Int J Mater Form
DOI 10.1007/s12289-013-1130-2

ORIGINAL RESEARCH

On the effect of numerical noise in approximate optimization of forming processes using numerical simulations

J. H. Wiebenga · A. H. van den Boogaard

Received: 12 January 2013 / Accepted: 12 March 2013
© Springer-Verlag France 2013

Abstract The coupling of Finite Element (FE) simulations with approximate optimization techniques is becoming increasingly popular in forming industry. By doing so, it is implicitly assumed that the optimization objective and possible constraints are smooth functions of the design variables and, in case of robust optimization, design and noise variables. However, non-linear FE simulations are known to introduce numerical noise caused by the discrete nature of the simulation algorithms, e.g. errors caused by remeshing, time-step adjustments or contact algorithms. The subsequent usage of metamodels based on such noisy data reduces the prediction quality of the optimization routine and is known to even magnify the numerical errors. This work provides an approach to handle noisy numerical data in approximate optimization of forming processes, covering several fundamental research questions in dealing with numerical noise. First, the deteriorating effect of numerical noise on the prediction quality of several well-known metamodeling techniques is demonstrated using an analytical test function. Next, numerical noise is quantified and its effect is minimized by the application of local approximation and regularization techniques. A general approximate optimization strategy is subsequently presented and coupling with a sequential update algorithm is proposed. The strategy is demonstrated by the sequential deterministic and robust optimization of 2 industrial metal forming processes i.e. a V-bending application and a cup-stretching application. Although numerical noise is often neglected in practice, both applications in this work show that the general awareness of its presence is highly important to increase the overall accuracy of optimization results.

Keywords Finite element simulations · Approximate optimization · Numerical noise · Regularization · Sequential optimization

J. H. Wiebenga (✉)
Materials innovation institute (M2i),
P.O. Box 5008 2600 GA, Delft, the Netherlands
e-mail: J.Wiebenga@m2i.nl

A. H. van den Boogaard
University of Twente Faculty of Engineering Technology,
P.O. Box 217 7500 AE, Enschede,the Netherlands
e-mail: A.H.vandenBoogaard@UTwente.nl

Introduction

Decisions in a large number of optimization problems in the metal forming industry are made by use of Finite Element (FE) simulations coupled with a suitable mathematical optimization algorithm. These so-called simulation-based optimization approaches have proven to be much more efficient than the conventional trial-and-error processes. Moreover, they have contributed significantly in fulfilling the continuously increases technical and economical requirements.

Approximate optimization is an often used and well-known approach to couple computationally expensive FE simulations with an optimization procedure. An overview of metamodeling applications in structural optimization can be found in Barthelemy and Haftka [1] and more recently in Simpson et al. [2]. Using these cheap surrogates, optimization can be performed more efficiently by reducing the number of required FE simulations.

An increasing variety of optimization problems in the metal forming industry are being solved by approximate optimization, e.g. design optimization problems [3, 4], multi-objective optimization problems [5, 6], Robust Design

Published online: 07 April 2013

Optimization (RDO) problems [7, 8] and Reliability Based Design Optimization (RBDO) problems [9, 10]. In case of RDO and RBDO, approximations of the statistical measures of the objective function and constraints are generated by applying e.g. Monte Carlo sampling on the metamodel. This is feasible since the surrogate models can be evaluated very efficiently.

The use of metamodels in an optimization approach hinges on the assumption that the metamodel is a correct representation of the FE calculations and that the FE simulation correctly predicts the physical process. However, it is well known that this is often not the case and that different sources of errors are present. Oden et al. [11] state that quantification of these errors is required to increase the reliability and utility of simulation methods in the future. Simpson et al. [2] recognize such errors as a major issue to be solved in future research on approximate optimization, especially with the continuous increase of the computational complexity of FE simulations. The objective of this work is to initiate this research.

Notorious sources of numerical noise are discretization errors, adaptive mesh refinement, automatic (adaptive) step-size selection, round-off errors or changing contact conditions [12]. These numerical errors often appear as fluctuations around an expected smooth response. Achieving a good compromise between accurate simulation results (by reducing these errors) and optimization results on the one hand and computational efficiency on the other hand is an on-going challenge. Especially in the metal forming industry where computationally expensive non-linear simulations are being used. An increasing number of commercial FE codes offer numerical tools such as automatic procedures for re-meshing or mesh refinement to answer to this need. Although these techniques have proven their value for a single deterministic simulation, it is demonstrated in e.g. van Keulen and Toropov [13] that they may cause increasing numerical errors when used in multiple simulations.

To demonstrate numerical noise, consider Fig. 1 showing results for 200 FE simulations of a V-bending application. The bending angle (θ) is plotted as a function of a change in tooling geometry (L). This V-bending application will be discussed in more detail in Section "Application 1: V-bending process". The FE simulation response clearly shows deviations from the expected smooth numerical response. These fluctuations are referred to as numerical noise in this paper.

Several approaches have been proposed in literature which account for the presence of numerical noise [14–16]. These approaches make use of Response Surface Methodology (RSM) approximations to filter out numerical noise. Although very suitable for this purpose, these models also show low flexibility with respect to possible strongly non-linear response functions in the design space. As an example, note the bend in the response at $L = 4.7$ mm in Fig. 1 which cannot be accurately described by a linear or quadratic RSM model. Moreover, Oden et al. [11] already mentioned that many naturally arising research questions regarding numerical noise remain open, i.e. how to minimize its deteriorating effect on prediction quality and how to quantify numerical noise.

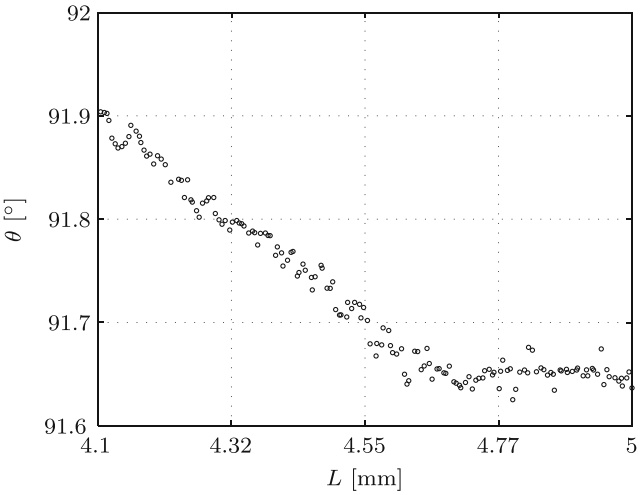

Fig. 1 FE simulation response of a V-bending application demonstrating the presence of numerical noise

A general approach is therefore presented in this work to handle and minimize the deteriorating effect of noisy response data in approximate optimization, enabling the use of any metamodeling technique. Both deterministic and robust optimization problems are considered. For the latter, the effect of numerical noise on the prediction of statistical measures is considered. The above mentioned research questions will be covered and a coupling is made with a sequential optimization algorithm based on Expected Improvement (EI). It is well known that the metamodel quality can be improved by sequentially adding sampling points, but the quality improvement is limited in case numerical noise is present. Instead of relying solely on an arbitrary chosen threshold value to terminate the EI algorithm, an algorithm is presented based on the magnitude of noise present in the response.

The paper is organized as follows, Section "Effect of numerical noise" will first demonstrate the deteriorating effect of numerical noise on the metamodel prediction accuracy using an analytical test function. The numerical noise is subsequently quantified and its effect is minimized by the application of local approximation and regularization techniques respectively, see Section "Accounting for numericalnoise". These techniques are subsequently incorporated in a general approximate optimization strategy

and a coupling is made with an update algorithm in Section "Approximate optimization strategy". This approach is applied in the sequential optimization of 2 industrial metal forming processes in Sections "Application 1: V-bending process" and "Application 2: Cup-stretching process". The former application is a deterministic optimization problem whereas the latter application is a robust optimization problem. Finally, Section "Conclusions" will present the conclusions and directions for future work.

Effect of numerical noise

Computer simulations are deterministic in nature meaning that repeated runs for the same input parameters will yield exactly the same result [17–19]. However, a small deviation in the variable selection can cause the FE code to have a different mesh or different number of increments. Consequently, this may give a relatively large difference in response which is for the greater part a numerical artifact rather than a physical response change. Other causes of noisy or perturbed behavior of objective functions in optimization problems are demonstrated in Giunta et al. [14] and Toropov et al. [20]. It is shown in Siem and den Hertog [21] that the subsequent usage of metamodels to approximate the noisy response may result in a low quality response approximation and may even magnify the prediction error. Combining the resulting metamodel with any optimization algorithm will provide the user with an inaccurate prediction of the optimum.

Bias-Variance trade-off

Following the approximate optimization approach, the metamodel prediction $\hat{y}(x)$ of the simulation-based response $y(x)$ can be formulated as:

$$y(x) = \hat{y}(x) + \varepsilon \tag{1}$$

When considering deterministic computer experiments, the remaining fitting error ε should ideally be zero in the training points. This corresponds to a zero *bias error* which means that the metamodel should interpolate through the response values at the training points. The bias error can be decreased by increasing the complexity of the metamodel (e.g. increasing the order of polynomial) or use interpolating metamodels. However, the prediction accuracy of such models deteriorate when dealing with noisy data since decreasing the bias error will tend to provide a larger *variance error* as a result of over fitting. The variance error is the variation of the metamodel prediction for different subsets of DOE points extracted from the full training set. The variance error can again be decreased by smoothing the metamodel, but if this idea is taken too far then the bias error will increase again as a result of under fitting. This natural trade-off is referred to as the *bias-variance trade-off* [22].

In addition, both the bias and variance error can be decreased by increasing the number of training points. However, when using e.g. computationally expensive FE simulations, the number of points is severely limited. As long as data is sparse, i.e. the number of Design Of Experiment (DOE) points is small, interpolating models are still able to generate a smooth approximation of the noisy data. However, metamodels based on sparse data will generally only serve as a first estimate of the true response function. Sequential optimization steps are subsequently applied to efficiently increase the accuracy of the response prediction at regions of interest containing the optimal design [23, 24]. By doing so, data become more dense in a local region of the design space which may again result in erroneous approximate predictions if the calculated response data are contaminated with noise.

Application to an analytical test function

To demonstrate these phenomena and the effect of noise, consider the following analytical test function:

$$\begin{aligned} y(x) &= (6x-2)^2 \sin(12x-4) + \varepsilon \\ \varepsilon &\sim \mathcal{N}\left(\mu_\varepsilon, \sigma_\varepsilon^2\right) = \mathcal{N}(0, 1^2) \end{aligned} \tag{2}$$

Two data sets are considered, a smooth and perturbed data set. For the latter, the analytical test function is perturbed with a normally distributed noise (ε) with mean $\mu_\varepsilon = 0$ and constant standard deviation $\sigma_\varepsilon = 1$ where the magnitude of the perturbation is assumed small compared to the overall change in response behavior. Both data sets contain 20 DOE points, 10 globally distributed points augmented with 10 local DOE points added near the global optimum

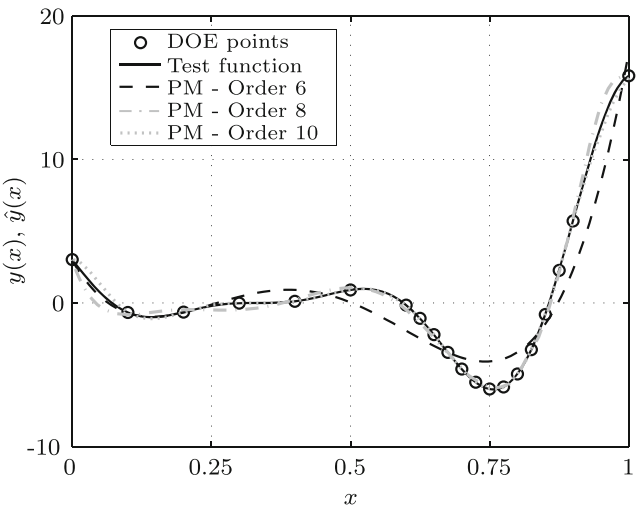

Fig. 2 Polynomial Model (PM) approximation of the smooth data set

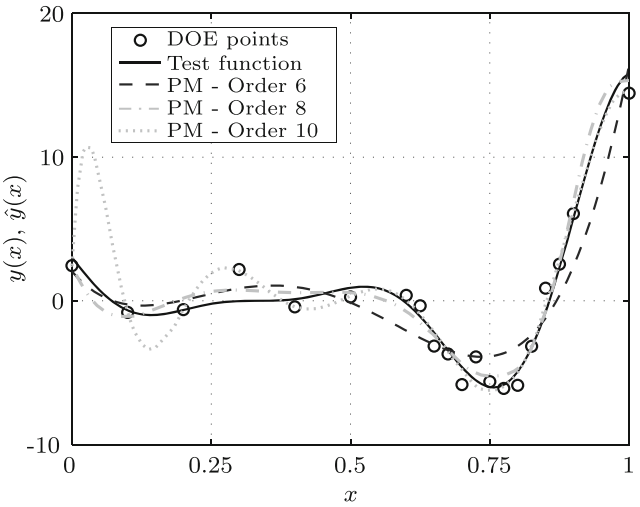

Fig. 3 Polynomial Model (PM) approximation of the perturbed data set

of the test function to create a more dense sampling such as obtained by sequential approximate optimization. A family of Polynomial Models (PM) with an order up to 10 is fitted based on the commonly used quadratic loss function or L_2-norm, see Section "Response surface methodology" in Appendix. Application of this loss function is referred to in this work as the *basic* form. The deviation of the metamodel $\hat{y}(x)$ from the test function $y(x)$ is determined using the Root Mean Squared Error (RMSE) evaluated at n points x_i:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}(x_i) - y_i)^2} \tag{3}$$

The results for the smooth and perturbed data set are presented in Figs. 2 and 3 respectively. The resulting error estimates for the smooth and perturbed data set are given in Table 1 and denoted by RMSE_S and RMSE_P respectively. In case no noise is present in the data, increasing the order of the polynomial will decrease the RMSE_S. The best fit for the smooth data set is found to be a polynomial fit of order 10. Now for the perturbed data set, the best approximation of the test function is found to be a polynomial fit of order 8, see Table 1. Decreasing the order to 6 will result in a poor representation of the test function and a high bias error

Table 1 Error estimates of the Polynomial Model (PM) approximation in basic and regularized form of the smooth and perturbed data set

Metamodel form	RMSE_S basic	RMSE_P basic	RMSE_P regularized
PM - Order 6	10.9	11.8	11.2
PM - Order 8	5.0	8.6	7.3
PM - Order 10	1.8	22.1	12.2

(especially near the optimum), see Fig. 3. On the other hand, increasing the order of the polynomial fit to 10 will decrease the bias error but increase the variance error. This can be recognized in Fig. 3 by the over fitting of the polynomial model at the sparsely sampled regions of the design space. The increasing flexibility of the model is becoming more tuned to the noise present in the data. For the perturbed data set, the polynomial of order 8 has the best *generalization capability* providing an optimal trade-off between closeness to the data and smoothness of the function.

Accounting for numerical noise

Next, two fundamental research questions considering numerical noise will be covered: how to minimize its deteriorating effect on prediction quality and how to quantify noise. The former question requires finding the optimal balance between bias and variance errors in the construction of metamodels without *a priori* knowledge of the underlying true response function. This can be achieved by regularization techniques and will be discussed in Section "Regularization". An approach for estimating the magnitude of noise will be presented in Section "Quantification of noise".

Regularization

In case of regularization, the L_2-norm is extended with an additional term [25]. In this work, *ridge regression*, also known as *Tikhonov regularization* [26], is applied where the best approximate model from the family H of approximate models is selected as the solution of:

$$\min_{\hat{y}\in \text{H}} \left(\mathbb{Z}(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}(x_i) - y_i)^2 + \lambda \int \|D^2\hat{y}(x)\|_\text{H} \text{d}x \right) \tag{4}$$

where λ is the regularization parameter and $D^2\hat{y}(x)$ represents the value of the second derivative of the proposed model. The first term includes the quadratic loss function which enforces closeness to the data or a low bias error. The second term penalizes high local curvature enforcing a smooth approximate model or low variance error. The optimal regularization parameter can be identified by minimizing the Generalized Cross-Validation (GCV) function [27]. This function is based on leaving out an arbitrary element y_i after which the corresponding regularized solution $\hat{y}_\lambda$ should predict this observation well. The regularization parameter is chosen that minimizes the GCV function [28]:

$$\text{GCV}(\lambda) = \frac{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_\lambda(x_i) - y_i)^2}{(1 - tr\ A(\lambda)/n)^2} \tag{5}$$

where $A(\lambda)$ is the matrix which produces the regularized solution when multiplied with y, i.e. $\hat{y}_\lambda = A(\lambda)y$.

A description of the regularized fitting procedure for the family of polynomial models is provided in Section "Response surface methodology" in Appendix. The $RMSE_P$ results for these models considering the perturbed data set is presented in the fourth column of Table 1. For all polynomial models, a closer representation of the test function is obtained by application of regularization, represented by the decreasing $RMSE_P$ values. Also note that for the polynomial model of order 10 the over fitting is significantly suppressed. Again, the polynomial model of order 8 results in the lowest $RMSE_P$. The GCV function for this model is plotted in Fig. 4 from which the optimal regularization parameter $\lambda = 0.1$ can be identified. Application of other well known (regularized) metamodel types to the analytical test function will be demonstrated in Section "Approximate optimization strategy".

Quantification of noise

Quantification of noise requires determining the deviation between response measurements and the expected smooth, but unknown, underlying true function. Therefore, a low complexity approximation of the true function is created by smoothing. Each response measurement is replaced by a local average of surrounding measurements such that the level of noise is reduced without (much) biasing the value obtained. In this work, a local smooth approximation is created based on Least-Squares Smoothing (LSS) after the technique used in *Savitzky-Golay* filters [29]. Applying a local smooth approximation benefits from a more densely sampled design space obtained by sequential optimization used to accurately predict a local estimator and thus result in a low bias error. Moreover, the true function is assumed to be low multimodal in a local region of the design space. One could argue to use the metamodel approximation itself, but this assumes that the mean of the true function is locally estimated accurately. This assumption can be violated in practice due to a variety of reasons, e.g. the existence of a local bias error in the metamodel due to the global metamodel fitting procedure.

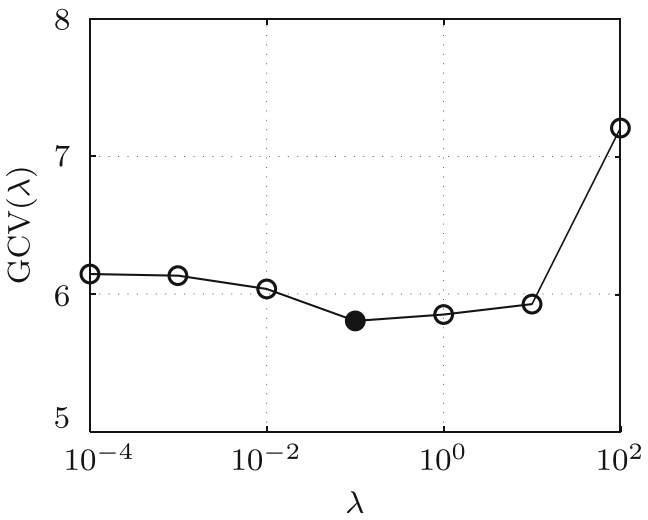

Fig. 4 Generalized Cross Validation (GCV) function for the regularized polynomial model of order 8

In the LSS approach, a polynomial least–squares fit is created inside a moving window [25]. The subsets of DOE points are determined by a nearest neighbors algorithm using an ellipsoid search region for a multi–dimensional moving window. In case of a 1 dimensional moving window, this reduces to finding an equal number of points to the left (n_l) and to the right (n_r) of the DOE point where $n_l = n_r = 2$. By utilizing a moving window, it is assumed that relatively distant DOE points have some redundancy which can be used to reduce the level of noise and that the true function can be locally well-fitted by a polynomial function. To account for the irregularly sampled data, a LSS prediction $\hat{y}_{lss}$ is made based on a quadratic polynomial approximation (see in Section "Response surface methodology" in Appendix) :

$$\hat{y}_{lss} = \mathbf{X}\beta \tag{6}$$

where $\mathbf{X}$ is the design matrix containing the quadratic basis functions and β are the regression coefficients. Note that the method of least squares chooses the β's such that the L_2-norm is minimized considering the points that are within the moving window. For the 1D moving window, this results in:

$$L_2 = \sum_{k=-n_l}^{n_r} \varepsilon_{i+k}^2 = \sum_{k=-n_l}^{n_r} (\hat{y}_{lss}(x_{i+k}) - y_{i+k})^2 \tag{7}$$

This procedure is repeated for each point x_i. The resulting LSS model is a more accurate description of the true function than the scattered data points. Assuming the error variance (σ_ε^2) to be constant and the errors to be approximately normally distributed, an estimate of the magnitude of noise ($\hat{\sigma}_\varepsilon$) can be obtained using the smooth function by taking the the variance of the estimated residuals ($\hat{\varepsilon}_i$) evaluated at all x_i where:

$$\hat{\varepsilon}_i = y_i - \hat{y}_{lss}(x_i) \tag{8}$$

The local LSS model of the perturbed data set is provided in Fig. 5. The smoothing is applied in a local region with an increased sample density, i.e. $0.6 \leq x \leq 0.9$. Evaluating the magnitude of numerical noise through Eq. 8 results in $\hat{\sigma}_\varepsilon = 0.86$. In case $\hat{y}_{lss}(x_i)$ is replaced by the function $y(x)$ in Eq. 8, the magnitude of noise is determined at $\sigma_\varepsilon = 0.89$. Clearly, the local LSS model results in an accurate estimation of the noise present in the perturbed data set. Based on

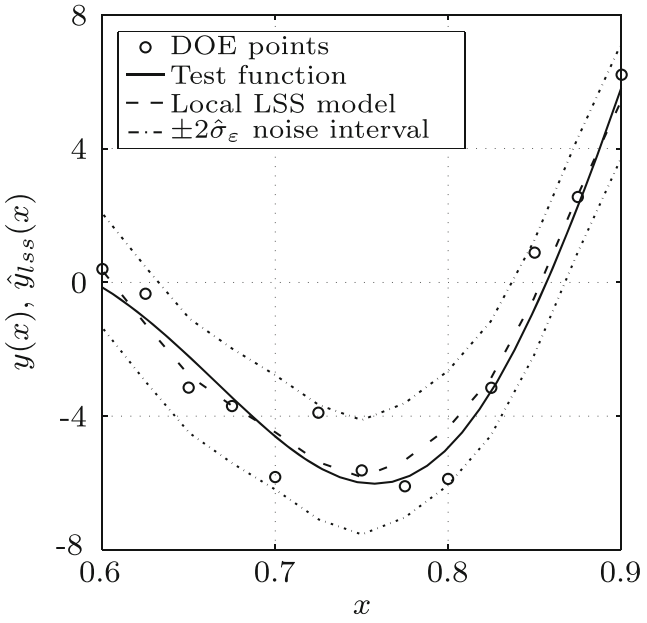

Fig. 5 Local Least-Squares Smoothing (LSS) model of the perturbed data set and accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval

this estimation, one can plot the $\pm 2\hat{\sigma}_\varepsilon$ noise interval around the local LSS model, see Fig. 5.

Approximate optimization strategy

This work is centered around application of approximate models in optimization strategies. The first step in approximate optimization is the definition of an objective function (f) and possible constraints (**g**) which are functions of the design variables (**x**). In case of robust optimization, also noise variables (**z**) are present. Note the difference here between noise variables and numerical noise. The former are physical parameters whereas the latter is a numerical artifact. In case numerical noise is present in the response function, it is shown that application of regularization and local approximation allows the user to minimize the deteriorating effects of numerical noise and quantify its magnitude. Next, these techniques are incorporated in a general approximate optimization strategy and a coupling is made with a sequential robust optimization algorithm. The strategy will be demonstrated by application to the analytical test function and sequential optimization of 2 metal forming processes in Sections "Application 1: V-bending process" and "Application 2: Cup-stretching process".

Data sampling

At the basis of any approximate model or metamodel is a DOE plan. In this work, a DOE is created based on a space-filling Latin Hypercube Design (LHD) combined with a Full Factorial Design (FFD). This ensures that both the interior and the outer boundaries of the design space are well sampled. Since the non-linear simulations are very time-consuming, a minimum number of simulations is preferred. The number of DOE points is chosen equal to 10 times the number of variables as recommended in [30].

As mentioned in Section "Effect of numerical noise", metamodels based on sparse data will generally only serve as a first estimate of the FE simulation response. Sequential optimization steps are applied to efficiently increase the accuracy of the response prediction at regions of interest containing the optimal design. This is simulated for the analytical test function by supplementing the initial DOE of 10 points with an additional 10 DOE points near the global optimum. It is shown that problems occur if the density of data points varies in the design space. The noise present in the data especially causes over fitting of the approximate model in the sparsely sampled regions of the design space. The update algorithm used in this work will be discussed in more detail in Section "Optimization and update algorithm".

Training and evaluation

For research purposes, multiple metamodel types are used for creating a family of approximate models. The shape and complexity of the response behavior of the objective function and constraints in the whole design space is unknown beforehand. Next to the polynomial models, the following metamodel types are considered in this work:

- Kriging
- Radial Basis Functions (RBF)
- Neural Networks (NN)

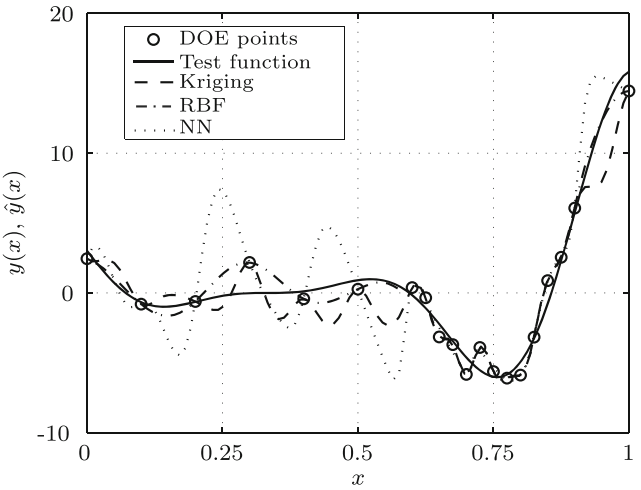

Fig. 6 Basic metamodel approximation of the perturbed data set

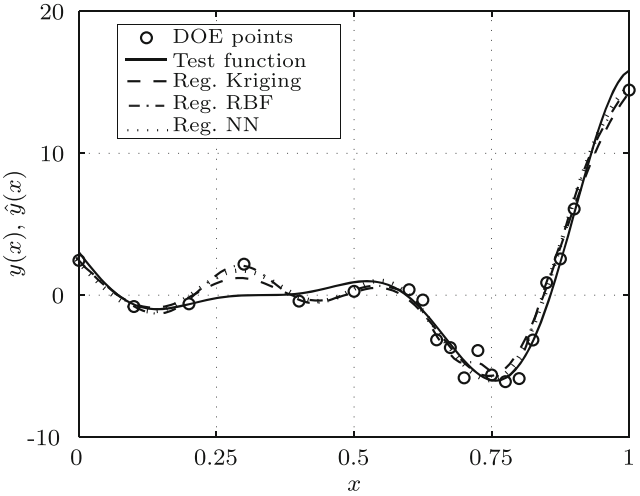

Fig. 7 Regularized metamodel approximation of the perturbed data set

In the basic form, these models are fitted based on the L_2-norm and show interpolating behavior. Similar to polynomial models, the loss function can be replaced by a regularized form to account for numerical noise. A description of the different types of metamodeling techniques used in this work, both in basic and regularized form, is provided in Section "Appendix: Metamodel types". Essentially, one can choose any type of metamodel to be used in this procedure as long as the fitting quality for each response is sufficient for use in approximate optimization. This will be discussed in more detail in Section "Model selection and validation".

Application of the above mentioned metamodel types in basic and regularized form to the perturbed data set is presented in Figs. 6 and 7 respectively. Studying Fig. 6 in more detail, one can recognize the over fitting behavior of the models caused by forcing the models to interpolate through the perturbed data points. Figure 8a shows a detail plot near the global optimum from which the interpolating behavior becomes even more apparent. Combining these metamodels with an optimization algorithm will likely result in erroneous results due to the early stalling of the optimization algorithm in a local optimum. Also note that some additional local optima appear following from the over fitting behavior of the polynomial model.

The problem of approximating perturbed response data with a varying sample density in the design space is largely resolved by allowing the models to regress the data, see Fig. 7 and the detail plot in Fig. 8b. Note that both the optimal value for the objective function and design variable can be accurately predicted using the regularized metamodels. The significant improvement of approximation accuracy with respect to the test function is also confirmed by the reduction of the RMSE_P for all metamodels, see Table 2.

Model selection and validation

Until now, the quality of the metamodel approximation is determined using the response function $y(x)$ in Eq. 3. In case the function is unknown, one can resort to *Cross Validation* (CV) where the ability of a metamodel to predict untried measurements is tested. Using CV, one leaves out a measurement point x_{-i} and fits the metamodel through the remaining response measurements. Analogous to Eq. 3, one can determine a Cross Validation Root Mean Squared Error (RMSE_CV) by replacing $\hat{y}(x_i)$ with the metamodel prediction $\hat{y}_{-i}$ at x_{-i}. Leave One Out Cross Validation (LOOCV) is applied for all metamodel types present in the family of metamodels, both in basic form and regularized form. For each response, the metamodel type is chosen that minimizes the RMSE_CV.

Table 3 shows the resulting RMSE_CV values for all metamodels in basic and regularized form. Similar to the RMSE_P values as presented in Table 2, the improvement of the approximation accuracy is predicted based on the CV results, although these values are now based on the (perturbed) response measurements instead of the known test function. Model selection is based on the lowest RMSE_CV value, i.e. NN. This example shows that CV can be used to validate metamodel accuracy in case noise is present in the response measurements.

Table 2 Error estimates of the metamodel approximation of the perturbed data set

Metamodel form	RMSE_P basic	RMSE_P regularized
Polynomial - Order 8	8.6	7.3
Kriging	15.2	6.7
Radial Basis Function	9.0	7.0
Neural Network	28.3	6.5

Table 3 Error estimates of the metamodel approximation of the perturbed data set obtained by Cross Validation (CV)

Metamodel form	RMSE_CV basic	RMSE_CV regularized
Polynomial - Order 8	3.9	2.5
Kriging	4.9	1.4
Radial Basis Function	3.3	1.1
Neural Network	5.8	0.9

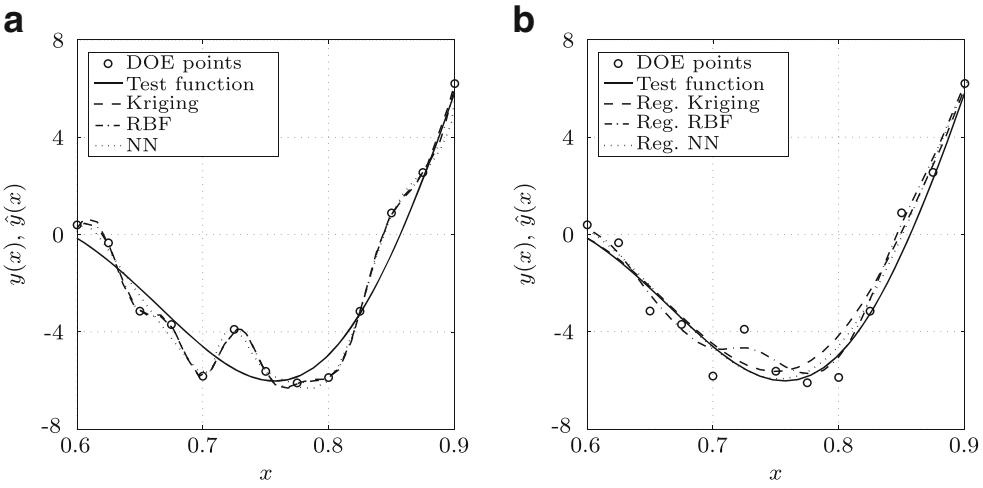

Fig. 8 Detail plot of the (**a**) basic and (**b**) regularized metamodel approximation of the perturbed data set

Optimization and update algorithm

The regularized metamodels can now be used for optimization by application of an optimization algorithm. To obtain an accurate and reliable solution, the optimal metamodel prediction has to be validated by performing FE simulations in the optimum. If the accuracy is not sufficient according to the design engineer, a sequential improvement step can be applied to update the metamodel successively.

An update algorithm is required to select the location of the next DOE points. In this work, the location of the newly added DOE points are based on maximizing the Expected Improvement (EI) [17]. The expected value of the improvement can be expressed in a closed form given by Jones et al. [18]:

$$\mathrm{E}(I) = \begin{cases} \left(f_{\min} - \hat{y}(x)\right)\Phi\left(\frac{f_{\min}-\hat{y}(x)}{s(x)}\right) + s(x)\phi\left(\frac{f_{\min}-\hat{y}(x)}{s(x)}\right) & \text{if } s>0 \\ 0 & \text{if } s=0 \end{cases} \quad (9)$$

with ϕ and Φ the probability density and cumulative distribution function of the standard normal distribution respectively. The improvement is calculated by taking the difference between the mean value prediction $\hat{y}(x)$ with respect to the minimum feasible calculated response of the true objective function value $f_{\min}$ if $y < f_{\min}$. In addition to the mean value prediction $\hat{y}(x_0)$ at any location x_0, the EI algorithm makes use of the *prediction error* $s(x_0)$ provided by the metamodel. For RSM, Kriging, RBF and NN, the estimate of the prediction error $\hat{s}$ equals the square root of Eqs. 17, 27, 33, and 40 respectively. In other words, the predictor $\hat{y}(x_0)$ represents a realization of a stochastic process Y in which the randomness is governed by the uncertainty $s(x_0)$ about the true objective function [17, 19]. Maximizing the EI finally provides the coordinates of the infill point x'. After evaluating the new infill point by a FE simulation, the metamodel is globally updated and validated taking into account the additional response. An optimization procedure is started and the new prediction of the optimum is determined.

The termination of the update algorithm is generally based on an arbitrarily chosen threshold for the EI. In case no numerical noise is present in the response data, a fast decline of the EI function is generally obtained by adding DOE points [24]. However, the quality improvement is limited in case numerical noise is present resulting in an inefficient termination algorithm. Therefore, a termination threshold is proposed in this work based on the magnitude of noise present in the data. The approach makes use of the increasing accuracy with which the actual noise is predicted for an increasing number of DOE points. To explain this in more detail, consider Fig. 9 in which the local LSS model

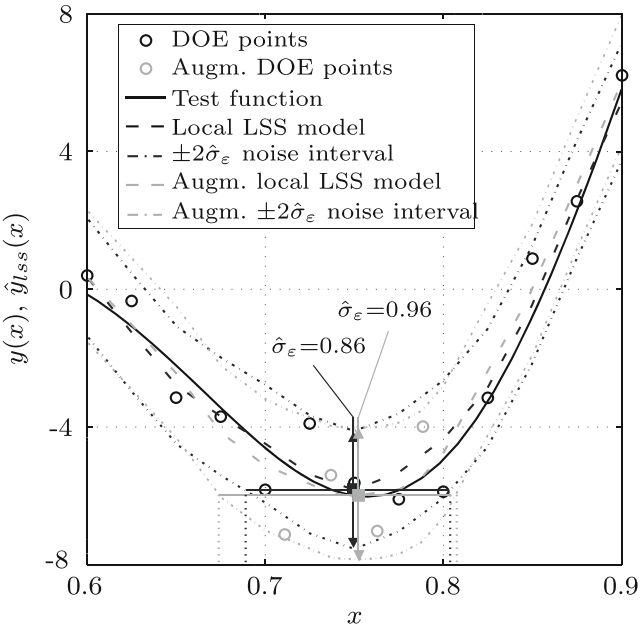

Fig. 9 Local LSS model, accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval and resulting noise bandwidth of the perturbed data set and augmented data set

of the original data set is plotted including the $\pm 2\hat{\sigma}_\varepsilon$ noise interval. In addition, the local LSS model and accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval is plotted based on an augmented data set with 4 sequentially added DOE points near the global optimum. Evaluating the magnitude of numerical noise through Eq. 8 for the initial and augmented local LSS model results in $\hat{\sigma}_\varepsilon = 0.86$ and $\hat{\sigma}_\varepsilon = 0.96$ respectively. For a sparse data set, the magnitude of noise is underestimated. Adding DOE points results in a more accurate approximation of the magnitude of noise, converging to $\sigma_\varepsilon = 1$ for the test function, see Eq. 2.

The $\pm 2\hat{\sigma}_\varepsilon$ noise interval for both local LSS models can be translated to noise bandwidths for x. The bandwidth of the original and augmented local LSS model around the global optima (visualized by the solid square markers) are $0.690 \leq x \leq 0.805$ and $0.675 \leq x \leq 0.810$ respectively and are depicted in Fig. 9. Note how the bandwidth growths for an increasing number of DOE points, converging to a maximum size for $\sigma_\varepsilon = 1$. Now any sequentially added data point will only contribute to an increased metamodel prediction accuracy if it falls outside these noise bandwidths. In other words, the update algorithm is to be terminated if sequentially added points fall within the noise bandwidths. The update procedure will automatically terminate based on the noise present in the response data due to the property of increasing bandwidth for an increasing number of DOE points, and thus increasing accuracy with which σ_ε is calculated. Depending on the calculation time of the FE simulation, the user can choose to terminate the algorithm if multiple subsequent points (e.g. 3 points) fall within the noise bandwidth, also increasing the stability of the update algorithm.

Points of discussion

As a point of discussion, note that the proposed approach provides a measure for the magnitude of numerical noise present in the data which is subsequently used for determining an efficient termination threshold for the update algorithm. In case no numerical noise is present, the local LSS model will predict a noise bandwidth of zero, preventing the termination algorithm to become active. In this case, a threshold for the EI has to be defined to terminate the update algorithm.

In case numerical noise is present in the response data, the proposed approach becomes active. One could argue to set up more detailed FE model (e.g. finer mesh) to reduce the magnitude of numerical noise. However, these models will be far more computationally expensive and unsuitable for use in an optimization procedure. Moreover, it is noted here that it is of less interest to accurately determine the exact measure of noise or to provide an accurate estimate of the confidence interval on the objective function value. Again, this would require more detailed FE models or many computationally expensive simulations. If the user is interested in the exact average performance in the optimum, it would be more beneficial to set up a more detailed FE model after termination of the update algorithm.

The proposed approach will be demonstrated next by the sequential optimization of 2 metal forming processes.

Application 1: V-bending process

The proposed approximate optimization strategy, including the regularization and local approximation techniques, will now be applied to optimize a V-bending process. The industrial application is performed in cooperation with Philips Consumer Lifestyle. An impression of the production process and resulting product is shown in Fig. 10. A piece of sheet metal is placed in between a punch (upper tool) and a die (lower tool), after which the punch is lowered by a prescribed displacement. During the bending process, the material experiences local elastic and plastic deformation. After withdrawal of the punch, the product shows elastic springback.

Figure 10a shows the FE model of the part and tools. A 2D model is created assuming a plane strain condition. The implicit code MSC. Marc has been used as FE code. Due to symmetry of the product, only one half of the geometry has been modeled. Both the die and punch are modeled to be non-rigid and discretized using quadrilateral elements.

The dimensions of the punch (L) is chosen as a design variable x with values ranging between 4.1 mm and 5 mm, see Fig. 10b. The flange angle θ, spanned up by the marked line segments in Fig. 10c, is essential to ensure an optimal performance of the final assembled product. The challenge of this study is therefore to determine the optimal setting for L for which θ equals the target angle of $\theta_t = 91.75°$. Note that this application is considered in Wiebenga et al. [24] as a 9D robust optimization problem. The focus in this work is on numerical noise, therefore a single design variable and response function is considered

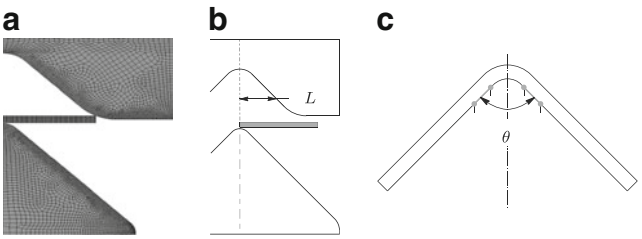

Fig. 10 (**a**) 2D FE model of the V-bending process, (**b**) definition of the design variable L and (**c**) angle θ defining the flange shape of the final product

for simplicity. The quantified optimization formulation is given by:

$$\begin{aligned} &\text{find} && x \\ &\min && |\theta - \theta_t| \\ &\text{s.t.} && 4.1 \leq L \leq 5 \end{aligned} \tag{10}$$

For the purpose of this research, a reference data set is generated by performing 200 FE simulations. The response values for θ and reference model are presented in Fig. 11. The reference model is obtained by LSS approximation of the response set using $n_l = n_r = 10$. The bending angle shows uncorrelated fluctuations about a smooth trend. The smooth response trend line is expected when modeling a physical phenomenon. The fluctuations are particularly caused by changes in the computational mesh and contact conditions. Clearly, numerical noise is present. Evaluating the magnitude of numerical noise through Eq. 8 for this data set results in $\hat{\sigma}_\varepsilon = 0.01$. Moreover, the target objective function value of $\hat{f}_{\text{opt}} = 91.75°$ is found for the optimal setting $L_{\text{opt}} = 4.45$ mm.

Data sampling and initial approximate optimization

Following the approximate optimization approach as presented in Section "Approximate optimization strategy", an initial DOE of 10 points is created. The family of regularized metamodels is subsequently fitted and validated. Based on the validation results as presented in Table 4, a regularized Kriging model is identified as the most accurate fit. Figure 12a presents the initial 10 DOE points and the regularized Kriging approximation. In addition, its basic counterpart is plotted showing a rather smooth behavior. It shows that in case perturbed data is very sparse, problems are not likely to occur since perturbations in the data can still be accommodated with a smooth approximate function. This is also reflected in the coefficient of determination of the basic Kriging model, i.e. $R^2 = 0.95$. Based on the regularized Kriging model, the optimal objective function value is determined at $L_{\text{opt}} = 4.49$ mm, see the diamond marker in Fig. 12a.

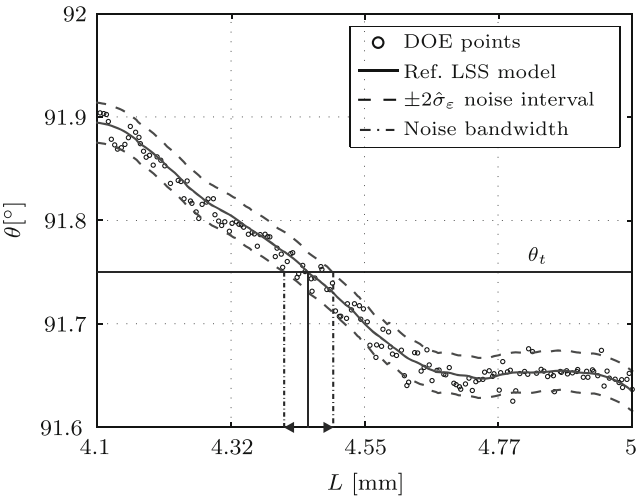

Fig. 11 V-bending simulation response and resulting reference LSS model

Table 4 Coefficient of determination (R^2) of the family of metamodels based on the V-bending data set

Number of data points	10 (initial)	20 (final)
Reg. Polynomial - Order 2	0.93	0.92
Reg. Kriging	0.98	0.98
Reg. Radial Basis Function	0.94	0.95
Reg. Neural Network	0.97	0.99

Sequential optimization

The deviation between the optimal design variable setting prediction based on the metamodel and the reference LSS model can be decreased by the sequential optimization procedure. The EI, given by Eq. 9 and calculated based on the Kriging model, is given in Fig. 12a. Maximizing the EI results in the design variable coordinate $L' = 4.49$ mm of the first infill point. After determining the response at this location, the family of metamodels is updated. As a result, also the prediction error and the EI are revised.

The intermediate metamodel results and accompanying local LSS models after adding 3, 6 and 9 DOE points are presented in Fig. 13a, b and c respectively. In addition, the $\pm 2\hat{\sigma}_\varepsilon$ noise intervals are plotted where the magnitude of noise is estimated at $\hat{\sigma}_\varepsilon = 0.0069$, $\hat{\sigma}_\varepsilon = 0.0071$ and $\hat{\sigma}_\varepsilon = 0.0078$ respectively. Compared to the reference data set, the magnitude of noise is initially underestimated but converges to the reference solution for an increasing number of locally added DOE points. As a consequence, also the width of the resulting noise bands increase, terminating the update algorithm after adding 10 DOE points. The last 3 subsequent DOE points fall within the noise band.

Figure 12 shows the final result of applying the sequential optimization procedure. The validation results of fitting the family of regularized metamodels to the final data set is presented in the last column of Table 4. Although very close to the regularized Kriging model, the regularized NN model is identified as the most accurate final fit and plotted in Fig. 12 with its basic counterpart. Clearly, the latter model shows over fitting behavior which is also reflected by its low coefficient of determination of $R^2 = 0.80$. Also note that this metamodel approximation results in multiple

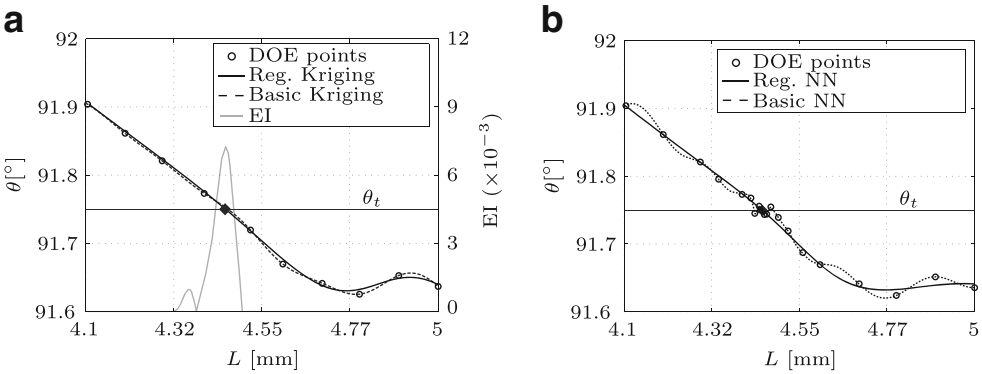

Fig. 12 Metamodel approximation of the (**a**) initial and (**b**) final updated V-bending data set

optima that meet the target angle $\theta_t = 91.75°$. All these solutions fall within the bandwidth of noise. The regularized NN model accurately predicts a unique optimal setting of $L^{'} = 4.45$ mm which meets the reference solution.

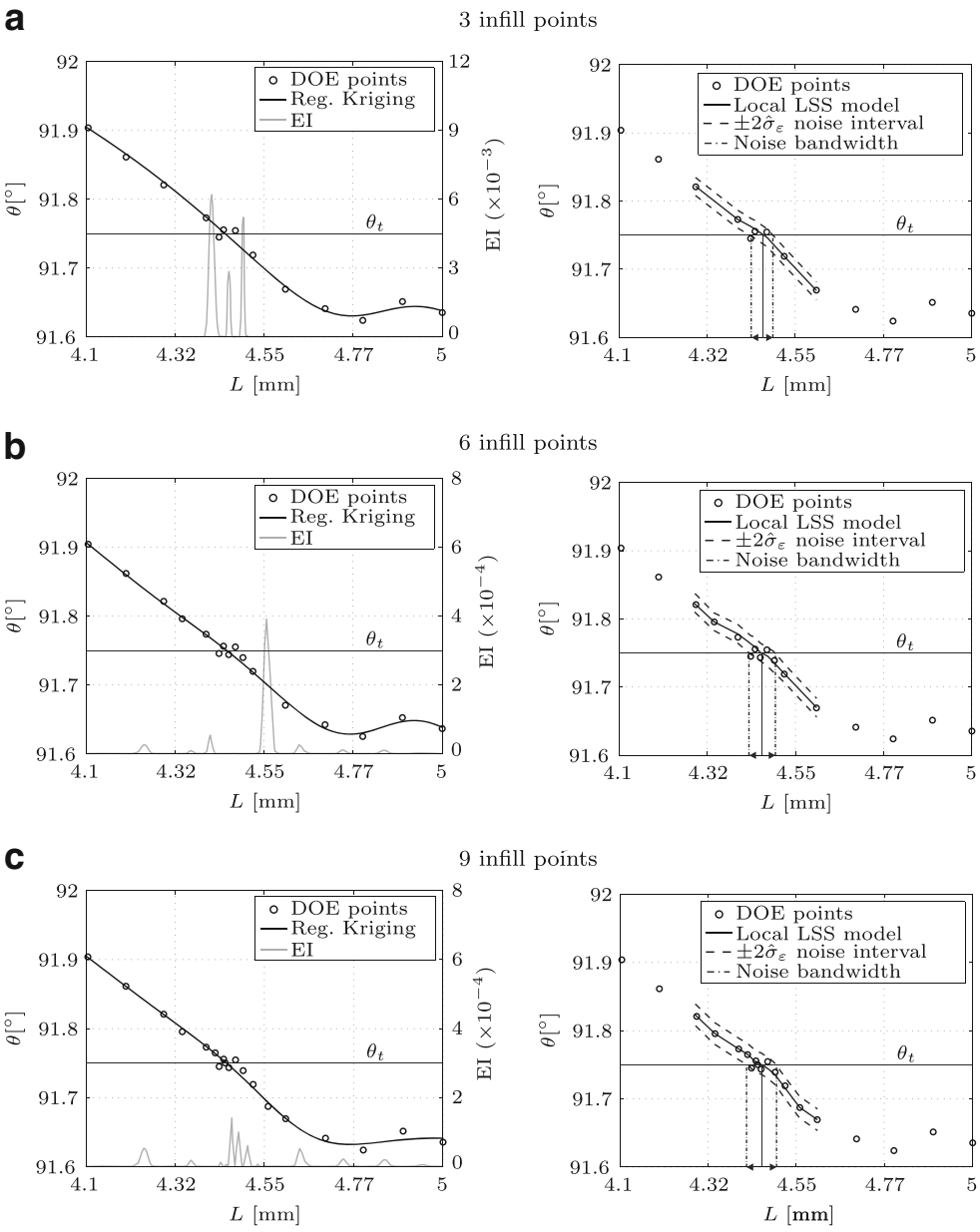

Fig. 13 Best metamodel approximation (left) and local LSS approximation (right) of the V-bending data set after (**a**) 3, (**b**) 6 and (**c**) 9 updates

This industrial application shows that the presence of numerical noise cannot be neglected in practice. Accounting for it in approximate optimization is highly important to increase the overall accuracy of the optimization results, especially in combination with sequential optimization procedures or dense sampling strategies.

Application 2: Cup-stretching process

The second industrial application considered in this work is the stretching process of a hemispherical cup, performed in cooperation with Tata Steel. See Fig. 14 for an impression of the 3D model and resulting product. The cup is produced out of a deep drawing steel for which the full anisotropic material behavior is obtained by mechanical testing. The in–house implicit FE code DiekA [31] is used to set up a FE model of the forming process, see Fig. 14a. The quarter circle blank is discretized using triangular Kirchhoff shell elements with 5 integration points through the thickness, see Fig. 14b. The blank is subsequently stretched by a prescribed punch displacement of 42 mm with the friction coefficient set to 0.14.

During the forming process, the material in the cup wall will be stretched significantly. After 42 mm punch displacement, the material mainly experiences plane strain deformation in a band halfway the cup wall, see Fig. 14b. Fracture will occur at this location once the peak strain exceeds the forming limit. To determine the moment of fracture, an accurate prediction of the strain in the cup wall is required.

For the purpose of this research, the maximum major strain (ε) is evaluated in the Rolling Direction (RD) of the blank, see Fig. 14b. The blank radius (R) is chosen as a design variable ranging between 80 mm $\leq R \leq$ 100 mm. Since the element size of the quarter circle blank is kept constant at 1.4 mm, the number of elements per blank size configuration increases from approximately 6022 to 9424 elements for the smallest and largest blank size respectively. As with the process in practice, the material thickness (t) shows variation around a mean value. The scatter in material thickness is obtained by measuring 48 blanks and can be described by a normal distribution $t \sim \mathcal{N}(\mu_t, \sigma_t^2)$ with $\mu_t = 0.8$ mm and $\sigma_t = 0.02$ mm. As a result, the objective becomes a function of the weighted sum of both the mean μ_f and standard deviation σ_f of the strain response. Note that σ_f is the standard deviation as a result of the noise variable t which differs from σ_ε which is the standard deviation of the error caused by numerical noise. The challenge of this robust optimization study is to find the optimal setting R for which the strain in RD, including its scatter caused by the noise variable t, does not exceed the forming limit strain of $\varepsilon_l = 0.4$. The quantified optimization formulation is given by:

$$\begin{aligned} &\text{find } x \\ &\max\ \mu_f \\ &\text{s.t. } \mu_f + 3\sigma_f \leq 0.4 \\ &\quad 80 \leq R \leq 100 \\ &\quad t \sim \mathcal{N}(0.8, 0.02^2) \end{aligned} \tag{11}$$

A reference data set is obtained beforehand by performing 1071 FE simulations at an equidistant grid of 51 $\times$ 21 design and noise variable settings respectively for the purpose of this investigation. The upper and lower bound of the noise variables are set at $\mu_t + 3\sigma_t$ and $\mu_t - 3\sigma_t$, respectively. A graphical representation of the maximum major strain in the RD is given in Fig. 15. The response shows sudden discontinuities in the direction of the blank radius caused by changes in the computational mesh. A more smooth response can be observed in the direction of the blank thickness. Although the characteristics of noise are different compared to the previous application, the sudden discontinuities in the response can be seen as numerical noise since they do not represent the smooth physical performance and thus a similar problem is faced in approximate optimization.

A smooth reference solution is obtained by LSS approximation of the response set using 10 nearest neighbors determined by an elliptical search region. Evaluating the magnitude of numerical noise in the major strain response through Eq. 8 results in $\hat{\sigma}_\varepsilon = 0.003$. The LSS model of the reference data set and accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval is depicted in Fig. 16a. Note that for this multi-dimensional optimization problem, the greatest magnitude of numerical noise in the blank radius direction is dominant in determining $\hat{\sigma}_\varepsilon$ for the whole design space. The objective function can now be determined based on the reference data set as well as the LSS model by Monte Carlo sampling. Both results are plotted in Fig. 16b. Note the significant effect

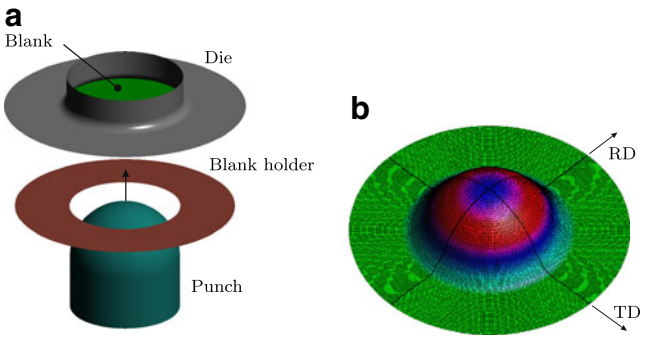

Fig. 14 (**a**) 3D model of the stretching tools and (**b**) final hemispherical cup

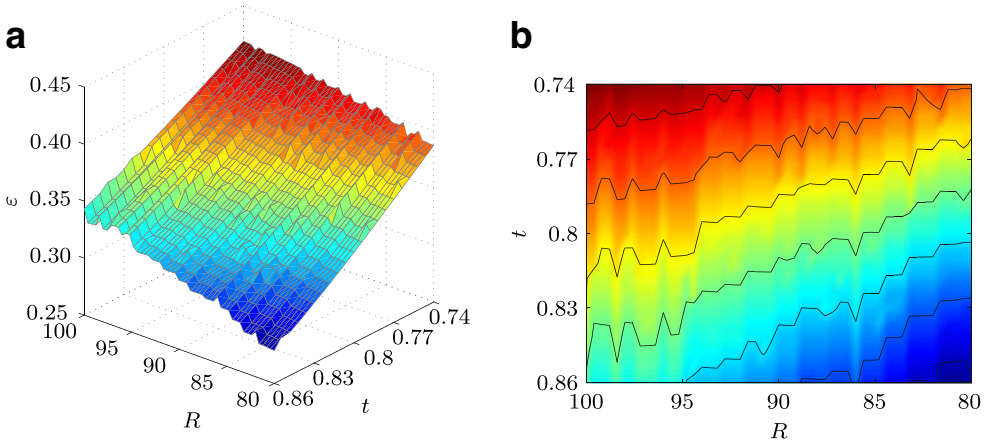

Fig. 15 Reference data set of the cup-stretching application represented by a (**a**) surface plot and (**b**) contour plot

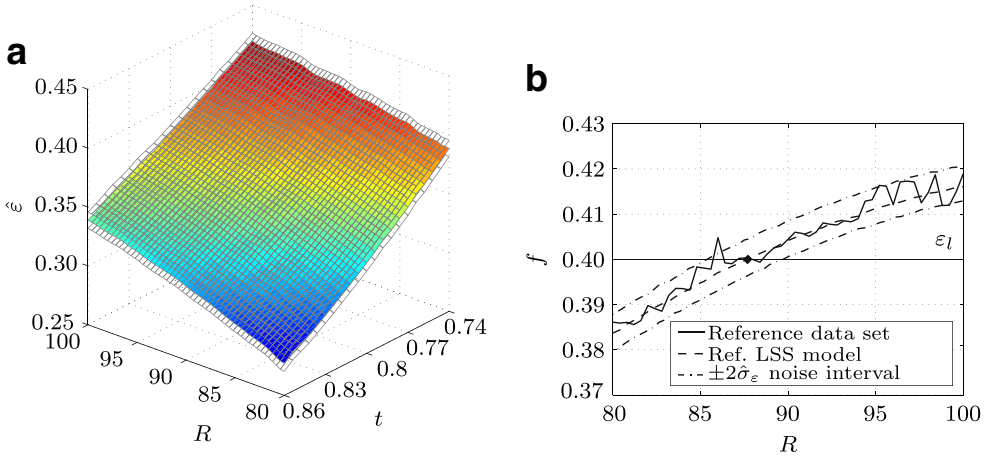

Fig. 16 (**a**) Reference LSS model and accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval, and (**b**) objective function prediction based on the reference data set and reference LSS model

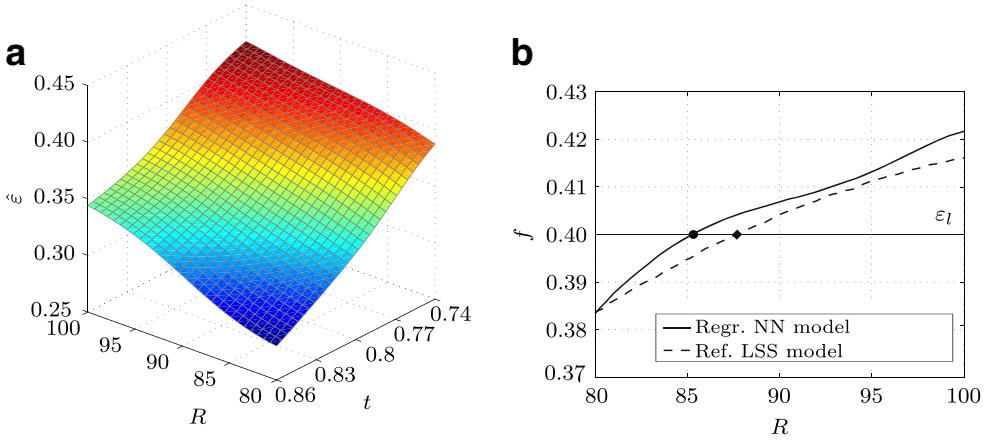

Fig. 17 (**a**) Initial regression NN model and (**b**) resulting objective function prediction

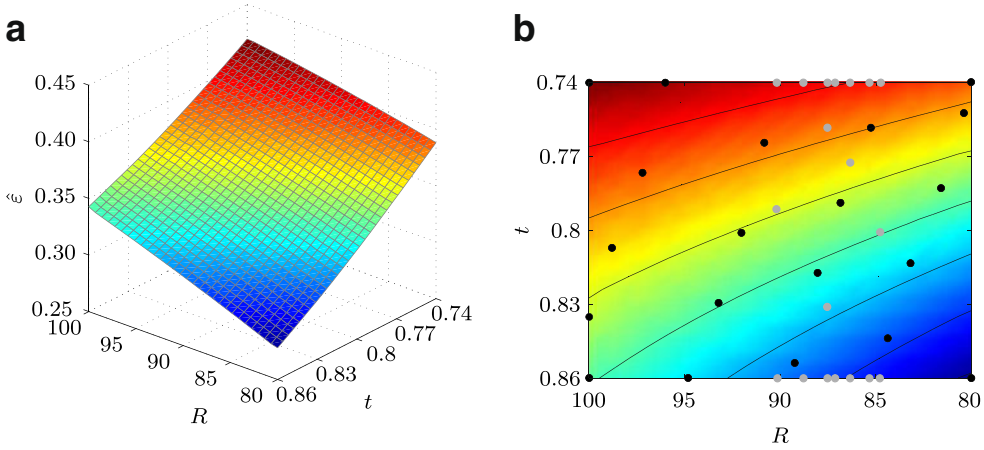

Fig. 18 (**a**) Final regression Kriging model and (**b**) contour plot showing the initial (*black*) and added (*gray*) DOE points

of numerical noise on the robust objective function resulting from the reference data set. In addition, the $\pm 2\hat{\sigma}_\varepsilon$ noise interval of the objective function is plotted obtained by sampling onto the LSS model. The magnitude of numerical noise for the objective function is determined at $\hat{\sigma}_\varepsilon$ = 0.002. Using the reference LSS model of the objective function, the target value of $\hat{f}_{\text{opt}}$ = 0.4 is found for the optimal setting $R^{'}$ = 87.5 mm, see the diamond marker in Fig. 16b.

Robust optimization

An initial DOE of 20 points is created in the 2D combined design–noise variable space. After running the FE simulations corresponding to the settings specified by the DOE, the family of regularized metamodels is fitted in the combined space. The regularized NN model is identified as the most accurate fit, see Fig. 17a. The resulting objective function prediction is plotted in Fig. 17b. Solving Eq. 11 results in the optimal value $R^{'}$ = 85.1 mm, see the circle marker in Fig. 17b. Clearly, sequential optimization is required to improve the overall prediction accuracy of the metamodel and decrease the discrepancy with the optimal value prediction of the reference solution.

Sequential robust optimization

The initial DOE is augmented with additional DOE points by the sequential optimization procedure. The extension of the EI algorithm to account for the influence of noise variables is presented in Wiebenga et al. [24]. The termination criterion as presented in Section "Optimization and update algorithm" is reached after adding 19 DOE points. The final regularized Kriging metamodel solution is depicted in Fig. 18a. The initial and sequentially added DOE points are plotted in Fig. 18b. Note how the design space has been sampled extensively near the global optimum in the control variable space.

A plot of the final objective function prediction and reference objective function is provided in Fig. 19a. The final optimum is found close to the reference optimum at $R^{'}$ = 87.3 mm, see the circle and diamond marker in Fig. 19a. The local LSS model and accompanying $\pm 2\hat{\sigma}_\varepsilon$ noise interval is provided in Fig. 19b where the magnitude of noise is estimated at $\hat{\sigma}_\varepsilon$ = 0.0016. Although the magnitude of noise is still underestimated compared to the reference data set, the update algorithm is efficiently terminated after adding 3 subsequent DOE points within the noise band. Further improvement of the robust objective function prediction is

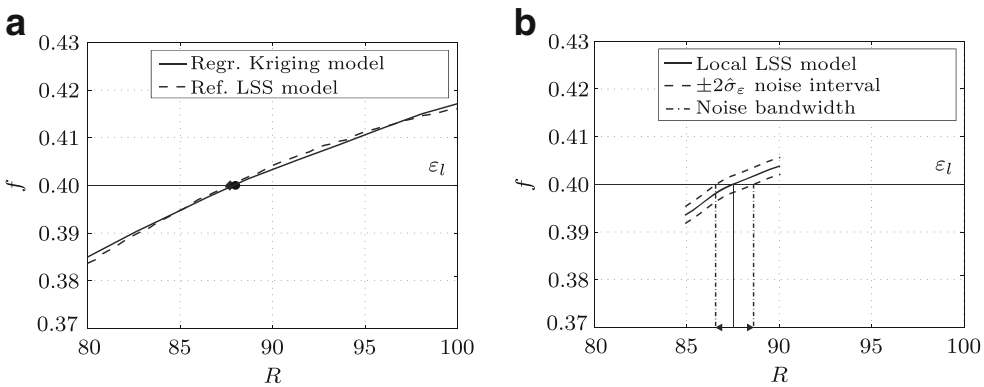

Fig. 19 (**a**) Final objective function prediction and (**b**) local LSS model

not expected based on the presence of numerical noise in the FE simulation response.

Conclusions

This work demonstrates the presence and the severe deteriorating effect of numerical noise on the approximation quality of metamodels, limiting the accuracy and general usability of approximate optimization techniques. It is shown that the use of regularization in the fitting procedure of metamodels can alleviate the problem when noise is present in the FE simulation response. Quantification of noise is achieved by local smoothing of response data. The proposed general approximate optimization approach includes these techniques and makes a coupling with a sequential optimization algorithm based on expected improvement. Application of the approach to two industrial optimization cases show the efficient termination of the update algorithm based on the magnitude of numerical noise present in the response data. Moreover, despite the presence of numerical noise the approach shows a fast and accurate convergence towards the global optimum while maintaining the benefits of approximate optimization.

Acknowledgments This research was carried out under the project number M22.1.08303 in the framework of the Research Program of the Materials innovation institute (www.m2i.nl). The industrial partners co-operating in this research are gratefully acknowledged for their useful contributions to this research.

Appendix: Metamodel types

The aim of a metamodel, denoted by $\hat{y}(\mathbf{x})$, is to accurately predict the trend of the FE simulation response or true model $y(\mathbf{x})$. Consider a nonlinear regression model, including a random error term ε, defined by:

$$y(\mathbf{x}) = \hat{y}(\mathbf{x}) + \varepsilon \tag{12}$$

What follows is a description of different types of metamodeling techniques used in this work to construct $\hat{y}(\mathbf{x})$.

Response surface methodology

The *Response Surface Methodology* (RSM) is a well known method for creating an approximate model of a response [32]. Although this method is generally used for constructing a response surface from physical experiments, many authors have applied it to numerical experiments as well. One of the reasons is it's ability to filter out numerical noise [14–16].

Using RSM, a polynomial model is fitted through the n response measurements or observations $\mathbf{y}$ allowing for a random error term ε. Equation 12 can now be written in matrix form as:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{13}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \text{ and } \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Now, $\mathbf{X}$ is an $n \times p$ matrix of the levels of independent variables with $p = m + 1$, β is a $p \times 1$ vector of regression coefficients, and ε is an $n \times 1$ vector or random error terms. Note that the design matrix $\mathbf{X}$ can incorporate non-linear terms with respect to the m design variables. The order of these terms are referred to as the order of the polynomial model. The metamodel is given by $\hat{y} = \mathbf{X}\beta$. The unknown regression coefficients β are determined by minimizing the error sum of squares at the training points, also referred to as quadratic loss function or L_2-norm:

$$\varepsilon^{\mathrm{T}}\varepsilon = (\mathbf{y} - \mathbf{X}\beta)^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\beta) \tag{14}$$

Differentiating Eq. 14 with respect to β and setting the results to zero yields the best estimation of β:

$$\hat{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{15}$$

where $\hat{\beta}$ denotes the estimator of β. The response prediction $\hat{y}_0$ at an unknown design variable setting $\mathbf{x}_0$ is now given by the explicit function:

$$\hat{y}_0 = \mathbf{x}_0^{\mathrm{T}}\hat{\beta} \tag{16}$$

The variance at this location is given by:

$$\mathrm{var}[\hat{y}_0] = \sigma^2\mathbf{x}_0^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{x}_0 \tag{17}$$

The unbiased estimate of the error variance σ^2 is given by:

$$\begin{aligned} \hat{\sigma}^2 &= \frac{\varepsilon^{\mathrm{T}}\varepsilon}{n-p} \\ &= \frac{\sum_{i=1}^{n} \varepsilon_i^2}{n-p} \\ &= \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n-p} \end{aligned} \tag{18}$$

The prediction uncertainty of the metamodel is given by the square root of the variance as calculated in Eq. 17.

Regression in response surface methodology

Instead of estimating the unknown regression coefficients based on the error sum of squares through Eq. 14, in case of ridge regression these are obtained by minimizing the regularized loss function:

$$\varepsilon^{\mathrm{T}}\varepsilon + \lambda\beta^{T}\beta \tag{19}$$

where the regularization parameter λ governs the relative importance of the regularization term, penalizing large weights, compared with the error sum of squares term. The ridge regression formulation results in the solution:

$$\hat{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} - \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{20}$$

where the optimal λ can be identified by generalized cross-validation. A modification of Eq. 18 in case of ridge regression is provided in MacKay [33].

Kriging

Computer simulations are deterministic in nature meaning that repeated runs for the same input parameters will yield exactly the same result. Therefore, the remaining error, denoted by ε in Eq. 12, should formally be zero [19]. In other words, the metamodel should interpolate through the response values at the training points.

The approach proposed in Sacks et al. [17] and Jones et al. [18] is referred to as *Design and Analysis of Computer Experiments* (DACE) where generally Kriging is used as interpolation technique. Kriging involves a defined *base function* or regression part, similar to fitting a RSM metamodel. The random error term ε in Eq. 12 is replaced by *basis functions* or a stochastic part $Z(\mathbf{x})$ to compute the exact predictions at the available training points:

$$\mathbf{y} = \mathbf{X}\beta + Z(\mathbf{x}) \tag{21}$$

where $Z(\mathbf{x})$ is assumed to be a Gaussian stochastic process with mean zero, process variance σ_z^2, and spatial covariance function given by:

$$\mathrm{cov}(Z(x_i), Z(x_j)) = \sigma_z^2 R(x_i, x_j) \tag{22}$$

where $R(x_i, x_j)$ describes the correlation between the known measurement points x_i and x_j. The correlation function R determines the shape of the metamodel between measurement points and is, in case of a Gaussian exponential correlation function, given by:

$$R(\theta, x_i, x_j) = \exp^{-\theta(x_i - x_j)^2} \tag{23}$$

Now, in case m design variables are present, the correlation function depends on the m one-dimensional correlation functions as follows:

$$R(\theta, \mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^{m} \exp^{-\theta_l (x_{il} - x_{jl})^2} \tag{24}$$

The entries of the vectors $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}^{\mathrm{T}}$ and the distance between the known measurement points $\mathbf{x}_i$ and $\mathbf{x}_j$ determine the structure of $R(\theta, \mathbf{x}_i, \mathbf{x}_j)$. Analogous to RSM, a Kriging metamodel is fitted in order to minimize the mean squared error between the Kriging metamodel $\hat{y}(\mathbf{x})$ and the true but unknown response function $y(\mathbf{x})$ [19, 34]:

$$\begin{aligned} &\min\ E(\hat{y}(\mathbf{x}) - y(\mathbf{x}))^2 \\ &\text{s.t. } E(\hat{y}(\mathbf{x}) - y(\mathbf{x})) = 0 \end{aligned} \tag{25}$$

In other words, the mean squared error is minimized subject to the unbiasedness constraint that ensures there is no systematic error between the metamodel and the true function. The Best Linear Unbiased Predictor (BLUP) $\hat{y}_0$ at an untried design variable setting x_0 is now given by:

$$\hat{y}_0 = \mathbf{x}_0^{\mathrm{T}}\beta + \mathbf{r}_0^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{X}\beta) \tag{26}$$

where $\mathbf{x}_0$ is the design matrix containing the settings of the untried point x_0 and $\mathbf{X}$ the design matrix containing the training points. The vector $\mathbf{r}_0$ contains the correlation between the point (x_0, y_0) and the known measurement (x_i, y_i). $\mathbf{R}$ is a matrix containing the correlation between the training points given by Eq. 23.

The Mean Squared Error (MSE) can be calculated at location x_0 by:

$$\mathrm{MSE}(y_0) = \sigma_z^2 \left(1 - \left[\mathbf{x}_0^{\mathrm{T}}\mathbf{r}_0^{\mathrm{T}}\right] \begin{bmatrix} \mathbf{0} & \mathbf{X}^{T} \\ \mathbf{X} & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{r}_0 \end{bmatrix}\right) \tag{27}$$

The unknown Kriging parameters β, σ_z^2, and θ can be estimated by Maximum Likelihood Estimation (MLE) [17]. Note that maximization of the likelihood function is equivalent to a minimization of the error sum of squares when the error can be assumed to be a Gaussian noise. This optimization procedure is solved using the DACE toolbox provided by Lophaven et al. [34].

Regression in Kriging

In case data is contaminated with noise, it makes more sense to approximate the given data instead of interpolating the data. The generalization capability of Kriging models can be improved by adding a regularization constant λ to the leading diagonal of the correlation matrix $\mathbf{R}$ as $\mathbf{R} + \lambda\mathbf{I}$ [35]. This enables a Kriging model to regress the data and approximate noisy functions. Without the regression constant, each point is given an exact correlation with itself, forcing the metamodel to pass through the training points. The regression constant enables control on the interpolation feature of the Kriging model. The regression constant λ is now optimized along with the other unknown parameters in the MLE providing the regression Kriging predictor:

$$\hat{y}_0 = \mathbf{x}_0^{\mathrm{T}}\beta + \mathbf{r}_0^{\mathrm{T}}(\mathbf{R} + \lambda\mathbf{I})^{-1}(\mathbf{y} - \mathbf{X}\beta) \tag{28}$$

A modification of Eq. 27 in case of regression Kriging is provided in Forrester et al. [36].

Radial basis functions

A function approximation constructed by the linear combination of basis functions $h_i(\mathbf{x})$ takes the form:

$$y(\mathbf{x}) = \sum_{i=1}^{n} w_i h_i(\mathbf{x}) \tag{29}$$

where each basis function is weighted by an appropriate coefficient w_i. The idea behind Radial Basis Functions (RBF) is that every known DOE point i 'influences' its surroundings the same way in all directions according to a basis function, so that $h_i(\mathbf{x}) = \phi(r)$ where r is the radial distance $r = \|\mathbf{x} - \mathbf{x}_i\|_2$. Now the RBF approximation is a linear combination of the basis functions centered at all n DOE points:

$$y(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) \tag{30}$$

A commonly used radial basis function is the Gaussian exponential function. Referring to Eq. 23 and compose the Gaussian with the radial distance r, the radial basis function is given by:

$$\phi(r) = \exp^{-(\theta r)^2} \tag{31}$$

The weights w_i can be found by minimizing the error sum of squares at the training points. Evaluating Eq. 29 results in solving a linear system of equations of the form $\mathbf{Hw} = \mathbf{y}$. The estimated mean response $\hat{y}_0$ at $\mathbf{x}_0$ is provided by:

$$\hat{y}_0 = \mathbf{h}_0^{\mathrm{T}} \hat{\mathbf{w}} \tag{32}$$

The variance at this location is given by:

$$\mathrm{var}[\hat{y}_0] = \sigma^2 \mathbf{h}_0^{\mathrm{T}} (\mathbf{H}^{\mathrm{T}} \mathbf{H})^{-1} \mathbf{h}_0 \tag{33}$$

Similar to RSM, the unbiased estimate of the error variance σ^2 is given by Eq. 18.

Regression in radial basis function approximation

The regularized loss function is formulated as:

$$\varepsilon^T \varepsilon + \lambda \mathbf{w}^T \mathbf{w} \tag{34}$$

Minimization of the loss function results in the best estimation of the regularized weight coefficients:

$$\hat{\mathbf{w}} = (\mathbf{H}^T \mathbf{H} - \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{y} \tag{35}$$

Also note the resemblance with Eq. 20. A modification of the error variance in case of ridge regression is provided in MacKay [33] and Orr [37].

Artificial neural networks

Neural Networks (NN) follow the same form as Eq. 29 where the choice of the letter h for the basis functions reflects the interest in NN which have hidden units. In addition to the basis functions, the building blocks of NN are neurons and connections. Differences in the learning rules and the network topology result in different NN architectures or NN concepts. In this work, two layer *feedforward backpropagation* NN are utilized.

A two layer NN architecture is presented in Fig. 20. This architecture is referred to as *feed forward* since information only proceeds forward through the network and there are no feedback loops in between the layers. Starting with the first layer of S neurons, the output $\mathbf{a}$ of the so-called *Hidden layer* (HL) is given by:

$$\mathbf{a} = \mathbf{G}^{(1)}(\mathbf{d}_{(\mathrm{HL})}), \quad \mathbf{d}_{(\mathrm{HL})} = \mathbf{W}_{(\mathrm{HL})}\mathbf{x} + \mathbf{b}_{(\mathrm{HL})} \tag{36}$$

The layer includes a weight matrix $\mathbf{W}_{(\mathrm{HL})} \in \mathbb{R}^{S \times m}$, an input vector $\mathbf{x}$, a bias vector $\mathbf{b}_{(\mathrm{HL})} = \{b_{(\mathrm{HL})1}, b_{(\mathrm{HL})2}, \ldots, b_{(\mathrm{HL})S}\}^{\mathrm{T}}$, basis functions or activation functions $\mathbf{G}$ and an output vector $\mathbf{a} = \{a_1, a_2, \ldots, a_S\}^{\mathrm{T}}$. The basis functions used in this work are the *tangent sigmoid* and the *linear* basis functions. The tangent sigmoid basis function $G^{(1)}(d)$ can take any arbitrary input value $d \in \mathbb{R}$ and suppress the output into the range $(-1, 1)$ by:

$$G^{(1)}(d) = \frac{2}{1 + \exp(-2d)} - 1 \tag{37}$$

The output of the linear basis function $G^{(2)}(d)$ equals its input:

$$G^{(2)}(d) = d \tag{38}$$

The output of the hidden layer $\mathbf{a}$ is the input for the next layer. This layer is referred to as the *Ouput Layer* (OL) since its output is also the output of the network. The basis function used in the hidden layer is the tangent simoid function whereas the linear function is used in output layer. These functions are preferred because of their differentiability which enables determining partial derivatives used in parameter estimation.

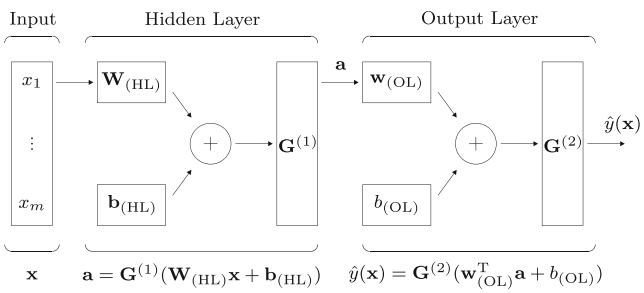

Fig. 20 Two layer NN architecture

The predictor of a two layer architecture with a single network output is now given by:

$$\hat{y}(\mathbf{x}) = \mathbf{G}^{(2)}(\mathbf{d}_{(\mathrm{OL})}) = \mathbf{d}_{(\mathrm{OL})}, \quad \mathbf{d}_{(\mathrm{OL})} = \mathbf{w}_{(\mathrm{OL})}^{\mathrm{T}}\mathbf{a} + b_{(\mathrm{OL})} \tag{39}$$

In essence, Eq. 39 is the linear combination of the weighted tangent sigmoid basis functions. The unknown parameters in Eq. 39 are the bias term of the output layer $b_{(\mathrm{OL})}$, the vector with output layer weights $\mathbf{w}_{(\mathrm{OL})} = \{w_{(\mathrm{OL})1}, w_{(\mathrm{OL})2}, \ldots, w_{(\mathrm{OL})S}\}^{\mathrm{T}}$ and the hidden layer bias vector $\mathbf{b}_{(\mathrm{HL})}$ and weight matrix $\mathbf{W}_{(\mathrm{HL})}$. The unknown weight and bias parameters can be estimated by minimizing the error sum of squares at the training points. This unconstrained nonlinear optimization problem is solved using a *Levenberg-Marguardt* optimization algorithm. The procedure is also referred to as Bayesian regulation backpropagation [33].

The variance estimation theory for nonlinear regression as in Eq. 17 and 33 also applies to NN [38]:

$$\mathrm{var}[\hat{y}_0] = \sigma^2 \mathbf{g}_0^{\mathrm{T}}(\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{g}_0 \tag{40}$$

where $\mathbf{J}$ is a matrix whose ijth entry is given by $\partial\hat{y}(\mathbf{x}_i)/\partial z_j$ and $\mathbf{g}_0$ is a vector whose ith entry is $\partial\hat{y}(\mathbf{x}_0)/\partial z_j$, evaluated at the optimal parameter vector $\hat{\mathbf{z}}$ where $\mathbf{z}$ represents the collection of all unknown parameters. Note that for estimating the weights in NN, $\mathbf{J}$ is already calculated as part of the optimization procedure. The unbiased estimate of the error variance σ^2 is given by Eq. 18. The procedure as described in this section is solved using the NN Matlab toolbox [39].

Regression in artificial neural networks

With many weight and bias parameters involved in NN, there is a considerable danger of overfitting. The generalization capability can be improved by minimizing the regularized loss function as in Eq. 34. Note that regularization both assists in avoiding over fitting due to a high number of hidden units S (and thus many weights and biases to be determined) and the presence of numerical noise in the response data. The loss function is minimized using the *Levenberg-Marguardt backpropagation* algorithm as implemented in the NN Matlab toolbox [39]. A modification of the error variance in case of ridge regression is provided in [38].

References

1. Barthelemy JFM, Haftka RT (1993) Approximation concepts for optimum structural design - a review. Struct Multidiscip Optim 5:129–144
2. Simpson TW, Toropov V, Balabanov V, Viana FAC (2008) Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come - or not. In 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, MAO, art. no. 2008–5802
3. Wang H, Li G (2010) Sheet forming optimization based on least square support vector regression and intelligent sampling approach. Int J Mater Form 3:9–12
4. Jansson T, Andersson A, Nilsson L (2005) Optimization of draw-in for an automotive sheet metal part: An evaluation using surrogate models and response surfaces. J Mater Process Technol 159:426–434
5. Ejday M, Fourment L (2010) Metamodel assisted evolutionary algorithm for multi-objective optimization of non-steady metal forming problems. Int J Mater Form 3:5–8
6. Chenot J-L, Bouchard P-O, Fourment L, Lasne P, Roux E (2011) Optimization of metal forming processes for improving final mechanical strength Computational Plasticity XI - Fundamentals and Applications, COMPLAS XI, pp 42–55
7. Clees T, Steffes-lai D, Helbig M, Sun D-Z (2010) Statistical analysis and robust optimization of forming processes and forming-to-crash process chains. Int J Mater Form 3:45–48
8. Li YQ, Cui ZS, Ruan XY, Zhang DJ (2006) Cae-based six sigma robust optimization for deep-drawing sheet metal process. Int J Adv Manuf Technol 30:631–637
9. Strano M (2008) A technique for fem optimization under reliability constraint of process variables in sheet metal forming. Int J Mater Form 1:13–20
10. Kleiber M, Knabel J, Rojek J (2004) Response surface method for probabilistic assessment of metal forming failures. Int J Numer Anal Model 60:51–67
11. Oden JT, Belytschko T, Fish J, Hughes TJR, Johnson C, Keyes D, Laub A, Petzold L, Srolovitz D, Yip S (2006) Simulation based engineering science. Technical report, National Science Foundation
12. Tekkaya AE, Martins PAF (2009) Accuracy, reliability and validity of finite element analysis in metal forming: a user's perspective. J Eng Comput 26:1026–1056
13. van Keulen F, Toropov VV (1997) New developments in structural optimization using adaptive mesh refinement and multi-point approximations. Eng Optim 29:217–234
14. Giunta AA, Dudley JM, Narducci R, Grossman B, Haftka RT, Mason WH, Watson LT (1994) Noisy aerodynamic response and smooth approximations in hsct design. AIAA J Proc 5th Symp Multidiscip Struct Optim 94–4376-CP:1117–1128
15. Papila M, Haftka RT (2000) Response surface approximations: Noise, error repair and modeling errors. AIAA J 38:2336–2343
16. Goel T, Haftka RT, Papila M, Shyy W (2006) Generalized pointwise bias error bounds for response surface approximations. Int J Numer Methods Eng 65:2035–2059
17. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4:409–423
18. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. Global Optim 13:455–492
19. Santner T, Williams B, Notz W (2003) The design and analysis of computer experiments. Springer–Verlag, New York
20. Toropov V, van Keulen F, Markine V, de Boer H (1996) Multipoint approximations for structural optimization problems with noisy response functions. AIAA J Proc Symp Multidiscip Struct Optim A96-38701:10–31
21. Siem AYD, den Hertog D (2007) Kriging models that are robust with respect to simulation errors. CentER discussion paper No. 200768, ISSN 0924-7815, Tilburg University, p 1–29
22. Bishop CM (2006) Pattern recognition and machine learning. Springer Science + Business Media, New York

23. Bonte MHA, Fourment L, Do TT, van den Boogaard AH, Huetink J (2010) Optimization of forging processes using finite element simulations : A comparison of sequential approximate optimization and other algorithms. J Struct Multidisc Optim 42(5):797–810
24. Wiebenga JH, van den Boogaard AH, Klaseboer G (2012) Sequential robust optimization of a v-bending process using numerical simulations. J Struct Multidisc Optim 46:137–153
25. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) Numerical recipes. Cambridge University Press, Cambridge
26. Tikhonov AN, Arsenin VY (1977) Solutions of ill-posed problems. Wiley, New York
27. Golub GH, Heath M, Wahba G (1979) Generalized cross-validation as a method choosing a good ridge parameter. Technometrics 21:215–223
28. Craven P, Whaba G (1979) Smoothing noisy data with spline functions. Numer Math 31:377–403
29. Savitzky A, Golay MJE (1964) Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 36:1627–1639
30. Schonlau M (1997) Computer experiments and global optimization. PhD thesis. University of Waterloo, Ontario, Canada
31. DiekA (2012) In-house finite element code for forming simulations of the University of Twente. http://www.utwente.nl/ctw/tm/research/NSM/software/dieka/. Enschede, the Netherlands
32. Myers RH, Montgomery DC (2002) Response surface methodology. Wiley, New York
33. MacKay DJC (1992) Bayesian interpolation. Neural Comput 4:415–447
34. Lophaven SN, Nielsen HB, Sondergaard J (2002) Dace: a Matlab Kriging Toolbox Version 2.0. Technical Report imm-tr-2002-12. Technical report. Technical University of Denmark, Copenhagen, Denmark
35. Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley, New York
36. Forrester A, Keane AJ, Bresslo NW (2006) Design and analysis of noisy computer experiments. AIAA J 44:2331–2339
37. Orr MJL (1996) Introduction to radial basis function networks. Technical report. University of Edinburgh, UK
38. de Veaux RDD, Schumi J, Schweinsberg J, Ungar LH (1998) Prediction intervals for neural networks via nonlinear regression. Technometrics 40:273–282
39. Matlab Version 7.13.0 (2012) The MathWorks Inc, Natick Massachusetts, USA