on the training set

Figure 3 shows the different errors based on the training data as derived in Sects. 3.3 and 3.4. By including more basis

vectors in the reduced basis, the $\overset{\wedge}{FVU}$ due to truncation becomes smaller. With a full basis the $\overset{\wedge}{FVU}$ due to truncation will be 0, as the training data exist in the subspace that is spanned by the full basis.

As it should, the interpolation error with $K = 1$ is the same for the models using SVD with array and scalar interpolation. Including more basis vectors in the model increases the $\overset{\wedge}{FVU}$ due to interpolation, as the part of the result space that is spanned by the interpolation model increases with each addition of a mode. With each added mode (i.e. direction in the result space), an interpolation error (≥ 0) is added, which leads to a gradual increase of the interpolation error. The model with direct interpolation indicates the maximum $\overset{\wedge}{FVU}$ in the training set. The $\overset{\wedge}{FVU}$ of the model based on SVD with array interpolation increases towards the $\overset{\delta}{FVU}$ of the model with direct interpolation.

The $\overset{\wedge}{FVU}$ due to interpolation converges to a constant value when more basis vectors are added. If this constant value is reached, including more basis vectors does not improve the model anymore. As expected, using more sample points in the training set decreases the $\overset{\wedge}{FVU}$ due to interpolation and the $\overset{\delta}{FVU}$ of the model based on direct

interpolation. In other words, larger training sets result in better models. Using more sample points also decreases the bandwidth of the results, hence the quality of the model becomes less dependent on the sampling of the training set.

5.2 Results with a validation set

A validation set is obtained on a 9×9 grid, on a range from 10 to 90% in the normalized input parameter space of the model. To obtain the validation set an additional $N_{val} = 81$ simulations were performed. Only the centre point of the grid was included in the initial training sets. Figure 4 shows the different types of FVU based on the validation set.

The trends in the FVU in Fig. 4 are similar to the trends in FVU based on the training set in Fig. 3. The following observations made based on the error in the training set, also hold for the FVU based on the validation set. First, by including more basis vectors in the reduced basis, the $\overset{\wedge}{FVU}$ due to truncation becomes smaller. However, note that the $\overset{\wedge}{FVU}$ due to truncation does not drop to 0 when all basis vectors are included in the basis.

Fig. 3 Mean FVU calculated using the training set of surrogate models based on five different training sets constructed using a zero centered and scaled snapshot matrix. The shaded area represents the range over all training sets

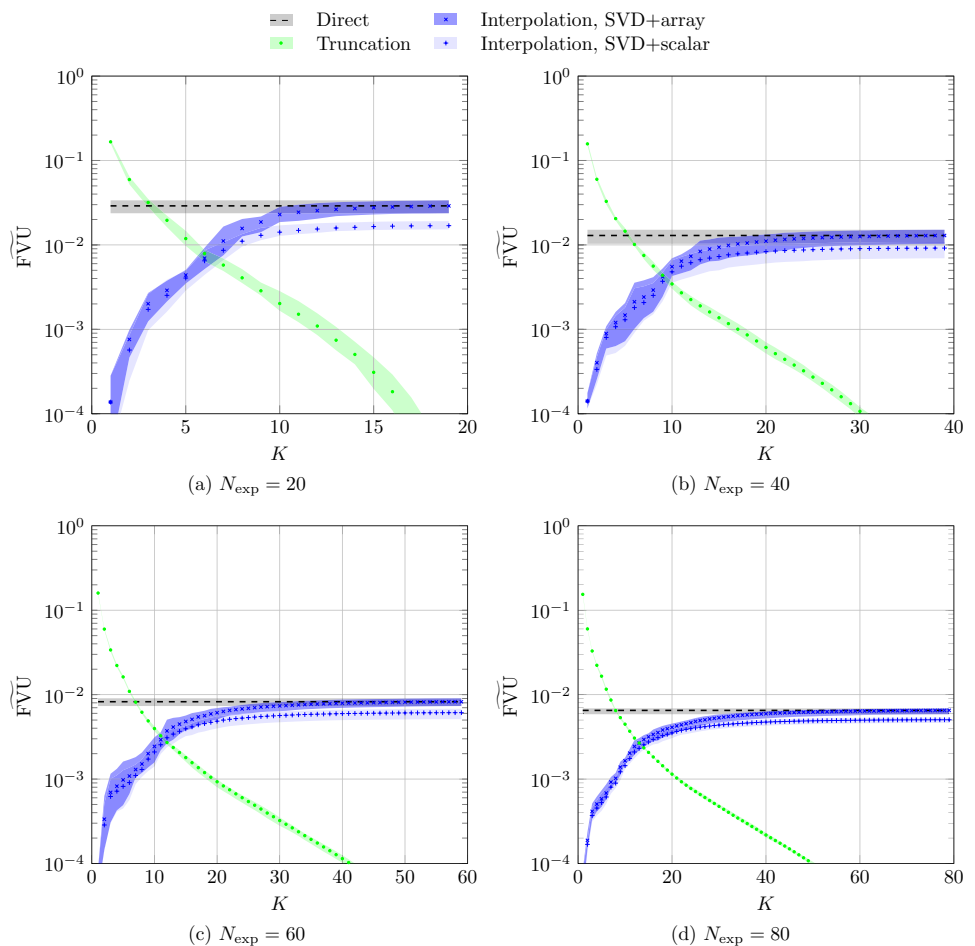
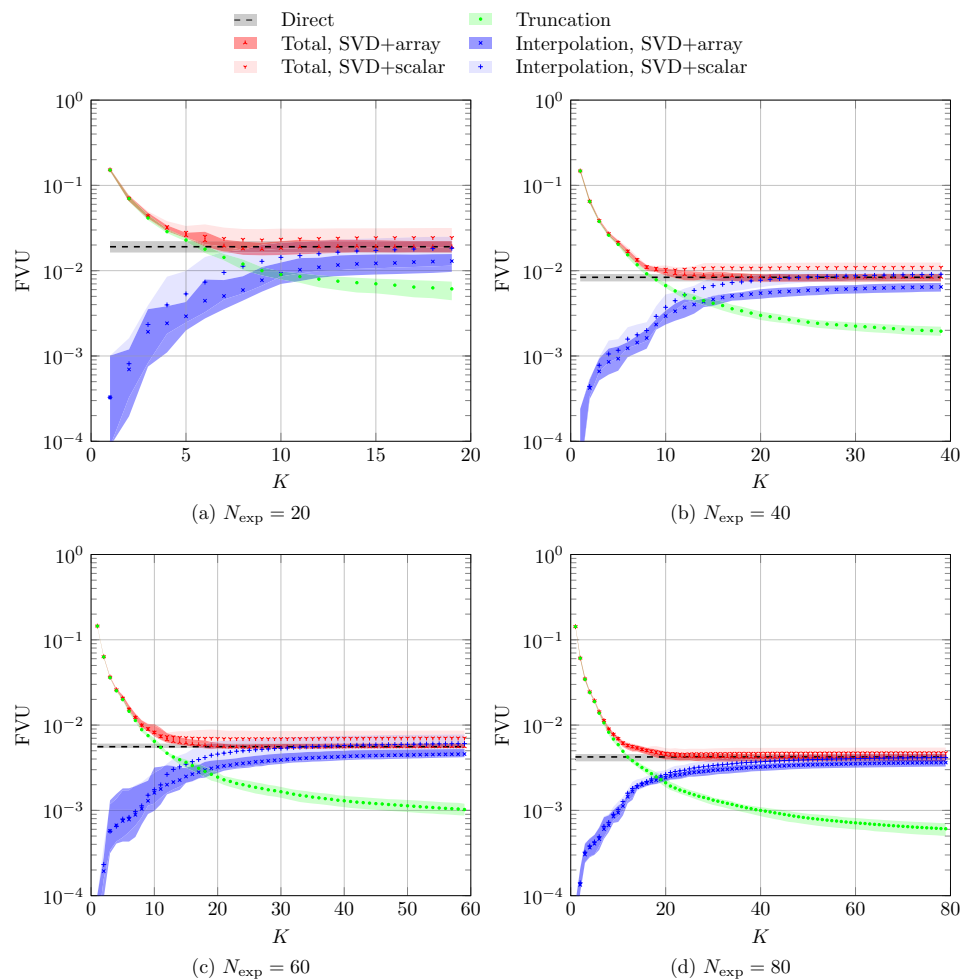


Fig. 4 Mean FVU calculated using a validation set of surrogate models based on five different training sets constructed using a zero centered and scaled snapshot matrix. The shaded area represents the range over all training sets



The truncation error calculated based on the training set gives a good approximation for the true K FVU for the first basis vectors. For higher number of basis vectors, the truncation error from the training set is an underestimation of the actual truncation error that is determined with the validation set.

Second, the interpolation error for the models using SVD with array and scalar interpolation are equal for a basis truncated to $K = 1$. Including more basis vectors in the model increases the FVU due to interpolation. The FVU due to interpolation and the total FVU converge to a constant value when more basis vectors are added. Using more sample points in the training set decreases the FVU and also decreases the bandwidth of the results.

The total FVU of the model based on SVD with array interpolation decreases towards the FVU of the model with direct interpolation. Based on the estimated FVU due to interpolation as determined with the training set, it would be expected that the SVD with scalar interpolation will perform best of all methods. However, the results from the validation data set indicate the contrary. The total FVU based on the validation set is higher, meaning less variance is explained.

This can be due to overfitting, especially in the higher modes. As found by Rao et al. [23] the increased number of hyperparameters can lead to overfitting and underestimation of the LOO error. For noisy data a general trend will perform better in predicting new data, than an overfitted metamodel. Generally the FVU due to interpolation calculated based on the training set is higher than the FVU calculated based on the validation set. Hence, the error based on the training set overestimates the error based on the validation set.

5.3 Selection of the number of basis vectors

The cumulative energy in (43) and the ratio in Eq. (44) are calculated based on the training set and will be used to determine when to truncate the reduced basis. In the top row of Fig. 5 the cumulative energy in Eq. (43) is plotted for both reduced order models. Thresholds on the cumulative energy C_K of 99.9% and 99.99% are used to truncate the reduced bases. The middle row shows the ratio in Eq. (44) for the surrogate models based on SVD with scalar and array interpolation. A threshold on the ratio R_K of 0.1 is chosen to truncate the reduced basis. With this threshold the error due

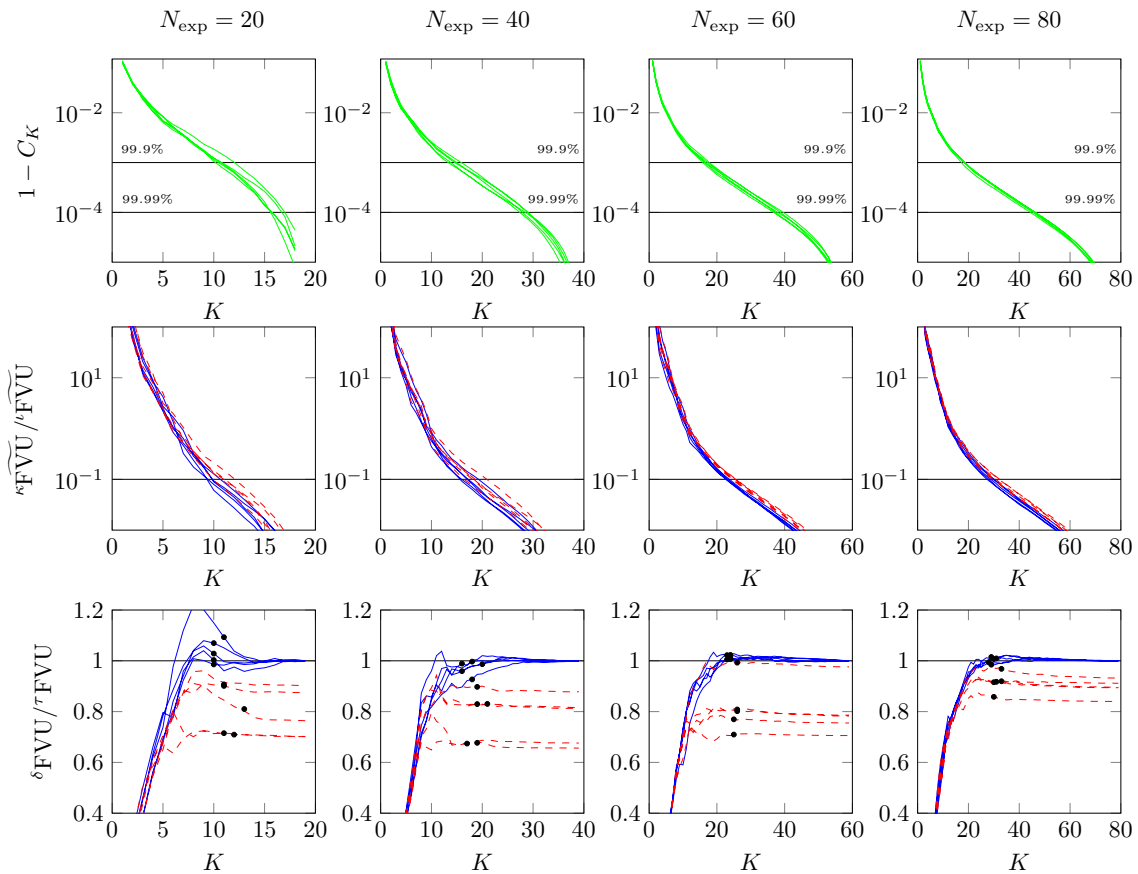


Fig. 5 Top row: cumulative energy. Mid row: ratio between the $\overline{\text{FVU}}$ due to truncation κ and interpolation ι based on the training set. $R_K = 0.1$ is indicated with a black line. Bottom row: ratio between the

FVU in the direct model δ and the total error τ in the reduced order models. Array interpolation is plotted with a blue solid line, scalar interpolation with a red dashed line

to truncation over all K included basis vectors is 10 times smaller than the error due to interpolation. The potential benefit of adding more basis vectors will be at most a reduction of 10% of the error that is already in the model due to interpolation. The models based on SVD and array interpolation reach this threshold faster than the models based on SVD and scalar interpolation

In the bottom row of Fig. 5 the ratio between the error in the direct model δ FVU based on the validation set and the total error τ FVU based on the validation set in the reduced order models is plotted. A ratio of one indicates that the direct model and the reduced order model have the same performance, and a ratio > 1 indicates that the truncated model performs better than the direct model.

The number of basis vectors K at which the model will be truncated based on the criterion that $R_K < 0.1$ is indicated with a black dot. These truncated models perform equally or slightly better than the direct interpolation models in case of array interpolation, except for the models based on the training sets with $N_{\text{exp}} = 40$. With more samples in the training

set, the difference in performance between the reduced order models with array and scalar interpolation becomes smaller.

The $\overline{\text{FVU}}$ due to interpolation based on the training set of the models with a truncated basis according to $R_K < 0.1$, $C_K > 99.9\%$ and $C_K > 99.99\%$ are presented in Fig. 6a. On average the modelling approach with SVD and scalar interpolation has the lowest interpolation error $\overline{\text{FVU}}$. This is a result of the larger number of hyperparameters that increases the flexibility in the interpolation, as a separate set of global scaling parameters θ is determined for each basis vector. In case of array interpolation, the average reduction of the data over all datasets using the criterion that $R_K < 0.1$ is 56%, which is in between the reductions with $C_K > 99.9\%$ (average reduction of 63%) and with $C_K > 99.99\%$ (average reduction of 30%).

Figure 6b displays the total τ FVU based on the validation set for the surrogate models truncated based on the criterion that $R_K < 0.1$, $C_K > 99.9\%$, $C_K > 99.99\%$ and using a full basis (the complete results can be found in Appendix 1). For all data sets, the model based on SVD+scalar interpolation has a larger error than the model

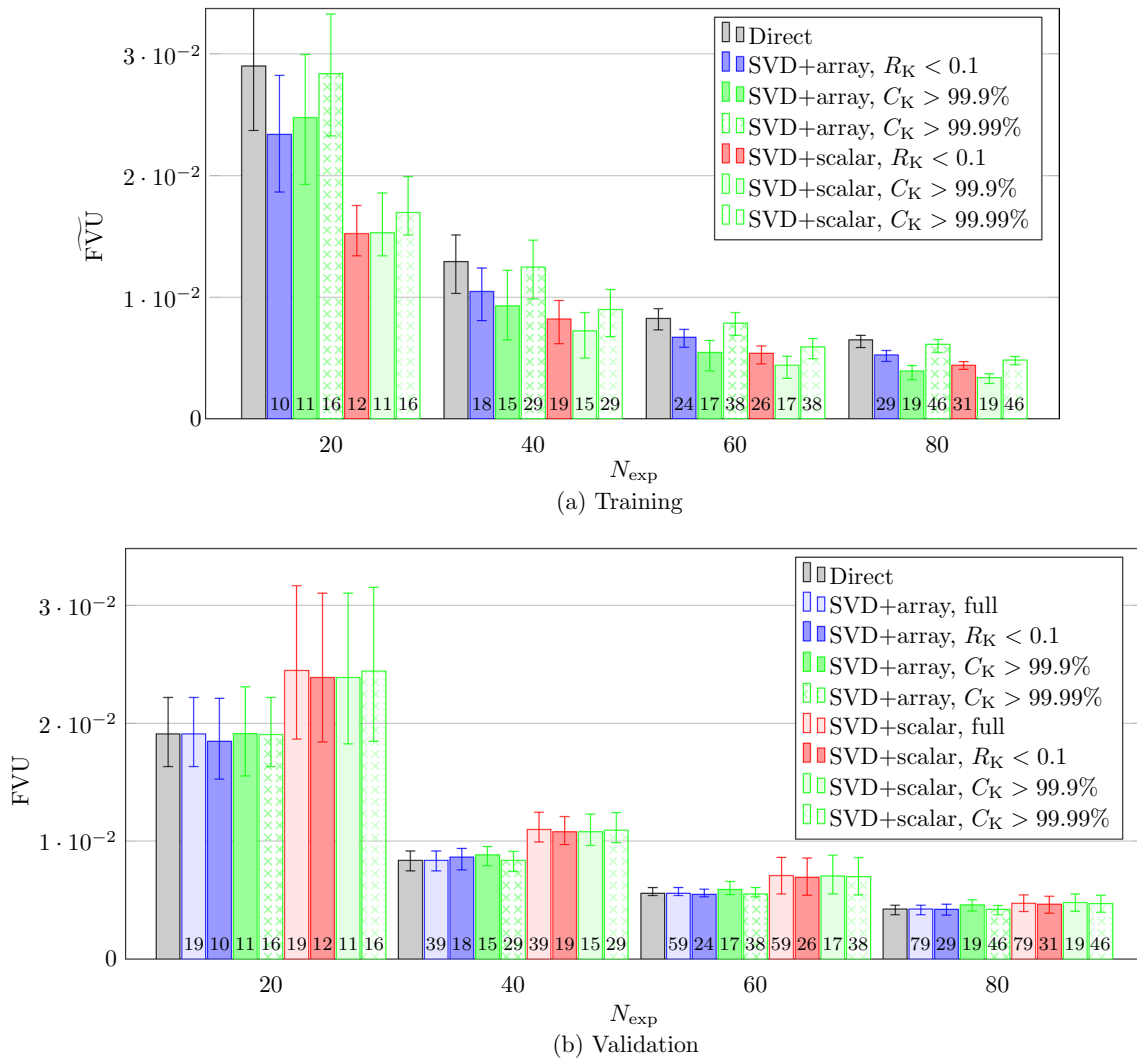


Fig. 6 Fraction of Variance Unexplained **a** due to interpolation based on the training set \widetilde{FVU} and **b** total based on the validation set rFVU .

based on direct interpolation and the models based on SVD+array interpolation, which indicates some amount of overfitting for the SVD+scalar interpolation models.

When comparing the different truncation criteria for the case of SVD+array interpolation using the validation data set, it is found that the FVU with the criterion that $R_K < 0.1$ is on average 5.5% lower than with the criterion that $C_K > 99.9\%$. In comparison to the criterion that $C_K > 99.99\%$ the FVU is on average 0.1% lower, which is no significant difference in accuracy. However, the number of used basis vectors is significantly lower for the new criterion.

6 Conclusions

The LOO-error as proposed by Rippa [24] has been successfully extended for use in error estimation of array surrogate models and SVD-based surrogate models. When comparing

the Fraction of Variance Unexplained (FVU) based on the training set (Fig. 3) and the FVU based on a validation set (Fig. 4) for a sheet bending demonstration problem that is modelled with FE, it is found that the errors based on the training set overestimate the interpolation error. The truncation error calculated based on the training set underestimates the truncation error in the validation set when many basis vectors are included, while it is a good estimate for the truncation error when few basis vectors are used.

Based on the interpolation error in the training set, the surrogate model constructed using SVD with scalar interpolation was expected to have the best performance. However, based on the validation set, this method for surrogate model construction has the largest FVU and thus the lowest performance. With larger training sets the performance of these surrogate models gets closer to the surrogate models constructed using direct interpolation and using SVD

with array interpolation. Nevertheless, it is recommended to use SVD with array interpolation.

If the amplitudes are interpolated using array interpolation, model reduction can be applied without loss of accuracy compared to using a model based on direct interpolation. The models based on SVD with array interpolation perform similar to and sometimes even better than the models based on direct interpolation, depending on where the basis is truncated.

To determine where to truncate the model, a new criterion is proposed in this work. It is shown that the ratio between the truncation error and the interpolation error in the training set (Eq. (44)) can be effectively used to balance between model reduction and accuracy. This is shown in a comparison with the commonly used cumulative energy (Eq. (43)), which only accounts for the truncation error and does not consider the interpolation

error. The FVU in the validation data set with the new criterion $R_k < 0.1$ is on average 5.5% lower than the FVU when using a contribution ratio threshold of $C_k > 99.9\%$, whereas it is comparable to the FVU when using a tighter threshold on the cumulative energy of $C_k > 99.99\%$. However, the reduction of the data set is significantly larger with the new criterion: on average 56% instead of 30% with the contribution ratio threshold of $C_k > 99.99\%$. Therefore, it is concluded that a threshold of $R_k < 0.1$ is appropriate for determining the number of basis vectors for the given problem.

Appendix 1

See Table 1.

Table 1 Different Fraction of Variance Unexplained based on the validation set using the model with direct interpolation versus the truncated models with K basis vectors

		$N_{\text{set}} =$	1	2	3	4	5	
$N_{\text{exp}} = 20$								
SVD+array	Direct	δ	1.63×10^{-2}	1.87×10^{-2}	1.67×10^{-2}	2.22×10^{-2}	2.15×10^{-2}	
	R_k	τ	1.53×10^{-2}	1.71×10^{-2}	1.69×10^{-2}	2.21×10^{-2}	2.09×10^{-2}	
	< 0.1	(K)	(10)	(11)	(10)	(10)	(10)	
	C_k	τ	1.55×10^{-2}	1.84×10^{-2}	1.70×10^{-2}	2.31×10^{-2}	2.16×10^{-2}	
	> 99.9%	(K)	(11)	(13)	(11)	(11)	(11)	
	C_k	τ	1.63×10^{-2}	1.87×10^{-2}	1.67×10^{-2}	2.22×10^{-2}	2.13×10^{-2}	
	> 99.99%	(K)	(16)	(17)	(16)	(17)	(16)	
	SVD+scalar	R_k	τ	2.30×10^{-2}	2.31×10^{-2}	1.84×10^{-2}	3.10×10^{-2}	2.39×10^{-2}
		< 0.1	(K)	(12)	(13)	(11)	(11)	(11)
		C_k	τ	2.29×10^{-2}	2.33×10^{-2}	1.83×10^{-2}	3.10×10^{-2}	2.39×10^{-2}
> 99.9%		(K)	(11)	(13)	(11)	(11)	(11)	
C_k		τ	2.32×10^{-2}	2.44×10^{-2}	1.85×10^{-2}	3.15×10^{-2}	2.45×10^{-2}	
> 99.99%		(K)	(16)	(17)	(16)	(17)	(16)	
$N_{\text{exp}} = 40$								
SVD+array	Direct	δ	7.49×10^{-3}	9.15×10^{-3}	8.17×10^{-3}	8.31×10^{-3}	8.69×10^{-3}	
	R_k	τ	7.57×10^{-3}	9.18×10^{-3}	8.52×10^{-3}	8.43×10^{-3}	9.37×10^{-3}	
	< 0.1	(K)	(16)	(18)	(16)	(20)	(18)	
	C_k	τ	7.93×10^{-3}	9.42×10^{-3}	8.60×10^{-3}	8.63×10^{-3}	9.53×10^{-3}	
	> 99.9%	(K)	(14)	(16)	(15)	(15)	(14)	
	C_k	τ	7.44×10^{-3}	9.13×10^{-3}	8.18×10^{-3}	8.26×10^{-3}	8.75×10^{-3}	
	> 99.99%	(K)	(28)	(29)	(28)	(29)	(29)	
	SVD+scalar	R_k	τ	1.11×10^{-2}	1.10×10^{-2}	1.21×10^{-2}	1.00×10^{-2}	9.68×10^{-3}
		< 0.1	(K)	(17)	(19)	(19)	(21)	(19)
		C_k	τ	1.11×10^{-2}	1.11×10^{-2}	1.23×10^{-2}	9.88×10^{-3}	9.61×10^{-3}
> 99.9%		(K)	(14)	(16)	(15)	(15)	(14)	
C_k		τ	1.10×10^{-2}	1.11×10^{-2}	1.24×10^{-2}	1.02×10^{-2}	9.85×10^{-3}	
> 99.99%		(K)	(28)	(29)	(28)	(29)	(29)	

Table 1 (continued)

		$N_{\text{set}} =$	1	2	3	4	5	
$N_{\text{exp}} = 60$								
SVD+array	Direct	δ	5.45×10 ⁻³	6.07×10 ⁻³	5.38×10 ⁻³	5.40×10 ⁻³	5.59×10 ⁻³	
	R_k	τ	5.40×10 ⁻³	5.94×10 ⁻³	5.34×10 ⁻³	5.27×10 ⁻³	5.56×10 ⁻³	
	< 0.1	(K)	(24)	(23)	(24)	(24)	(23)	
	C_k	τ	5.85 ×10 ⁻³	6.59 ×10 ⁻³	5.81 ×10 ⁻³	5.48 ×10 ⁻³	5.77 ×10 ⁻³	
	> 99.9%	(K)	(17)	(18)	(16)	(17)	(18)	
	C_k	τ	5.40 ×10 ⁻³	6.07 ×10 ⁻³	5.28×10 ⁻³	5.32 ×10 ⁻³	5.58 ×10 ⁻³	
	> 99.99%	(K)	(38)	(40)	(37)	(38)	(39)	
	SVD+scalar	R_k	τ	7.08×10 ⁻³	8.55×10 ⁻³	5.42×10 ⁻³	6.68×10 ⁻³	6.97×10 ⁻³
		< 0.1	(K)	(25)	(25)	(26)	(26)	(26)
		C_k	τ	7.14 ×10 ⁻³	8.80 ×10 ⁻³	5.52 ×10 ⁻³	6.91 ×10 ⁻³	6.92 ×10 ⁻³
		> 99.9%	(K)	(17)	(18)	(16)	(17)	(18)
		C_k	τ	7.14 ×10 ⁻³	8.56 ×10 ⁻³	5.44 ×10 ⁻³	6.81 ×10 ⁻³	7.06 ×10 ⁻³
> 99.99%		(K)	(38)	(40)	(37)	(38)	(39)	
$N_{\text{exp}} = 80$								
SVD+array	Direct	δ	4.57×10 ⁻³	4.40×10 ⁻³	4.33×10 ⁻³	4.10×10 ⁻³	3.76×10 ⁻³	
	R_k	τ	4.65×10 ⁻³	4.34×10 ⁻³	4.35×10 ⁻³	4.06×10 ⁻³	3.73×10 ⁻³	
	< 0.1	(K)	(29)	(29)	(28)	(31)	(30)	
	C_k	τ	5.04 ×10 ⁻³	4.66 ×10 ⁻³	4.76 ×10 ⁻³	4.45 ×10 ⁻³	4.07 ×10 ⁻³	
	> 99.9%	(K)	(19)	(19)	(18)	(19)	(19)	
	C_k	τ	4.54×10 ⁻³	4.36 ×10 ⁻³	4.33 ×10 ⁻³	4.05 ×10 ⁻³	3.74 ×10 ⁻³	
	> 99.99%	(K)	(46)	(47)	(45)	(47)	(46)	
	SVD+scalar	R_k	τ	5.33×10 ⁻³	4.80×10 ⁻³	4.72×10 ⁻³	4.46×10 ⁻³	3.88×10 ⁻³
		< 0.1	(K)	(30)	(31)	(30)	(33)	(33)
		C_k	τ	5.52 ×10 ⁻³	4.86 ×10 ⁻³	5.00 ×10 ⁻³	4.57 ×10 ⁻³	4.05 ×10 ⁻³
		> 99.9%	(K)	(19)	(19)	(18)	(19)	(19)
		C_k	τ	5.40 ×10 ⁻³	4.87 ×10 ⁻³	4.73 ×10 ⁻³	4.52 ×10 ⁻³	3.97 ×10 ⁻³
> 99.99%		(K)	(46)	(47)	(45)	(47)	(46)	

Best performing model for that training set is given in italics

Acknowledgements This research project is part of the ‘Region of Smart Factories’ project and is partially funded by the ‘Samenwerkingsverband Noord-Nederland (SNN), Ruimtelijk Economisch Programma’.

Declarations

Conflict of interest On behalf of all authors, the corresponding author declares that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdi H (2007) Singular and generalized singular value decomposition. In: Sps S (ed) Encyclopedia of measurement and statistics. Sage Publisher, Thousand Oaks, pp 907–912. <https://doi.org/10.4135/9781412952644.n413>
- Abdi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2(4):433–470. <https://doi.org/10.1002/wics.101>
- Behera AK, Verbert J, Lauwers B et al (2013) Tool path compensation strategies for single point incremental sheet forming using multivariate adaptive regression splines. CAD Comput Aided Design 45(3):575–590. <https://doi.org/10.1016/j.cad.2012.10.045>
- Bong HJ, Barlat F, Lee J et al (2016) Application of central composite design for optimization of two-stage forming process using ultra-thin ferritic stainless steel. Met Mater Int 22(2):276–287. <https://doi.org/10.1007/s12540-015-4325-x>
- Buljak V, Maier G (2011) Proper orthogonal decomposition and radial basis functions in material characterization based on instrumented indentation. Eng Struct 33(2):492–501. <https://doi.org/10.1016/j.engstruct.2010.11.006>

6. Dang VT, Labergère C, Lafon P (2019) Adaptive metamodel-assisted shape optimization for springback in metal forming processes. *Int J Mater Form* 12(4):535–552. <https://doi.org/10.1007/s12289-018-1433-4>
7. De Gooijer BM, Havinga J, Geijselaers HJM et al (2021) Evaluation of POD based surrogate models of fields resulting from nonlinear FEM simulations. *Adv Model Simul Eng Sci*. <https://doi.org/10.1186/s40323-021-00210-8>
8. Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1–3):50–79. <https://doi.org/10.1016/j.paerosci.2008.11.001>
9. Franke R (1982) Scattered data interpolation: tests of some methods. *Math Comput* 38(157):181. <https://doi.org/10.2307/2007474>
10. Guénot M, Lepot I, Sainvitu C et al (2013) Adaptive sampling strategies for non-intrusive POD-based surrogates. *Eng Comput Int J Comput Aided Eng Softw* 30(4):521–547. <https://doi.org/10.1108/02644401311329352>
11. Hamdaoui M, Le Quilliec G, Breikopf P et al (2014) POD surrogates for real-time multi-parametric sheet metal forming problems. *Int J Mater Form* 7(3):337–358. <https://doi.org/10.1007/s12289-013-1132-0>
12. Hamim SU, Singh RP (2017) Taguchi-based design of experiments in training POD-RBF surrogate model for inverse material modelling using nanoindentation. *Inverse Probl Sci Eng* 25(3):363–381. <https://doi.org/10.1080/17415977.2016.1161036>
13. Havinga GT, Klaseboer G, van den Boogaard AH (2016) Sequential improvement for robust optimization using an uncertainty measure for radial basis functions. *Struct Multidiscip Optim*. <https://doi.org/10.1007/s00158-016-1572-5>
14. Hu W, Enying L, Yao LG (2008) Optimization of drawbead design in sheet metal forming based on intelligent sampling by using response surface methodology. *J Mater Process Technol* 206(1–3):45–55. <https://doi.org/10.1016/j.jmatprotec.2007.12.002>
15. Huang Y, Lo ZY, Du R (2006) Minimization of the thickness variation in multi-step sheet metal stamping. *J Mater Process Technol* 177(1–3):84–86. <https://doi.org/10.1016/j.jmatprotec.2006.03.225>
16. Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13. <https://doi.org/10.1007/s00158-001-0160-4>
17. Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J Global Optim* 21(4):345–383. <https://doi.org/10.1023/A:1012771025575>
18. Kianifar MR, Campean F (2020) Performance evaluation of metamodeling methods for engineering problems: towards a practitioner guide. *Struct Multidiscip Optim* 61(1):159–186. <https://doi.org/10.1007/s00158-019-02352-1>
19. Liang Y, Lee H, Lim S et al (2002) Proper orthogonal decomposition and its applications—part I: theory. *J Sound Vib* 252(3):527–544. <https://doi.org/10.1006/jsvi.2001.4041>
20. Liew KM, Tan H, Ray T et al (2004) Optimal process design of sheet metal forming for minimum springback via an integrated neural network evolutionary algorithm. *Struct Multidiscip Optim* 26(3–4):284–294. <https://doi.org/10.1007/s00158-003-0347-y>
21. Mukhopadhyay A, Iqbal A (2009) Prediction of mechanical property of steel strips using multivariate adaptive regression splines. *J Appl Stat* 36(1):1–9. <https://doi.org/10.1080/02664760802193252>
22. Pronzato L, Müller W (2012) Design of computer experiments: space filling and beyond. *Stat Comput* 22(3):681–701
23. Rao RB, Fung G, Rosales R (2008) On the dangers of cross-validation. An experimental evaluation. In: *Society for industrial and applied mathematics—8th SIAM international conference on data mining 2008, proceedings in applied mathematics* 130(2):588–96. <https://doi.org/10.1137/1.9781611972788.54>
24. Rippa S (1999) An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv Comput Math* 11:193–210. <https://doi.org/10.1023/A:1018975909870>. [arXiv:0005074v1](https://arxiv.org/abs/0005074v1) [astro-ph]
25. Simpson TW, Poplinski JD, Koch PN et al (2001) Metamodels for computer-based engineering design: survey and recommendations. *Eng Comput* 17(2):129–150. <https://doi.org/10.1007/PL00007198>
26. Skillicorn D (2007) *Understanding complex datasets: data mining with matrix decompositions* (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series). Chapman & Hall/CRC <https://doi.org/10.1201/9781584888338>, <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20 &path=ASIN/1584888326>
27. Song SI, Park GJ (2006) Multidisciplinary optimization of an automotive door with a tailored blank. *Proc Inst Mech Eng Part D J Automob Eng* 220(D2):151–163. <https://doi.org/10.1243/095440706x72772>
28. Steffes-lai D, Turck S, Klimmek C et al (2013) An efficient knowledge based system for the prediction of the technical feasibility of sheet metal forming processes. *Key Eng Mater* 554–557:2472–2478. <https://doi.org/10.4028/www.scientific.net/KEM.554-557.2472>, <http://www.scientific.net/KEM.554-557.2472>
29. Sun G, Li G, Gong Z et al (2011) Radial basis functional model for multi-objective sheet metal forming optimization. *Eng Optim* 43(12):1351–1366. <https://doi.org/10.1080/0305215X.2011.557072>
30. Sun G, Li G, Zhou S et al (2011) Multi-fidelity optimization for sheet metal forming process. *Struct Multidiscip Optim* 44(1):111–124. <https://doi.org/10.1007/s00158-010-0596-5>
31. Tang Y, Chen J (2009) Robust design of sheet metal forming process based on adaptive importance sampling. *Struct Multidiscip Optim* 39(5):531–544. <https://doi.org/10.1007/s00158-008-0343-3>
32. Wang H, Li E, Li GY et al (2008) Development of metamodeling based optimization system for high nonlinear engineering problems. *Adv Eng Softw* 39(8):629–645. <https://doi.org/10.1016/j.advengsoft.2007.10.001>
33. Wang H, Li E, Li GY (2009) Development of hybrid fuzzy regression-based metamodeling technique for optimization of sheet metal forming problems. *Mater Des* 30(8):2854–2866. <https://doi.org/10.1016/j.matdes.2009.01.015>
34. Wang H, Li E, Li GY (2009) The least square support vector regression coupled with parallel sampling scheme metamodeling technique and application in sheet forming optimization. *Mater Des* 30(5):1468–1479. <https://doi.org/10.1016/j.matdes.2008.08.014>
35. Wang H, Li E, Li GY (2011) Probability-based least square support vector regression metamodeling technique for crashworthiness optimization problems. *Comput Mech* 47(3):251–263. <https://doi.org/10.1007/s00466-010-0532-y>
36. Wei D, Cui Z, Chen J (2008) Optimization and tolerance prediction of sheet metal forming process using response surface model. *Comput Mater Sci* 42(2):228–233. <https://doi.org/10.1016/j.commatsci.2007.07.014>
37. Wiebenga JH, van den Boogaard AH, Klaseboer G (2012) Sequential robust optimization of a V-bending process using numerical simulations. *Struct Multidiscip Optim* 46(1):137–153. <https://doi.org/10.1007/s00158-012-0761-0>
38. Zhao D, Xue D (2010) A comparative study of metamodeling methods considering sample quality merits. *Struct Multidiscip Optim* 42(6):923–938. <https://doi.org/10.1007/s00158-010-0529-3>