Engineering Computations

Data-driven model order reduction with surrogate elements for transient simulations

Markus Franke
OTH Regensburg University of Applied Sciences, Regensburg, Germany
Tom Janis Krause
OTH Regensburg University of Applied Sciences, Regensburg, Germany and BMW Group, Munchen, Germany, and
Marcus Wagner
OTH Regensburg University of Applied Sciences, Regensburg, Germany

2655

Received 12 June 2024
Revised 7 January 2025
Accepted 18 May 2025

Abstract

Purpose – The purpose of this study is to introduce surrogate elements for static and transient finite element simulations. These elements are designed to replace regions of several conventional solid elements with a single artificial element that possesses a reduced number of degrees of freedoms (dofs). A notable advantage of our surrogate elements is their seamless integration into standard finite element meshes.

Design/methodology/approach – The construction of the surrogate elements stiffness and mass matrices is achieved through an optimization process wherein displacements serve as the optimization objective. Moreover, the matrices are designed to possess properties analogous to those of standard finite elements. A particular focus is placed on ensuring that the artificial stiffness matrices are positive semi-definite. Furthermore, artificial degrees of freedom are introduced.

Findings – The efficacy of the proposed technique is demonstrated through its application to two different use cases. It is demonstrated that, despite being trained on examples comprising a single surrogate element, the surrogate elements can be employed multiple times within complex and practical models. The degree of accuracy achieved in these applications is noteworthy. Moreover, the proposed method is considerably faster than the fully discretized models.

Originality/value – The study expands the field of substructuring and model order reduction by incorporating artificial surrogate elements built by neural networks, which enables seamless integration with standard finite element analysis via positive semi-definite matrices. Furthermore, the introduction of artificial degrees of freedom, which are detached from the computational domain, is proposed. Once trained, the surrogate elements can be utilised in load and support independent scenarios.

Keywords Surrogate modelling, Multiscale modeling, Substructuring, Data-driven engineering, Informed neural network, Artificial degrees of freedom

Paper type Research paper

1. Introduction

The Finite Element Method (FEM) is a widely used numerical simulation method in almost all scientific and engineering disciplines. Despite its capacity to address the most complex problems in any physical domain, the method has the disadvantage of requiring significant computational resources, particularly for systems with a high number of degrees of freedom (dof).

This open access publication was funded by Ostbayerische Technische Hochschule Regensburg.

Data availability: The datasets generated during the current study are available from the corresponding author on request.

Engineering Computations
Vol. 43 No. 7, 2026
pp. 2655-2680
Emerald Publishing Limited
e-ISSN: 1758-7077
p-ISSN: 0264-4401
DOI 10.1108/EC-06-2024-0511

Achieving the goal of reducing computational costs involves several approaches. Recent research focuses on advancements in numerical solvers (Sato *et al*., 2023; Chen *et al*., 2021). Another promising area is the replacement of complete models with surrogates as described by Kudela and Matousek (2022) or Franke and Wagner (2024). While this significantly reduces computational cost, these surrogate models are typically limited to specific configurations. Another important area is called model order reduction (ROM). Methods such as proper orthogonal decomposition (POD), the Krylov subspace method or the modal reduction method are used (Bathe, 2007; Wriggers, 2008; Ingrid and Rottner, 1999; Schilders *et al*., 2008). All these methods have in common that the full system is approximated by a system with less dofs and after solving it is re-expanded to the full model. Furthermore, there are purely non-intrusive MOR techniques for finite element models. These include projection-based approaches (Mahdiabadi *et al*., 2021; Le Guennec *et al*., 2018), machine learning-based methods (Kneifl *et al*., 2024) or hybrid approaches combining both (Czech *et al*., 2022; Fresca *et al*., 2022).

Additionally, the method of dynamic substructuring with superelements, as described in Cammarata *et al*. (2019), Allen *et al*. (2020) can be employed. Here, dofs that are not relevant to other parts of the model are removed from the model by static condensation, see Wilson (1976). Similarly, the Guyan reduction addresses the same issue, as detailed by Guyan (1965). These substructures can now be solved once in an offline process and then reused on numerous occasions. Such a pre-solved part is called a superelement. This method is applicable to circular symmetric geometries where the connection to the other parts of the model or the boundary condition is known and unaltered. Consequently, it is not possible to adapt a superelement to other models, which limits their flexibility. This is intended to be circumvented by the proposed method of using machine learning as described in the following sections. It is important to note that substructuring does not introduce approximations beyond those inherent to the FE formulation itself, in contrast to the majority of machine learning approaches.

Furthermore, the multiscale FEM can also be employed to reduce computational costs, as demonstrated by Chung *et al*. (2023). In multiscale finite element simulations, a problem is initially solved based on a coarse discretization. Each component or element of the coarse solution is simulated again using a fine discretization. In this context, machine learning methods are described in Nguyen *et al*. (2023), Koeppe *et al*. (2020). In many instances, the calculation of the coarse discretization is not performed with regular finite elements, moreover, a method known as the unit displacement method is used (Pourazarm *et al*., 2011; Koeppe *et al*., 2020). A given part of the model in the macro scale is defined by a number of boundary nodes. For each of these nodes, an incremental element nodal force is evaluated for small unit deformations. This approach deviates from conventional FEM because shared shape functions across multiple elements, which are required to form the macroelement, are not available. In essence, the computation of the macroelement is analogous to the calculation of Craig–Bamptons interface modes, as elucidated by Craig and Bampton (1968). In recent research work, the fine discretization is modelled by artificial neural networks (ANN), as summarized in the review work of Bishara *et al*. (2023), or deep learning as demonstrated in Koeppe *et al*. (2020), Deng *et al*. (2024). However, at first the stiffness properties of the macroelements must be characterized. Huang *et al*. (2023) present a method for training an ANN to predict stiffness matrices for a coarse mesh and corresponding shape functions for substructure modeling. The deep NN uses the Young's modulus at each node of the coarse mesh as input for both the stiffness matrices and shape functions. As a result, these models are adaptable to various materials within the offline-trained range of Young's modulus. The stiffness matrix and shape functions are subsequently mapped to the specified material property. Beyond the context of substructuring, but also constructing stiffness matrices of quadrilateral finite elements, Jung *et al*. (2020) use deep learning and a large amount of strain data to corresponding material properties, displacements and geometries. Capuano and Rimoli (2018) employ machine learning techniques to establish a relationship between the element

state and its forces in multiscale simulations. This approach eliminates the need for element integration. The stiffness matrix is generated by applying unit loads to the nodes of the coarse mesh, identical to standard multiscale simulations. However, the stiffness matrix is then calculated by machine learning.

The identification of stiffness matrices falls within the domain of inverse problems. In accordance with the classification proposed by Beck and Woodbury (1998), inverse problems can be divided into two categories, *measurement* problems, which pertain to parameter identification (Chamekh *et al*., 2009; Römer *et al*., 2024) and *design* problems, as exemplified by Fachinotti *et al*. (2020). *Design* problems are formulated for cases where the underlying data is explicitly and accurately defined. In the following we focus on *design* problems, according to our use case. Solving inverse problems of partial differential equations (PDE) with (physically informed) NN is a topic that has been extensively discussed in the literature (Zhang *et al*., 2019; Berg and Nyström, 2021; Badia *et al*., 2024; Raissi *et al*., 2019). With regard to the calculation of stiffness matrices, it is possible for an ANN to work at different level. Oishi and Yagawa (2017) propose a replacement of the standard element integration routine. Still working with *Gauss-Legendre* quadrature, the weights and integration points are adjusted to an optimum for different cases. These changes are controlled by deep learning networks. From a more comprehensive perspective, the *data-driven computational mechanics paradigm* approach (Kirchdoerfer and Ortiz, 2016; Nguyen *et al*., 2020), also searches a stiffness matrix. All elements within a simulation are fit to existing (experimental) stress strain curves while satisfying compatibility and equilibrium due to the Lagrange multiplier method. Hence, the measured material data can be utilized directly in computational models. The *data-driven computational mechanics paradigm* also applies to multiscale simulations, as proposed by Karapiperis *et al*. (2021). The identification of stiffness matrices using ANNs as an inverse problem is described in Meethal *et al*. (2023). In this context, the stiffness matrix of a physical model includes unknown components. It is therefore divided into a known part, an unknown part and two corresponding unknown cross-coupling components. The unknown portion of the stiffness matrix is determined using an ANN. Optimization of the unknown entries is achieved using the known force vector and displacement vector. However, it is important to note that the resulting predicted stiffness matrix is not compatible with other models and cannot be directly combined with other stiffness matrices.

In this work we propose a method, where the ideas behind the substructuring and the multiscale method are combined. Macroelements or superelements are created for repeated parts of a model, which can be used in a more flexible manner than substructuring or static condensation in different models. For this purpose, stiffness matrices of repeated parts of a large model are generated in a reduced manner. This implies that only a subset of the original number of dofs is utilized in the surrogate elements stiffness matrix. A significant focus is placed on positive semi-definite matrices, which align with conventional finite element. For this, we introduce *artificial degrees of freedom* (*adofs*). To reduce the training effort of the surrogate element matrices, the identification is conducted *in silico* on models with a low number of dofs. Subsequently, the surrogate element is used to model the repeating parts of large models as a forward problem. From a connectivity perspective, the stiffness matrix of the surrogate element can be treated in the same manner as a standard finite element, allowing it to be added to the dofs of adjacent elements. Consequently, no explicit coupling terms are required to connect two regions. The surrogate element behaves similar to macroelements known from multiscale FEM, wherein information on the region described by the element is only provided at the boundary. Nevertheless, the structural properties of the entire element are considered with high accuracy. As a benefit, calculation time is considerably reduced.

The proposed method is related to the work of Huang *et al*. (2023) and the work of Capuano and Rimoli (2018). Regarding the work of Huang *et al*. (2023), the ANN uses the material parameter as input, whereas our approach does not require input, as only the bias is trained. Hence, our architecture of the ANN is less complex, which benefits the training and prediction

costs. A limitation of our approach is that a separate model must be trained for each material. This restriction applies unless the ANN is provided with material-specific inputs, which parametrize the selected material model of the training data. But in scenarios where the material is predetermined, this issue is not a significant concern. In addition, our approach is applied to transient simulations. Regarding Capuano and Rimoli (2018), the main differences compared to the proposed method here, lies in the application to solid elements and the direct training of the surrogate elements on simulation data. Hence, in our approach no additional load cases for unit forces or displacements methods are necessary. Furthermore, in both works all elements within the simulations are replaced by surrogate elements, which are named coarse elements and smart elements in their publications. Novel contributions of the proposed method are the combination of the initial discretization with surrogate elements within one simulation and the introduction of *artificial degrees of freedom*.

The article is organized as follows: The initial section presents the fundamental principles of FEM and surrogate modeling. Subsequently, in Section 2.2, a comprehensive analysis of the ANN responsible for generating the matrices is presented. Section 2.3 describes in detail the concept of *adofs*. This is followed by Section 2.4 and 2.5, which present approaches for the generation of the surrogate elements stiffness- and mass matrices. Section 2.6 and 2.7 describe the data-generation routines and the training process, respectively. Later, in Section 3, at first results for the low dimensional training examples are presented. Thereafter, the application of the surrogate elements to large examples, which utilize several surrogate elements for modeling the problems, are discussed. There, we also focus on the application of the surrogate elements to complex dynamic structures. Finally, a conclusion is drawn in Section 4.

2. Surrogate modeling

In order to facilitate the surrogate modeling of mechanical structures, we propose the introduction of surrogate elements, which are described by stiffness matrices and mass matrices that depend on the adjacent elements. The following section will provide a detailed description of the surrogate elements. Furthermore, the architecture and training of the ANN generating the surrogate elements are discussed.

2.1 Surrogate elements

The discretized linear form of the partial differential equation (PDE) describing the motion of a mechanical system is given by

$$\boldsymbol{M}\ddot{\boldsymbol{u}}(t) + \boldsymbol{C}\dot{\boldsymbol{u}}(t) + \boldsymbol{K}\boldsymbol{u}(t) = \boldsymbol{f}(t), \tag{1}$$

see Hughes (1987). Thereby, $\boldsymbol{u}(t) \in \mathbb{R}^{N\times 1}$ and its derivatives contain the nodal displacements, velocities and accelerations. The variable N denotes the total number of dofs. Moreover, $\boldsymbol{f}(t) \in \mathbb{R}^{N\times 1}$ is the vector of the nodal forces. Further, $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ denote the stiffness, damping and mass matrix, respectively. If the problem is of a static nature, $\boldsymbol{C}, \boldsymbol{M}, \dot{\boldsymbol{u}}(t)$ and $\ddot{\boldsymbol{u}}(t)$ are neglected. Also the time dependency of $\boldsymbol{u}$ and $\boldsymbol{f}$ is neglected, when loads and boundary conditions are not time dependent. Thus, in the following static cases, they appear without (t). The quadratic and symmetric matrices are created by adding element matrices, for example $\boldsymbol{K}^{\mathrm{e}} \in \mathbb{R}^{N_{\mathrm{e}}\times N_{\mathrm{e}}}$ with N_{e} as number of dofs per element, taking into account node connectivity. Element matrices are built with numerical element integration of shape functions. If modeling of the stiffness matrix is correct, all $\boldsymbol{K}^{\mathrm{e}}$ are positive semidefinite (psd). This is due to the fact that these matrices represent the strain energy stored in the structure, which is modelled by the elements. Given that an energy can only be positive or zero, it follows that elements which are not psd cannot model physical systems in a satisfactory manner. This implies that the eigenvalues of the matrices are either zero or positive. Since all $\boldsymbol{K}^{\mathrm{e}}$ are psd, the

global stiffness matrix $\boldsymbol{K}$, which is built by summing all $\boldsymbol{K}^{\mathrm{e}}$ according to their corresponding dofs, is also psd. Before the kinematic boundary conditions are applied, Equation (1) cannot be solved for $\boldsymbol{u}$, because $\boldsymbol{K}$ is singular and hence, $\boldsymbol{K}^{-1}$ cannot be computed. After elimination of supported dofs, the mechanical problem is kinematically determined. That means $\boldsymbol{K}$ is now positive definite and can be inverted.

It is not possible to interpret a surrogate element in the same way as a standard finite element. Moreover, it can be employed to replace several standard finite elements. The construction of a surrogate element does not necessitate the use of shape functions, material models, transformation to local coordinate systems or numerical integration. Consequently, a surrogate element is defined as a region within the matrices that model a specific portion of a domain Ω. The modeling is not of a pure physical manner, instead it is based on a data-driven representation.

To describe the fundamentals of this approach, it is necessary to introduce different regions. In essence, a specific part of the physical domain is substituted with a surrogate element. This region is called the *Replaced Region* and can be found in Figure 1 as Ω_{RR}.

The region which persists is called *Preserved Region*, Ω_{PR}. This region is fully modelled in the discretization process with standard finite elements. Hence, results of that region can be fully interpreted after solving the system of equations. In contrast to that, Ω_{RR} is not fully discretized. Only the interface between the two regions, Γ_{IR}, is discretized. Furthermore, we state that $\Gamma_{\mathrm{IR}} \subset \Omega_{\mathrm{RR}}$ and $\Gamma_{\mathrm{IR}} \subset \Omega_{\mathrm{PR}}$ and therefore, $\Gamma_{\mathrm{IR}} = \Omega_{\mathrm{RR}} \cap \Omega_{\mathrm{PR}}$ and $\Omega = \Omega_{\mathrm{RR}} \cup \Omega_{\mathrm{PR}}$. The selection of Ω_{RR} should be guided by two primary criteria. First, this region should not contain areas of interest, as its results will not be directly calculated. Second, Ω_{RR} should represent a structural component that recurs multiple times within the model to maximize the reduction in system size.

A schematic assembly of the stiffness matrix $\boldsymbol{K}$ from the sub-areas is shown in Figure 2. There, it is assumed that the non-zero coefficients in the global matrix are sorted diagonally dominant. The stiffness matrix $\boldsymbol{K}$ can be subdivided in blocks corresponding to Ω_{PR}, Ω_{RR} and Γ_{IR}. The dofs of the elements, which are not affected of the model reduction process, are part of $\boldsymbol{K}_{\mathrm{PR}}$. Elements, which are not modelled in the reduced stiffness matrix, are part of $\boldsymbol{K}_{\mathrm{RR}}$. Due to the compatibility, there are dofs, which are part of $\boldsymbol{K}_{\mathrm{PR}}$ as well as part of $\boldsymbol{K}_{\mathrm{RR}}$, see $\boldsymbol{K}_{\mathrm{IR}}$. The entries of $\boldsymbol{K}_{\mathrm{IR}}$ created from elements in Ω_{PR} are preserved. In contrast, if they originate from Ω_{RR}, they are deleted.

The reduced stiffness matrix, denoted as $\tilde{\boldsymbol{K}}$, can be built from $\boldsymbol{K}$ by keeping $\boldsymbol{K}_{\mathrm{PR}}$ and replacing $\boldsymbol{K}_{\mathrm{RR}}$ with $\boldsymbol{K}_{\mathrm{SE}}$, the stiffness matrix of the *Surrogate Element*. To achieve this, one has to add $\boldsymbol{K}_{\mathrm{SE}}$ to the corresponding parts of the interface regions, which are indexed with the subscript i, $\boldsymbol{K}_{\mathrm{IR},i}$, see Figure 3.

The dimension of $\boldsymbol{K}_{\mathrm{SE}}$ has to be at least equal to the sum of $\boldsymbol{K}_{\mathrm{IR},i}$ dimensions, which is N_{IR}. Optionally, the dimension of $\boldsymbol{K}_{\mathrm{SE}}$ can be increased by inserting *adofs* from the matrix $\boldsymbol{K}_{\mathrm{AD}}$. Once $\boldsymbol{K}_{\mathrm{SE}}$ is generated to represent a specific $\boldsymbol{K}_{\mathrm{RR}}$, the region Ω_{RR} can not be modified thereafter. Consequently, always the same number of finite elements is replaced by the surrogate element. The reduced mass matrix $\tilde{\boldsymbol{M}}$ is built in an identical manner, whereas $\tilde{\boldsymbol{C}}$ is

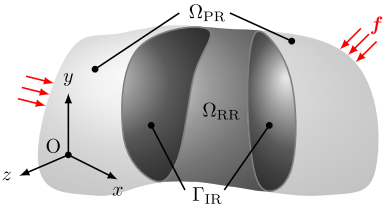

Figure 1. The complete domain Ω can be partitioned into Ω_{PR} and Ω_{RR} by the faces Γ_{IR}. Source: Authors' own work

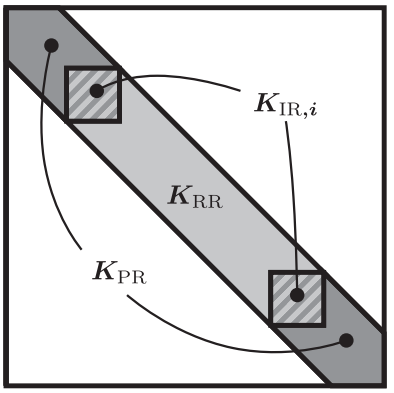

Figure 2. Schematic stiffness matrix $\boldsymbol{K}$ of the fully discretized model. Filled regions denote entries, which are not necessarily zero. The shaded regions, $\boldsymbol{K}_{\mathrm{IR},i}$ with $i = [1,2]$, are part of $\boldsymbol{K}_{\mathrm{PR}}$ and $\boldsymbol{K}_{\mathrm{RR}}$ due to addition of elements. Source: Authors' own work

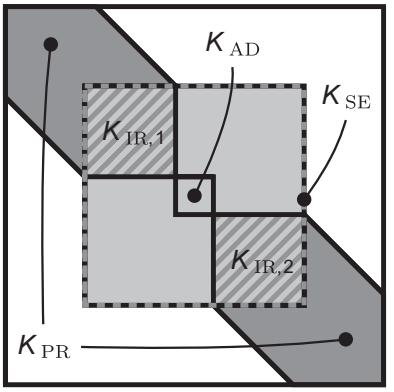

Figure 3. Schematic stiffness matrix $\tilde{\boldsymbol{K}}$ of the reduced discretized model. Filled regions denote entries, which are not necessarily zero. In this setup the optional dimension expansion of $\boldsymbol{K}_{\mathrm{SE}}$ is depicted. The shaded regions, $\boldsymbol{K}_{\mathrm{IR},i}$ with $i = [1,2]$, are part of $\boldsymbol{K}_{\mathrm{PR}}$ and $\boldsymbol{K}_{\mathrm{RR}}$ due to addition of elements which refer to Γ_{IR}. $\boldsymbol{K}_{\mathrm{SE}}$ extends over the entire light grey region. Source: Authors' own work

obtained by the Rayleigh damping approach. A more detailed description on the reduced matrices is given in Sections 2.2, 2.4 and 2.5.

With regard to Equation (1), the system of equations of the reduced model is described by

$$\tilde{\boldsymbol{M}}\ddot{\tilde{\boldsymbol{u}}}(t) + \tilde{\boldsymbol{C}}\dot{\tilde{\boldsymbol{u}}}(t) + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{u}}(t) = \tilde{\boldsymbol{f}}(t), \tag{2}$$

where $\tilde{\boldsymbol{K}} \in \mathbb{R}^{n\times n}$, $\tilde{\boldsymbol{C}} \in \mathbb{R}^{n\times n}$ and $\tilde{\boldsymbol{M}} \in \mathbb{R}^{n\times n}$ as well as $\tilde{\boldsymbol{u}}(t) \in \mathbb{R}^{n\times 1}$, respectively the derivations and $\tilde{\boldsymbol{f}}(t) \in \mathbb{R}^{n\times 1}$ describe the reduced matrices and vectors with n as number of dofs of the reduced approximation.

Standard surrogate models reproduce complete systems and therefore, their application is limited to the very same systems. In contrast, the proposed surrogate elements require only the interface elements to fit to the trained system if a different model is considered. Next to Γ_{IR}, an arbitrary region can be modelled. Hence, the offline trained surrogate elements are independent of the boundary conditions applied to an arbitrary model. In order to achieve a

MOR character, it is necessary that the dimension of $\boldsymbol{K}_{\mathrm{SE}}$ is smaller than the dimension of $\boldsymbol{K}_{\mathrm{RR}}$, so that $n < N$.

2.2 Architecture of the neural net

The stiffness and mass matrices are independently modelled using two separate dense layers, which are designed without interconnections. This separation is necessitated by the significantly different scales and distinct properties of the two matrices. An overview of the ANN is shown in Figure 4.

First we will focus on the stiffness matrix. The method will be introduced to be theoretically capable of handling geometrically nonlinear elastic problems for future work, although it will be applied only to linear elastic problems in this article.

As an input, the displacement vector from the dofs in $\boldsymbol{K}_{\mathrm{IR}}$ of the previous iteration, schematically denoted as $\boldsymbol{u}_{\mathrm{IR}}(-1)$, is utilized. This vector is processed by a dense layer and an activation function, see Goodfellow *et al.* (2016)

$$\boldsymbol{k}_{\mathrm{SE}} = \tanh(\boldsymbol{W}_{\mathrm{K}}\boldsymbol{u}_{\mathrm{IR}}(-1) + \boldsymbol{b}_{\mathrm{K}}) \tag{3}$$

with $\boldsymbol{k}_{\mathrm{SE}} \in \mathbb{R}^{n_{\mathrm{b}} \times N_{\mathrm{SE}}}$ where n_{b} denotes the batch size and N_{SE} is the number of the surrogate elements dofs. The weight-matrix of the dense layer for the stiffness matrix is labelled $\boldsymbol{W}_{\mathrm{K}}$, whereas the bias-vector is denoted $\boldsymbol{b}_{\mathrm{K}}$. Furthermore, the hyperbolic tangent activation function is defined by $\tanh = (e^{x} - e^{-x})/(e^{x} + e^{-x})$. This function is selected because it produces both negative and positive values, which is essential for accurately modeling the stiffness properties.

From this point onward, the method is applied specifically to linear elastic modeling. In this context, no previous iterations are required as the system is solved once using the undeformed geometry, under the assumption of small deformations. Consequently, the input data is set to $\boldsymbol{u}_{\mathrm{IR}}(-1) = \boldsymbol{0}$. In general, this represents the displacements from a previous iteration in nonlinear problems, denoted as (-1). Hence, even for linear transient simulation $\boldsymbol{u}_{\mathrm{IR}}(-1)$ is zero. The input of the ANN is multiplied element-wise with the weights $\boldsymbol{W}_{\mathrm{K}}$. As a result, the calculation of $\boldsymbol{k}_{\mathrm{SE}}$ is independent of the weights $\boldsymbol{W}_{\mathrm{K}}$ and depends solely on the bias $\boldsymbol{b}_{\mathrm{K}}$. In summary, this reduces the problem to a straightforward optimization task, where a single dense layer is sufficient.

In a similar way, the surrogate element mass matrix is constructed. Unlike the stiffness matrix, the mass matrix only allows positive entries. Hence, Equation (3) is adapted to utilize a sigmoid activation function $\sigma = 1/(1 + e^{-x})$,

$$\boldsymbol{m}_{\mathrm{SE}} = \sigma(\boldsymbol{W}_{\mathrm{M}}\boldsymbol{u}_{\mathrm{IR}}(-1) + \boldsymbol{b}_{\mathrm{M}}) \tag{4}$$

where $\boldsymbol{W}_{\mathrm{M}}$ is the mass matrix dense layer weight matrix and $\boldsymbol{b}_{\mathrm{M}}$ the bias vector.

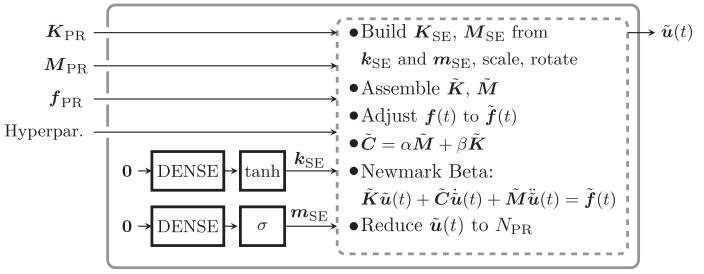

Figure 4. Architecture of the informed ANN for transient simulations. Source: Authors' own work

A more detailed description is required for the FEM block of the architecture, which is shown on the right in Figure 4. In this instance, the ANN is informed, as the system of equations comprising the discretized model is solved. Initially, the stiffness matrix $\boldsymbol{K}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{SE}} \times N_{\mathrm{SE}}}$ of the surrogate element has to be built. This is explained in detail in Section 2.4. Next, $\tilde{\boldsymbol{K}}$ is assembled from $\boldsymbol{K}_{\mathrm{SE}}$ and $\boldsymbol{K}_{\mathrm{PR}}$. Similarly, if the analysis is not of a static nature, the mass matrix is assembled. After that, the reduced force vector $\tilde{\boldsymbol{f}}$ is built from the original force vector $\boldsymbol{f}$ by adjusting dimensions. If required for transient simulations, the reduced damping matrix $\tilde{\boldsymbol{C}}$ is calculated following the Rayleigh damping approach, see Wagner (2022),

$$\tilde{\boldsymbol{C}} = \alpha_d \tilde{\boldsymbol{M}} + \beta_d \tilde{\boldsymbol{K}} \tag{5}$$

where α_d denotes the mass proportional damping factor and β_d denotes the stiffness proportional damping factor.

In the static case, where $\tilde{\boldsymbol{K}}\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}}$ has to be solved for $\tilde{\boldsymbol{u}}$, the *Cholesky decomposition*, is used, see Wagner (2022). We choose this specific method for solving the system, since every physical system modelled correctly with FEM has to be psd, which is ensured by successfully solving using the *Cholesky decomposition*. Due to this, solving with a multifrontal method is also possible. Furthermore, another widely known solving method, the conjugate gradient method, also requires the use of psd matrices. While iterative methods like the preconditioned conjugate gradient and generalized minimal residual methods do not require positive-definite matrices, solution techniques relying on positive-definite matrices remain central to finite element simulations. This highlights the importance of our focus on psd surrogate element matrices. For transient problems, Equation (2) is frequently integrated over time using an unconditionally stable implicit Newmark Beta scheme; see Hughes (1987).

The surrogate element is trained on simple models, which contain only one surrogate element, see Section 2.6 and 2.7. Nevertheless, in the inference phase it should be possible to place the surrogate element several times in the same model in different orientations and locations. Since training of each orientation is not practical, the surrogate element is transformed from the training orientation to the desired orientation. For this, we use Euler angles ψ, θ, ϕ to rotate around the z, x' and z'' axis with the rotation matrices $\boldsymbol{R}$:

$$\boldsymbol{K}_{\mathrm{SE,\,dof,\,oriented}} = (\boldsymbol{R}_{\mathrm{z}}(\psi)\boldsymbol{R}_{\mathrm{x'}}(\theta)\boldsymbol{R}_{\mathrm{z''}}(\phi))\boldsymbol{K}_{\mathrm{SE,\,dof}}(\boldsymbol{R}_{\mathrm{z}}(\psi)\boldsymbol{R}_{\mathrm{x'}}(\theta)\boldsymbol{R}_{\mathrm{z''}}(\phi))^{\mathrm{T}}. \tag{6}$$

Equation (6) is performed on $\boldsymbol{K}_{\mathrm{SE,\,dof}}$, which denotes the parts of $\boldsymbol{K}_{\mathrm{SE}}$ separated by the assignment of dofs in $\boldsymbol{K}_{\mathrm{SE}}$ to their nodes. For a better readability, in the following the oriented matrices, $\boldsymbol{K}_{\mathrm{SE,\,dof,\,oriented}}$, are named $\boldsymbol{K}_{\mathrm{SE}}$. Equation (6) applies identically to $\boldsymbol{M}_{\mathrm{SE}}$.

The last step in the model is only necessary for the training process. From the vector of displacements, $\tilde{\boldsymbol{u}}$, only those dofs that originate from Ω_{PR} are extracted for the loss calculation. The forward propagation of the ANN is described in detail in Algorithm 1.

2.3 Artificial degrees of freedom

As depicted in Figure 3, it is possible to incorporate *adofs as* $\boldsymbol{K}_{\mathrm{AD}}$. Consequently, the number of instances of $\boldsymbol{K}_{\mathrm{SE}}$ is increased, thereby enhancing the ability to adequately model Ω_{RR}. The placement of the *adofs* can be arbitrary; we chose rows at the right and consequently columns at the bottom of $\boldsymbol{K}_{\mathrm{SE}}$. Noteworthy, the dimension of the surrogates stiffness matrix is increased by the number of *adofs* N_{AD}, such that $\boldsymbol{K}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{IR}}+N_{\mathrm{AD}} \times N_{\mathrm{IR}}+N_{\mathrm{AD}}}$. As previously stated, $\boldsymbol{K}_{\mathrm{SE}}$ is then added to an enlarged version of $\boldsymbol{K}_{\mathrm{PR}}$ to construct $\tilde{\boldsymbol{K}}$. Due to this enlargement, there are no underlying entries of $\boldsymbol{K}_{\mathrm{PR}}$ where $\boldsymbol{K}_{\mathrm{AD}}$ is added. The initialization of $\boldsymbol{K}_{\boldsymbol{AD}}$ with small random values can give rise to two potential issues:

(1) The system of equations cannot be solved, despite sufficient boundary conditions.

(2) The positive-definiteness of $\boldsymbol{K}_{\mathrm{SE}}$ may be compromised when eliminating dofs associated with the boundary condition. On the beam example, this will happen when the boundary condition is only set on one side of the surrogate element. The introduction of *adofs* with a zero initialization will result in the formation of two non-connected regions, which renders the solution of the problem impossible.

Therefore, initialization of $\boldsymbol{K_{AD}}$ is done with

$$\boldsymbol{K}_{\mathrm{AD}} = \boldsymbol{I} f_{\mathrm{i}} \tag{7}$$

where $\boldsymbol{I} \in \mathbb{R}^{N_{\mathrm{AD}} \times N_{\mathrm{AD}}}$ is the identity matrix and f_{i} denotes an additional hyperparameter for the training process. Consequently, the unit of f_{i} is identical to the unit of $\boldsymbol{K_{AD}}$, which depends on the unit system of the training dataset. It is recommended to chose values of the magnitude of the diagonal elements from $\boldsymbol{K_{PR}}$ for f_{i}.

Before training, *adofs* are not coupled with FE dofs. The coupling is created during the training process, whereby off-diagonal entries are assigned with values mostly $\neq 0$ obtained from $\boldsymbol{k_{SE}}$. These values are then assigned with a stiffness with respect to FE dofs. The *adofs* are not identifiable with a specific location within Ω, and therefore cannot be included in any subsequent interpretation of the results. This approach is analogous to the hierarchic FEM, where dofs assigned to higher order polynomials are also incapable of forming interpretable location and result pairs, as no additional nodes are created and the dofs are purely mathematical in nature, see Zienkiewicz *et al*. (1983).

2.4 Generation of the surrogate element stiffness matrix

The stiffness matrix of a surrogate element should generally possess the same properties as that of a standard finite element. Hence, after adding the surrogate element $\boldsymbol{K}_{\mathrm{SE}}$, $\tilde{\boldsymbol{K}}$ still has to be psd. Furthermore, following the elimination of the boundary dofs, $\tilde{\boldsymbol{K}}$ has to be positive-definite. Ensuring this property, it is possible to solve the system of equations with the *Cholesky decomposition* as mentioned in Section 2.2. Therefore, $\boldsymbol{K}_{\mathrm{SE}}$ has to be symmetric and psd. In addition, to construct physical element stiffness matrices, the requirements can be enriched by positive non-zero main diagonal values, negative off-diagonal values and zero row sums. This implies that the sum of each row of the matrix must be zero.

The values in $\boldsymbol{k}_{\mathrm{SE}}$ are out of the interval $[-1, 1] := \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}$. Hence, they have a different magnitude compared to the values in $\boldsymbol{K}_{\mathrm{PR}}$. To adjust that, $\boldsymbol{k}_{\mathrm{SE}}$ is scaled by

$$\boldsymbol{k}_{\mathrm{SE,\,scaled}} = \mathrm{sgn}(\boldsymbol{k}_{\mathrm{SE}}) e^{f_{\mathrm{k}} \boldsymbol{k}_{\mathrm{SE}}} \tag{8}$$

with f_{k} as an dimensionless additional hyperparameter. Consequently, a minimum or maximum value, defined by $e^{f_{\mathrm{k}}}$, is nonlinearly scaled by the dense layer. It is challenging to maintain accurate records of the units in question. Consequently, in this approach, the unit of $\boldsymbol{k}_{\mathrm{SE,\,scaled}}$ is defined as a stiffness in the chosen unit system.

At first, we do not apply the method of *artificial degrees of freedom*. In order to ensure symmetry, only a lower triangular matrix of the surrogate element is formulated in the ANN. For this, the vector $\boldsymbol{k}_{\mathrm{SE,\,scaled}}$ is resorted to build the lower triangular matrix $\boldsymbol{k}_{\mathrm{SE,\,tri.}} \in \mathbb{R}^{N_{\mathrm{SE}} \times N_{\mathrm{SE}}}$. Furthermore, this approach resulted in reduced computational costs during the training and prediction phase of the ANN, in comparison to modeling a quadratic matrix. The number of output parameter of the dense layer, N_{kSE}, is reduced, such that $\boldsymbol{k}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{kSE}} \times 1}$ with

$$N_{\mathrm{kSE}} = \frac{1}{2} N_{\mathrm{SE}}^2 + N_{\mathrm{SE}}. \tag{9}$$

To ensure positive values on the main diagonal and symmetry, $\boldsymbol{K}_{\mathrm{SEmm}} \in \mathbb{R}^{N_{\mathrm{SE}} \times N_{\mathrm{SE}}}$ is calculated,

$$\boldsymbol{K}_{\mathrm{SEmm}} = \boldsymbol{k}_{\mathrm{SE,\,tri.}} \boldsymbol{k}_{\mathrm{SE,\,tri.}}^{T}. \tag{10}$$

As a result, non-diagonal entries may still retain negative values. The unit of $\boldsymbol{k}_{\mathrm{SE}}$ has to be expressed as the square root of a stiffness in the chosen unit system to result in a meaningful matrix $\boldsymbol{K}_{\mathrm{SEmm}}$. For psd, all eigenvalues λ of the unsupported $\tilde{\boldsymbol{K}}$ have to be $\lambda \geq 0$. The fulfilment of this requirement necessitates a specific approach, since $\boldsymbol{K}_{\mathrm{SE}}$ is optimized by the ANN with regard to the displacements $\tilde{\boldsymbol{u}}$, rather than the positive eigenvalues.

In general, there are numerous methods for the optimization of eigenvalues as shown by Lewis (2003). The implementation of these methods into our workflow is challenging in certain instances. Hence, we seek a more straight-forward solution. In the field of comparing measurements, it is also necessary to deal with psd matrices. In this context, it is beneficial to search for the nearest psd matrix, as was done by Higham (1988). Unfortunately, this approach is not applicable in our problem, as the physical models are no longer adequately described following the search for a substitution matrix.

In order to generate psd matrices, the final requirement is assessed, which primarily should ensure force equilibrium. For this, the sum of the row-wise entries in an element stiffness matrix must be zero. We force the surrogate element to inherit that property. Hence, the resulting matrix corresponds to a *Laplacian* matrix, which is psd. For the row-wise zero sums, the final three columns of $\boldsymbol{K}_{\mathrm{SE}}$, and due to symmetry the final three rows too, are not modelled directly by $\boldsymbol{k}_{\mathrm{SE}}$. Moreover, $\boldsymbol{K}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{IR}} \times N_{\mathrm{IR}}}$ is initialized as a zero matrix and $\boldsymbol{K}_{\mathrm{SEmm}}$ is added to the left top corner of $\boldsymbol{K}_{\mathrm{SE}}$. This approach allows for the reduction of the number of ANN output parameter

$$N_{\mathrm{kSE}} = \frac{1}{2}(N_{\mathrm{SE}} - 3)^2 + (N_{\mathrm{SE}} - 3). \tag{11}$$

The last columns are then populated with the negative column- and component-wise sums of $\boldsymbol{K}_{\mathrm{SE}}$. More precisely, in the three-dimensional representation, all dofs corresponding to the x, y and z-direction are added separately. Subsequently, in analogy, the last row is filled with the negative row-wise sum. Due to the summation process, the values of the last three rows and columns exhibit a different scale, especially entries in the lower right are most different. This distinctive configuration is prone to generating local anomalies in the results at these nodes.

By combining the method presented in Section 2.4 with *adofs*, this effect can be mitigated. It is possible to store the row- and column-wise sums on dofs that are not part of the interpretable output displacement vector. Therewith, the anormal values in the last three columns and rows can be transferred to dofs, which are not part of the physical displacements. Hence, they will not be evaluated. As an acceptable downside, the number of ANN output parameter increase once more, see Equation (9). Consequently, the surrogate elements matrix is now of the dimension $\boldsymbol{K}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{IR}}+3 \times N_{\mathrm{IR}}+3}$, since $N_{\mathrm{AD}} = 3$. Such a surrogate elements stiffness matrix is shown in Figure 5.

The procedure outlined in Section 2.2 and the steps described here are summarized in Algorithm 1, which illustrates the complete forward propagation of the network. Such a forward propagation is also part of the inference phase, when a simulation with surrogate elements is performed. The main steps in the linear case are the following:

(1) Propagation of the network (export bias for linear models: $\boldsymbol{u}(-1) = \boldsymbol{0}$)

(2) Scaling and transformation of the networks output

(3) Computations of surrogate elements matrices

(4) Computation of global matrices

(5) Static or dynamic solving

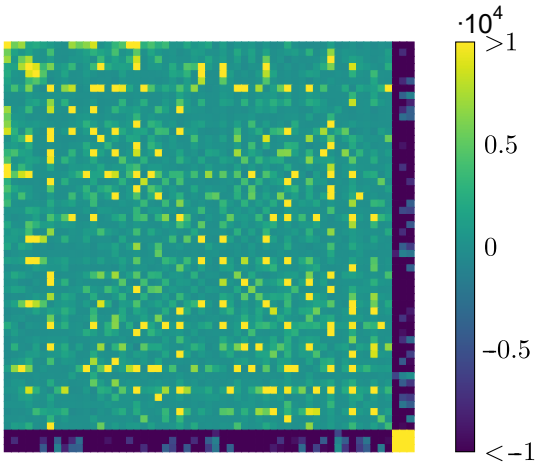

Figure 5. Stiffness matrix of a surrogate element with 54 dofs and three additional *adofs*, $N_{\mathrm{SE}} = 57$. The row- and column-wise sums are placed on the *adofs*. Hence, they are not interpretable as displacements. Source: Authors' own work

Algorithm 1. Forward propagation of the ANN for a static simulation

$\boldsymbol{u}_{\mathrm{IR}-1} \Leftarrow \mathbf{0}$ #Input for linear modeling

$\boldsymbol{K}_{\mathrm{SE}} \Leftarrow \mathbf{0}$

$\tilde{\boldsymbol{K}} \Leftarrow \mathrm{adapt\_dimensions}(\boldsymbol{K}_{\mathrm{PR}})$

$\tilde{\boldsymbol{f}} \Leftarrow \mathrm{adapt\_dimensions}(\boldsymbol{f})$

$\boldsymbol{k} \Leftarrow \mathrm{DENSE}(\boldsymbol{u}_{\mathrm{IR}-1})$

$\boldsymbol{k} \Leftarrow \tanh(\boldsymbol{k})$

$\boldsymbol{k} \Leftarrow \mathrm{sign}(\boldsymbol{k}) e^{f_{\mathrm{k}}\boldsymbol{k}}$ #Scaling

$\boldsymbol{a}, \boldsymbol{b} = \mathrm{indices\_lower\_triangular\_matrix}(N_{\mathrm{SE}} - 3)$

$\boldsymbol{K}_{\mathrm{SE}}(\boldsymbol{a}, \boldsymbol{b}) \Leftarrow \boldsymbol{k}$

$\boldsymbol{K}_{\mathrm{SE}}(\boldsymbol{b}, \boldsymbol{a}) \Leftarrow \boldsymbol{k}$

$\boldsymbol{K}_{\mathrm{SE}} \Leftarrow \boldsymbol{K}_{\mathrm{SE}} \boldsymbol{K}_{\mathrm{SE}}^{\mathrm{T}}$

$\boldsymbol{K}_{\mathrm{SE}} \Leftarrow \boldsymbol{R}(\psi, \theta, \varphi) \boldsymbol{K}_{\mathrm{SE,\ dof}} \boldsymbol{R}(\psi,\theta,\varphi)^{\mathrm{T}}$ #Correct orientation of the SE

$\boldsymbol{K}_{\mathrm{SE,\ idx\_right\_column}} \Leftarrow \mathrm{sum\_comp\_row}(\boldsymbol{K}_{\mathrm{SE}})$

$\boldsymbol{K}_{\mathrm{SE,\ idx\_bottom\_row}} \Leftarrow \mathrm{sum\_comp\_col}(\boldsymbol{K}_{\mathrm{SE}})$

$\boldsymbol{h}, \boldsymbol{p} = \mathrm{indices\_interface\_regions}(\Gamma_{\mathrm{IR}})$

for i **do** #Add SE to conventional elements

    $\tilde{\boldsymbol{K}}(\boldsymbol{h}(i), \boldsymbol{p}(i)) \Leftarrow \boldsymbol{K}_{\mathrm{SE}}$

end for

$\boldsymbol{L} \Leftarrow \mathrm{chol}(\tilde{\boldsymbol{K}})$

20: $\boldsymbol{\alpha} \Leftarrow \boldsymbol{L}^{-1}\tilde{\boldsymbol{f}}$

21: $\tilde{\boldsymbol{u}} \Leftarrow (\boldsymbol{L}^{\mathrm{T}}\boldsymbol{\alpha})^{-1}$

22: $\tilde{\boldsymbol{u}} \Leftarrow \tilde{\boldsymbol{u}} (\neq \text{additional dofs})$ #Remove adofs

2.5 Generation of the surrogate element mass matrix

The mass matrix of finite element simulations describes the distribution of the mass of the domain Ω to the dofs. As a surrogate element should not change properties of the standard finite element matrices, a surrogate element mass matrix must also possess two essential properties. These are symmetry and strictly positive entries, due to the non-physicality of negative masses. The positive entries of the surrogate element mass matrix are ensured by applying a sigmoid activation function after the dense layer, see Equation (4). This vector is then multiplied by the materials density ρ. Hence, the surrogate elements are not dependent on the density in the training dataset. The problems under investigation are of a three-dimensional structural nature, so that each dof is assigned to a directional component, *x*, *y* or *z*. Unlike stiffness matrices, where *x*, *y* and *z* components are not independent due to the material law, this is the case for mass matrices. Taking advantage of this circumstance, the dimension of $\boldsymbol{m}_{\mathrm{SE}}$ does not have to be N_{SE}^2, it can be reduced. The values from $\boldsymbol{m}_{\mathrm{SE}}$ are sorted in the following manner:

$$\boldsymbol{M}_{\mathrm{SE}} = \begin{bmatrix} m_{\mathrm{SE},1} & 0 & 0 & \dots & m_{\mathrm{SE},i} & 0 & 0 \\ 0 & m_{\mathrm{SE},i+1} & 0 & \dots & 0 & m_{\mathrm{SE},j} & 0 \\ 0 & 0 & m_{\mathrm{SE},j+1} & \dots & 0 & 0 & m_{\mathrm{SE},N_{\mathrm{SE}}} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ m_{\mathrm{SE},i} & 0 & 0 & \dots & m_{\mathrm{SE},N_{\mathrm{kSE}}-2} & 0 & 0 \\ 0 & m_{\mathrm{SE},j} & 0 & \dots & 0 & m_{\mathrm{SE},N_{\mathrm{kSE}}-1} & 0 \\ 0 & 0 & m_{\mathrm{SE},N_{\mathrm{SE}}} & \dots & 0 & 0 & m_{\mathrm{SE},N_{\mathrm{kSE}}} \end{bmatrix}, \tag{12}$$

where $m_{\mathrm{SE},i}$ is the i-th component of $\boldsymbol{m}_{\mathrm{SE}}$, with $\boldsymbol{m}_{\mathrm{SE}} \in \mathbb{R}^{N_{\mathrm{kSE}}/3 \times 1}$. A symmetric $\boldsymbol{M}_{\mathrm{SE}}$ is ensured due to the sorting process. The proposed approach is only applicable for N_{SE} being a multiple of three, which is suitable for the directional components in structural mechanics.

Assembling $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{M}}$ is identical and therefore only explained for the mass matrices. First, conventional elements originating from Ω_{PR} are built and added to $\tilde{\boldsymbol{M}}$. Subsequently, $\boldsymbol{M}_{\mathrm{SE}}$ is added to the entries of $\boldsymbol{M}_{\mathrm{IR}}$ in $\tilde{\boldsymbol{M}}$ and the optional *adofs*. A graphical representation of this procedure is provided in Figure 3.

2.6 Data generation

Prior to the training of the surrogate element, training, validation and test data have to be generated in an *in silico* manner. For this, the full model is solved for several random distributed load cases. Subsequently, the displacement vectors $\boldsymbol{u}(t)$ can be stored as dataset alongside the force vectors $\boldsymbol{f}(t)$. It is important to note that at this stage, a consistent system of units must be selected. Once the training process has commenced, it cannot be altered.

With ANNs it is best to scale the input data to a prescribed range or standard deviation, see Goodfellow *et al*. (2016). In the context of our linear elastic use case, this is not necessary, as the input of the ANN, see Equations (3) and (4), is a constant zero vector, $\boldsymbol{u}(-1) = \boldsymbol{0}$. Nevertheless, we strive for datasets, which lead to a similar size of error in the training process. This is not a trivial matter, since the calculation of the error is not done based on scalable input data of the ANN. In the data generation routine the exciting forces are drawn from an uniform distribution within the range $[-1, 1]$. The resulting vector $\boldsymbol{u}(t)$ is scaled to a prescribed absolute

mean value of the displacements for each calculation in the dataset, so that every dataset has the same mean. Consequently, the force vector is also scaled by the same factor. We also investigated scaling of the maximum displacement to the same value, which results in inferior datasets.

Furthermore, it is important to generate balanced datasets. Thereby, the forces should act equally in all directions across all simulations in the datasets. To achieve this property, each component of a load is mirrored on every plane in the global coordinate system, thereby directing it towards every octant in the 3D domain. Hence, for each load vector, additional seven load vectors are generated.

To illustrate, the data generation process is elucidated in depth with regard to the case study presented in Section 3.1. In general, five types of load cases are under investigation for the beam example, see Figure 6. Noteworthy, no supports or loads are applied in Ω_{RR}, marked with yellow dashed lines in Figure 6, since they will be replaced with the surrogate element in the training process:

(1) Type a: Random support and random force nodes on both halves

(2) Type b: Support on both end faces, force on every node in a random layer

(3) Type c: Support on both end faces, force on every node to apply a moment in a random layer

(4) Type d: Support on one end face, other end face is supported only in the axis of the beam, force on every node in a random layer

(5) Type e: Support on only one end face, force on every node in a random layer

In terms of dataset representativeness, we select load cases that represent the range of practical relevance. Additionally, space-filling considerations are not critical for the application of linear modeling, as the model input is always a zero vector. Hence, modeling is done by the bias, which is independent from the input. Due to linearity, each type of load case and its force directions need not be represented at different load levels, as the ANN-derived stiffness matrices remain constant. Consequently, the dataset size can be relatively small.

2.7 Training

The main objective function of the generation of $\boldsymbol{K}_{SE}$ is to minimize the error between the displacements of the full and reduced model. In respect thereof, only the displacements of the dofs associated with $\boldsymbol{K}_{PR}$ are relevant. Optional dofs from $\boldsymbol{K}_{AD}$ are not evaluated. In the case of static models the error is calculated using the mean squared error (MSE) function:

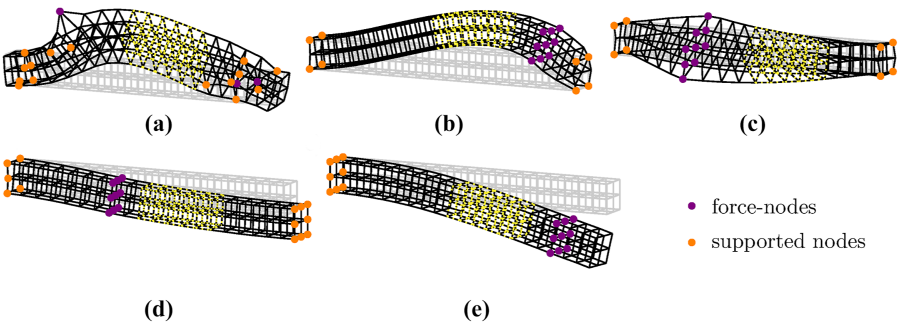

Figure 6. Exemplary boundary conditions and displacement results for all types of loads in the dataset. In every model, 24 dofs are deleted, which corresponds to eight nodes for a fixed support. Yellow dashed lines on the deformed beams mark Ω_{RR}. Source: Authors' own work

$$\epsilon_{\mathrm{MSE,u}} = \frac{1}{n_{\Omega_{\mathrm{PR}}}} \sum_{i=1}^{n_{\Omega_{\mathrm{PR}}}} \left(\boldsymbol{u}_{\mathrm{PR,i}} - \tilde{\boldsymbol{u}}_{\mathrm{PR,i}} \right)^2, \tag{13}$$

where $n_{\Omega_{\mathrm{PR}}}$ is the number of dofs originating from Ω_{PR}. Furthermore, a pure static dataset is used for training, see Section 2.6. The ANN is trained with the Adam optimizer, described by Kingma and Ba (2014).

For training of transient models two different approaches are available. At first, training can be conducted based on the stiffness matrices generated by the aforementioned routine. Hence, only the ANN of the mass matrices has to be optimized in addition. For this, a transient dataset is used. In the forward propagation of the transient training procedure, Equation (2) is solved by the Newmark beta time integration scheme for all timesteps n_{t} in the training dataset. Hence, Equation (13) has to be adapted for the error calculation, since in this procedure $\tilde{\boldsymbol{u}}_{\mathrm{PR}}(t)$ and $\boldsymbol{u}_{\mathrm{PR}}(t)$ can be iterated by $t = [1, n_{\mathrm{t}}]$,

$$\epsilon_{\mathrm{MSE}} = \frac{1}{n_{\mathrm{t}}} \sum_{t=1}^{n_{\mathrm{t}}} \left(\frac{1}{n_{\Omega_{\mathrm{PR}}}} \sum_{i=1}^{n_{\Omega_{\mathrm{PR}}}} \left(\boldsymbol{u}_{\mathrm{PR,i}}(t) - \tilde{\boldsymbol{u}}_{\mathrm{PR,i}}(t) \right)^2 \right). \tag{14}$$

The second approach applies when there is no trained ANN for the stiffness properties. Therefore, both the stiffness and mass matrices must be considered simultaneously. Once more, a transient dataset is required. However, it should be noted that due to the more complex solving process of dynamic simulations, the approach where training is split into a static and dynamic part is less expensive.

3. Results and discussion

In this section, examples of the proposed surrogate elements are presented. At first, only the stiffness matrices are evaluated by static simulations. The second part of the study involves the investigation of transient simulations involving the mass and damping matrix.

3.1 Evaluation of static simulations

The first example to examine is that of a beam structure, in which the central part is replaced by a surrogate element, see Figure 1. The beam has a geometric shape of 100 mm $\times$10 mm $\times$10 mm. The linear elastic material model is parametrized based on the properties of structural steel, young's modulus is $E = 2.1 \times 10^5$ MPa and the Poisson's ratio is $\nu = 0.3$. The full model comprises 540 dofs, where 24 are removed for the boundary condition. All elements are linear, regularly shaped hexahedrons. Element integration is conducted with two integration points per coordinate. Initially, there are 80 elements. For this example, 108 dofs are removed from the system to be later replaced by the surrogate element. In total this are 20 elements, which are marked with yellow dashed lines in Figure 6.

The data generated for the training process consists of 2,048 samples, distributed equally between type (a)–(e). As the model is small, generation of all data points takes 10 s on a single cpu. A total of 70% of the data is allocated for training, 20% for validation and 10% for testing, with the categories assigned randomly. The training dataset is used for training a model, whereas the validation dataset is used to check for overfitting and hyperparameter optimization. The test dataset is utilized to assess the performance of the fully trained model. The mean value for the displacements of all datasets is chosen to be $u_{\mathrm{mean}} = 0.1$ mm. The overall training time is 122 min on a Quadro P2200 graphic card. Furthermore, all neural network related operations are implemented by PyTorch (Paszke *et al.*, 2019). The parameters and the additional hyperparameter f_{k} are shown in Table 1.

Particularly, f_{k}, see Equation (8), is crucial for determining the appropriate stiffness, as low values result in an excessively flexible stiffness matrix. The learning rate α is not constant over

Table 1. Hyperparameters for the training of the static model

Hyperparameter	
Number of epochs	100
Number of updates	9,000
Batch size, n_b	16
Learning rate at epoch 1, α_0	0.01
Learning rate at epoch 101, α_{100}	0.001
Stiffness scaling factor, f_k	20
Size of neural networks output	1,512
Number of *adofs*	3

Source(s): Authors' own work

the epochs. The initial learning rate is gradually decreased linearly to enhance the stability of training and improve the ANNs performance. As illustrated in Figure 7, training for 100 epochs is sufficient, since the loss already has converged at this point.

3.1.1 Single application of a surrogate element. Reference solutions u_i and investigated solutions $\tilde{u}_i$ are compared using a local relative error measure to ensure comparability across different load cases

$$\epsilon_{\text{rel},i} = \frac{u_i - \tilde{u}_i}{u_i} \cdot 100 \tag{15}$$

with i as node number. However, for training a MSE measure is used, since a relative reference is not necessary in the optimization process and MSE is the standard with ANNs in our field. For a global assessment of the results, the mean of the absolute values calculated by Equation (15), $\epsilon_{\text{m.a.r.}}$, is used.

A comprehensive evaluation of the test dataset for each load type reveals low $\epsilon_{\text{m.a.r.}}$ as shown in Table 2. Due to the random nature of load case (a) and the sufficient accuracy, it is evident that the surrogate elements are capable of generalizing to different types of loads and supports.

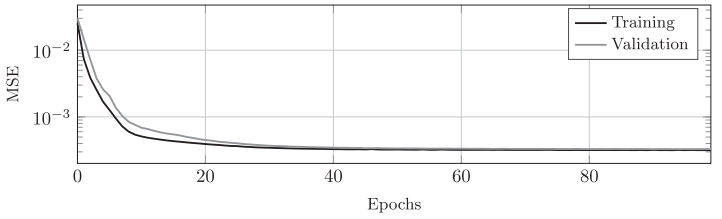

Figure 7. Loss curves of the training for the surrogate elements stiffness matrix over 100 epochs. Source: Authors' own work

Table 2. Global assessment of $\epsilon_{\text{m.a.r.}}$ in percent for a test load case of the types (a)–(e)

Load case (a)	(b)	(c)	(d)	(e)
2.8	0.5	1.6	0.5	2.5

Source(s): Authors' own work

In addition to that, two test load cases, one of type (d) and one of type (b), see Figure 8, are investigated in detail. In the first example, all dofs of the left side of the beam are fixed supported, whereas on the right side, only the dofs in the beams axis are fixed clamped. For the second example, all nodes on the left and right side are fixed clamped. In both cases, the structure is excited on every force node by a force orthogonal to the beams axis with $f_y = 55.6$ N in the first and $f_y = 1667$ N in the second example.

The proposed method yields promising results, as illustrated in Figure 9. Displacements of both test examples cover the reference solution with a very low relative error. In the case of the fixed-sliding example, the reference solution and the surrogate element solution diverge by $\epsilon_{\text{m.a.r.}} = 0.51\%$, while in the case of the fixed-fixed example, the error is even lower, $\epsilon_{\text{m.a.r.}} = 0.48\%$.

In summary, the proposed surrogate element is capable of accurately reproducing the results obtained from a FE simulation.

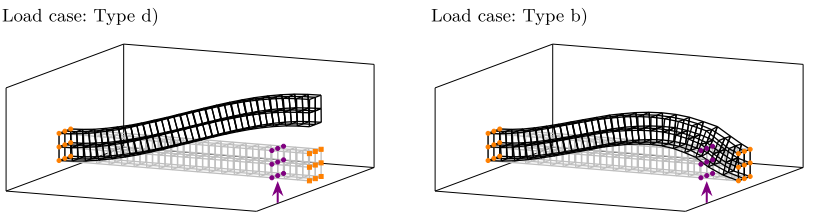

Figure 8. Load cases from the test dataset. The exciting forces are marked purple, the supported nodes orange, where squares support only the component normal to the supported face. The undeformed meshes are colored grey and the reference solutions are displayed black. Source: Authors' own work

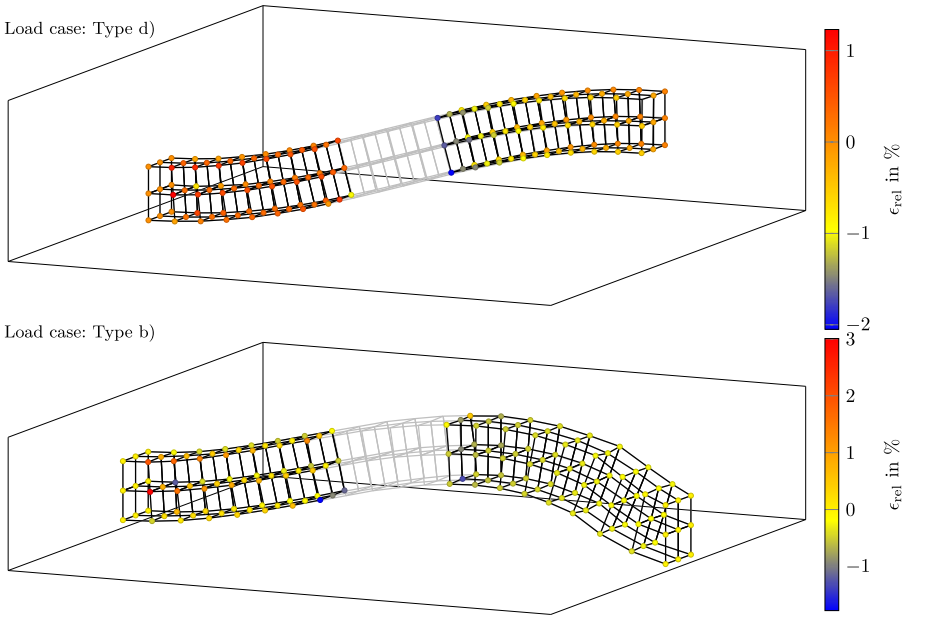

Figure 9. Displacements from the test dataset obtained with a surrogate element. Grey lines display the reference solution, the black meshes depict the solutions obtained with the surrogate elements. Displacements are scaled by the factor 100, $f_{\text{sd}} = 100$. Source: Authors' own work

In this section one more crucial aspect should be addressed. In the case of the beam example, which is fixed supported on both ends (load case d), it is basically possible to replace the surrogate element with four long finite elements. Hence, the model is fully modelled with standard finite elements and has the same number of dofs, and therefore, can be solved by identical computational costs as the model with surrogate elements. But, these four elements show a bad aspect ratio of five. In Figure 10 both versions can be seen. For the surrogate element, a maximum relative error of up to 3.5% can be calculated, whereas for the pure FE solution a higher relative error of 15% is present. It is noteworthy, that for the surrogate element solution, the error in the most deformed region is below 2%, with a majority of errors being below 1%. From a global perspective, $\epsilon_{\text{m.a.r.}} = 7.3\%$ for the pure FE solution and $\epsilon_{\text{m.a.r.}} = 0.47\%$ for the surrogate element solution. In summary, although from a computational point of view, the surrogate element is not advantageous in this comparison, but it clearly outperforms the coarse mesh in terms of precision.

3.1.2 Multiple applications of surrogate elements. The surrogate element demonstrates accurate results for models that are, from a geometric view, identical to the training data. The fundamental concept underlying the surrogate elements is the utilisation of such an artificial element in a multitude of distinct models. So it is to determine if the position of the surrogate element can be altered. Furthermore, if the stiffness behaviour of the region that is replaced with the surrogate element is accurately modelled, it should be possible to use the surrogate element on multiple occasions within the same model.

We use a longer beam with a length of $l = 225$ mm, where the surrogate element is applied five times, see Figure 11. The beam is again fixed at both end faces, a force of $f_z = 222.2$ N is applied on each force node.

The results obtained with the surrogate and FEM reference demonstrate a high degree of accuracy. Nevertheless, $\epsilon_{\text{m.a.r.}} = 3.7\%$ is slightly higher than in other examples. Further studies with a training dataset, wherein the surrogate element is trained on examples with more than one surrogate element in a simulation, yielded more favourable outcomes in this instance.

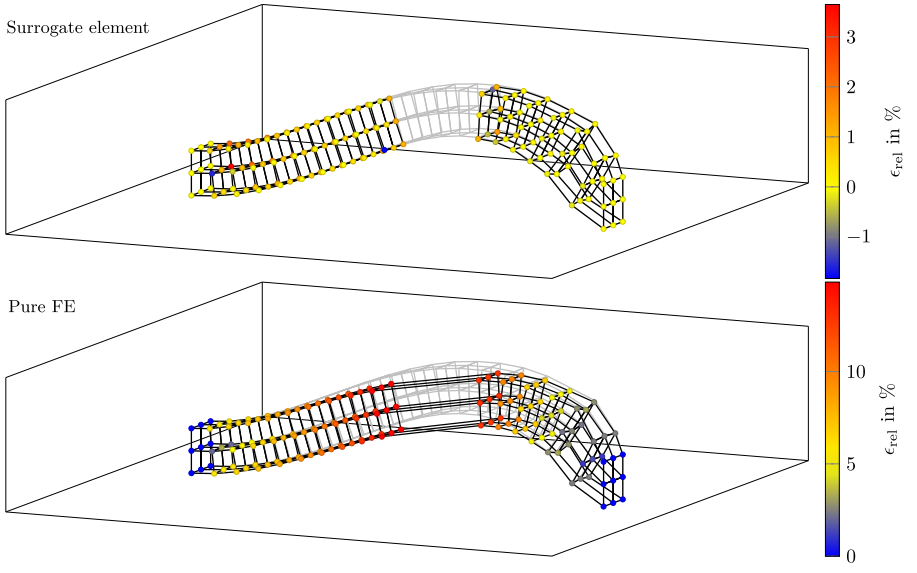

Figure 10. Comparison of the surrogate element with a similar expensive FE model. Grey lines display the reference solution, the black meshes depict the surrogate element solution and the pure FE solution, respectively, $f_{\text{sd}} = 100$. Source: Authors' own work

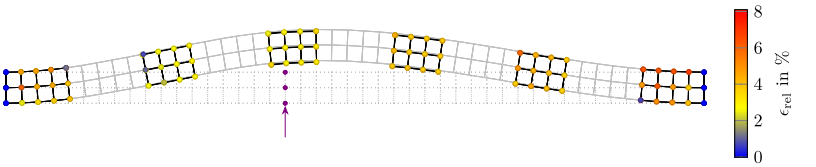

Figure 11. On this elongated three-dimensional beam the surrogate element is applied five times. The reference solution is marked with solid grey lines, the undeformed mesh with dotted grey lines. The surrogate element solution is shown by the black mesh. Purple nodes describe force nodes, $f_{\text{sd}} = 25$. Source: Authors' own work

However, optimising in this manner would result in surrogate elements that are biased towards the specific training data. Conversely, our objective is to facilitate the flexible application of the surrogate elements. This example allows for the assessment of the benefits of such a surrogate element. While $\boldsymbol{K}_{\text{SE}}$ is used several times, it only has to be propagated once. Due to the reduction from $N_{\text{dof, full}} = 1{,}242$ to $N_{\text{dof,SE}} = 702$ dofs, computational time with our in-house FE solver for the long beam example drops from $t_{\text{ref}} = 0.038$ s to $t_{\text{SE}} = 0.024$ s. To ensure comparability, computational times are averaged over ten simulations, with both types of simulations executed on a single CPU processor. Thus, a speed-up factor of 1.6 can be achieved in this example. Overall, assessing the training time of 122 min, a dynamic simulation of > 5.2 s with a timestep of 1×10^{-5} s amortizes the training costs.

In another, example the adaptability of the surrogate element should be discussed in greater detail. The structure under investigation initially comprises 604 elements and 1,332 nodes, see Figure 12. A force of $[222.2, -222.2, 0]$N is applied to each top node of the vertical structures. All dofs at the bottom are fixed. Hence, $\boldsymbol{K} \in \mathbb{R}^{3861 \times 3861}$ can be stated.

The surrogate element allows for the omission of 360 elements from the full model, which represents 1,944 dofs. This results in a reduced stiffness matrix with dimensions $\tilde{\boldsymbol{K}} \in \mathbb{R}^{1917 \times 1917}$. Our in-house FE code requires 1.26 s to solve for $\boldsymbol{u}$, whereas the same solver requires 0.12 s to solve for $\tilde{\boldsymbol{u}}$. Hence, a speed-up factor of ten can be achieved. The preserved dofs match accurately to the reference solution. The relative error is greatest at the nodes where the boundary conditions are located. This is due to the fact that small displacements lead to high relative error values, although the absolute error is minimal. Of greater importance are the regions with high displacements, for example at the top of the

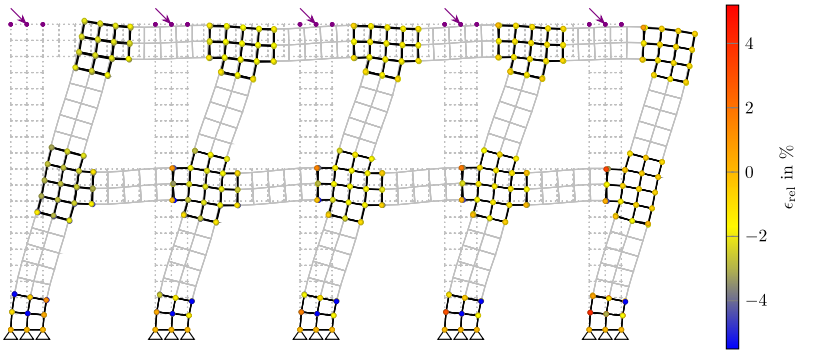

Figure 12. In this three-dimensional example the surrogate element is used 18 times. Purple nodes on the undeformed mesh, depicted with grey dots, indicate forces. The nodes at the bottom are fixed supported. The reference solution is marked with solid grey lines. The surrogate element solution is shown by the black mesh, $f_{\text{sd}} = 100$. Source: Authors' own work

structure. In this region, the absolute relative error does not exceed $\epsilon_{\text{rel}} = 2\%$. When all nodes are considered the error is $\epsilon_{\text{m.a.r.}} = 2.0\%$.

One more aspect, which should be examined, is the influence of adjacent elements. The surrogate element is only applicable when Γ_{IR} and consequently $\boldsymbol{K}_{\text{IR}}$ are identical to those used in the training process. In the following example, the elements at these positions are altered to deviate from the ideal hexahedrons. The circumference of the beam is increased for the adjacent node layers, see Figure 13. The beam is once again fixed supported on both end faces. A force of $[0, 555.6, 555.6]$ N acts on each purple marked node.

Once more, the greatest deviations regarding ϵ_{rel} can be observed in regions with low displacements. Nevertheless, the solution obtained with the surrogate element shows once again high accuracy, as the global error is $\epsilon_{\text{m.a.r.}} = 0.80\%$. Because of that, one can attest the surrogate element a high application flexibility. To summarize, surrogate elements can be employed independently of an arbitrary mesh, provided that the interface regions are identical to those used for training.

3.2 Evaluation of transient simulations

In this section, the surrogate element is employed multiple times within one model in transient simulations. Hence, besides $\boldsymbol{K}_{\text{SE}}$, $\boldsymbol{M}_{\text{SE}}$ is also generated by an ANN. First, a beam structure analogous to that presented in the previous section is trained. For this, the data generation routine from Section 2.6 is utilized. Hence, the surrogate element is trained on a single beam with different load cases and boundaries. The length of the partition, which is modelled by the surrogate element, is now eight elements. Consequently, it is longer compared to the one presented with the static simulations where the surrogate elements length is five conventional elements.

Following training, the surrogate element is applied to a scaled schematic model of a multistorey building, which is excited according the FF earthquakes NS acceleration curves, see Krishnamoorthy and Anita (2016). The full model comprises 26.7×10^3 dofs and 4,068 elements. The surrogate model is built with 99 surrogate elements and 900 conventional elements. Consequently, there are only 8,019 conventional dofs and $99 \cdot 3 = 297$ *adofs*.

The results presented in Figure 14 show high accordance to the full solution.

Similarly, the evaluation of displacements of the marked evaluation nodes from Figure 14 over time is accurate, see Figure 15. In the majority of the timesteps, the relative error is below 2%. The evaluation nodes in the vicinity of the excitation source exhibit the highest error, with an average of approximately 4%. Due to a slight shift in the oscillating nature of the displacements, on some timesteps the relative error is higher. At $t = 0$ s and $t = 0.02$ s, where the displacements are close to zero, the relative error measure provides high discrepancies. However, this is not crucial, as the absolute error is actually quite low.

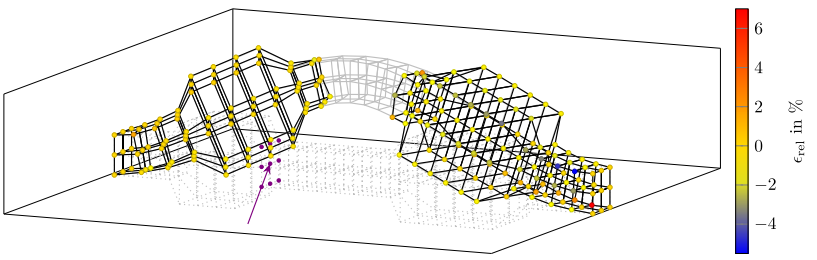

Figure 13. The surrogate element is connected to elements with an untrained shape, but a correct interface region. Nodes with forces are marked purple. The reference solution is marked with solid grey lines, whereas the surrogate element solution is shown by the black mesh, $f_{\text{sd}} = 100$. Source: Authors' own work

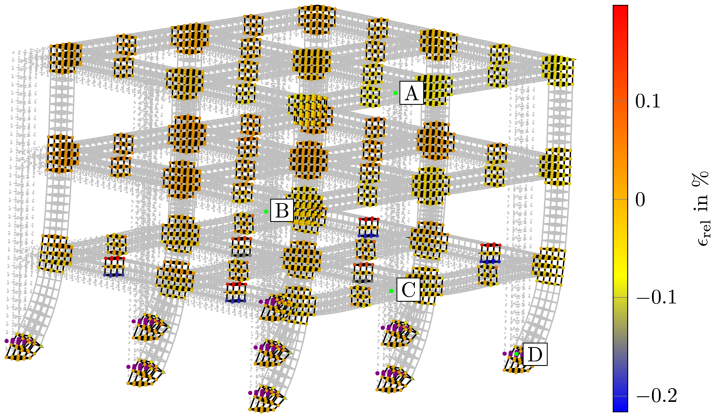

Figure 14. Undeformed and deformed mesh at $t = 0.016$ s. The excitation is applied to the purple colored nodes. The schematic buildings fixed boundary is located at the bottom nodes of the vertical frames. For evaluation of the transient displacements, the evaluation nodes are marked green. Source: Authors' own work

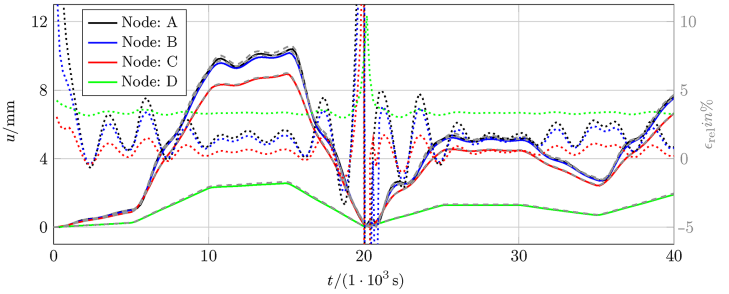

Figure 15. Displacements of the evaluation nodes over time of the excited component. Solid lines depict the surrogate solution, the dashed grey lines next to them are the solution of the full model. The dotted lines describe the relative error on the right vertical axis. Source: Authors' own work

In this example, the damping parameters are set to $\alpha = 1$ and $\beta = 1 \times 10^{-4.}$ These values are not included in the transient training dataset, as there the damping is $\alpha = 2$ and $\beta = 1 \times 10^{-7}$. This shows, that the damping specifications are not dependent on the training. From a computational perspective, solving for the full model requires 11.9×10^3 s, whereas solving the reduced model only takes 370 s, resulting in a speed-up factor of 32.

In a more complex example, a schematic gear is evaluated. But instead of training the surrogate element on the complete gear, it is trained only on a quarter disc. The boundaries and forces are applied according Figure 16. The gear is modelled with hexahedrons, based on 972 dofs in the full approximation and $720 + 3$ dofs in the reduced approximation. The material is identical to the beam examples. Two types of simulations were conducted, one in which forces were applied to each face, and another in which all faces were excited simultaneously. In order to create a balanced dataset, all three directions of the forces are alternated, as explained in Section 2.6.

The model is trained in accordance with the split approach outlined in Section 2.7. Hence, $\boldsymbol{K}_{\mathrm{SE}}$ is based on a static dataset, whereas the transient dataset is the foundation of $\boldsymbol{M}_{\mathrm{SE}}$. Two distinct damping sets are considered, $\alpha = [1,2]$ and $\beta = [10^{-5}, 10^{-6}]$. After evaluation of the dataset it is evident that the proposed method offers a significant advantage, as solving a full simulation for training takes 1.7 s. In contrast, solving the same model with the surrogate elements only takes 0.9 s. The use of the surrogate model resulted in a reduction in the time required by 47% for the example presented in Figure 16.

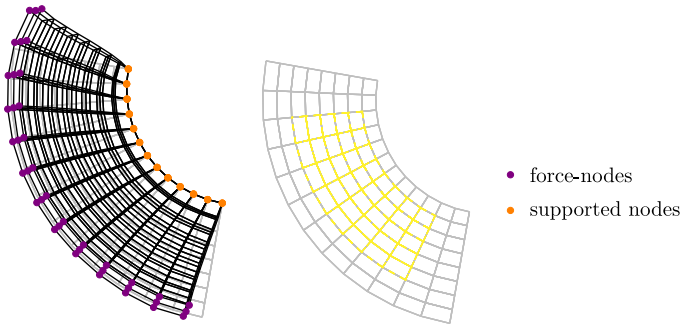

Figure 16. Undeformed mesh (grey) and deformed mesh (black) of the full simulations for training the surrogate element of the gear example with exemplary load case. On the right the elements in yellow show, where the surrogate element will be located in the inference phase. Source: Authors' own work

After training, the surrogate element derived from the quarter disc is applied to a gear. In contrast to the training model, the gear is now a full disc and on the outer boundary of the gear, teeth are present. Consequently, the excitation is now applied to the teeth, which represents a distinct load case compared to the training model. The gear is fixed at all nodes of the inner circle. Furthermore, the damping values, which were not included in the training dataset, are selected as $\alpha = [1]$ and $\beta = [2 \cdot 10^{-6}]$. The full model is discretized by 5,328 dofs, while the reduced model contains only $4{,}320 + 4 \cdot 3$ dofs. Despite the differences between the models during training, the reduced simulation demonstrates high accuracy with the full model. The meshes of the cogwheel at $t = 0.006$ s, are shown in Figure 17.

The maximum absolute relative deviation of 2.3% is observed at the nodes of the excited teeth. This phenomenon can be attributed to the tilting of this teeth because of some erroneous flexibility of the base cylinder. Apart from this, the deviations are below an absolute relative error of 0.5%.

Furthermore, the deformation of the single nodes over time is consistent with the reference, as illustrated in Figure 18. For this, the nodes marked in green in Figure 17 are evaluated.

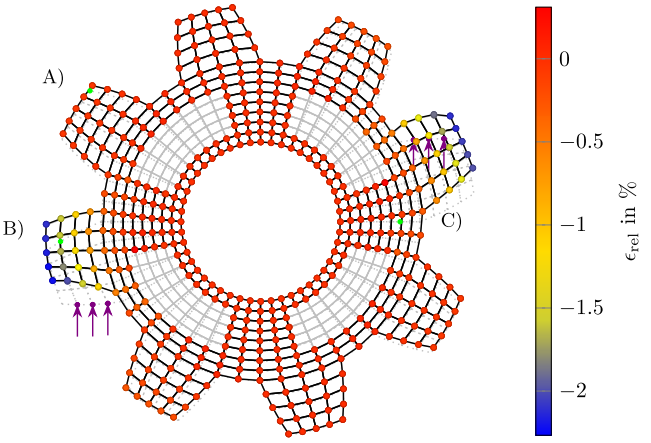

Figure 17. Displacements of a complete three-dimensional cogwheel at $t = 0.006$ s. The reference solution is marked with solid grey lines, the undeformed mesh with dotted grey lines. The surrogate element solution is shown by the black mesh. Purple nodes and arrows describe schematically the applied force. The cogwheel is fixed on the inner cylinder. Source: Authors' own work

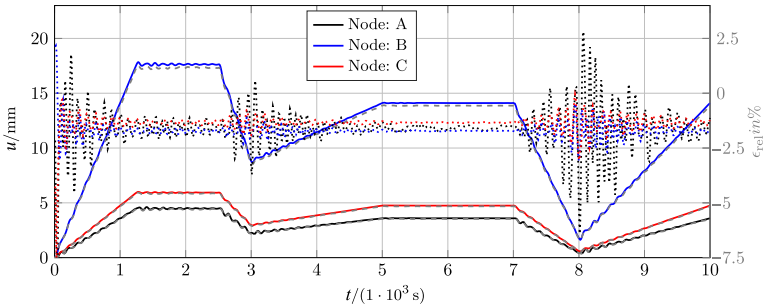

Figure 18. Displacements of the evaluation nodes from Figure 17 over time (solid lines). Dotted lines describe the relative error, dashed lines near the solid lines arise from the full solution. The exciting forces basically follow the oscillating nature of the displacement curves. Source: Authors' own work

A relative mean error of -2% can be identified. Hence, the structure is somewhat over-flexible when utilising surrogate elements. In the event of a significant change of the exciting signal, the error increases in an oscillating manner. Although, the errors are oscillating, the nature of the displacements are met with high accuracy. The discrepancies between the surrogate and reference curves are not transient in nature, rather, they are the result of a minimal deficiency in the stiffness of the surrogate elements. This issue requires further investigation and resolution, potentially through the use of a more suitable dataset and additional training.

The utilisation of surrogate elements enables a 37% reduction of computational time for the full model from 490 s to 310 s of the surrogate model. By increasing the area of the surrogate element, it is possible to achieve a further reduction in computational costs. Once more, all computations were conducted on the same machine with the same in-house FE-code. In a use case, where the inner part of this model remains constant and only the regions near the teeth undergo iterative design changes, the training effort of 36.9×10^3 s for the $\boldsymbol{K}_{\mathrm{SE}}$ and 7.5×10^3 s for $\boldsymbol{M}_{\mathrm{SE}}$ becomes more negligible. Furthermore, when transient simulations are conducted over a longer time span, the amortisation of the training process is accelerated. To illustrate, if the presented cogwheel is simulated for approximately at least $t_{\mathrm{sim}} = 2.5$ s, training and solving of the surrogate model is faster than solving the full model.

4. Conclusion

This paper addresses the training and application of surrogate elements, which are capable of replacing several finite elements in the sense of a load and boundary independent substructuring for static and transient simulations. Furthermore, the presented surrogate elements can be employed for machine learning-based modeling of macroelements in the field of multiscale simulations. The replacement of standard finite elements with surrogate elements results in a reduction in the dimensions of the discretised system of equations. Consequently, the computational costs for solving them is reduced up to 32 times for one of the presented examples. The stiffness matrices of a surrogate element generated by ANNs exhibit identical properties to those of standard finite elements. As a result, they can be seamlessly integrated with standard finite elements. Therefore, a strong focus is on the creation of artificial stiffness matrices, which are psd. Besides the stiffness matrices of the surrogate elements, also mass matrices for transient simulations are trained. Furthermore, the introduction of *artificial degrees of freedom*, which cannot be mapped to a specific geometric representation, is proposed. A data-driven *in silico* approach is employed for the training of the ANN. It is demonstrated that training surrogate elements on models with only a limited number of dofs is sufficient for the desired outcome. Experimental results demonstrate the high accuracy of

computations using surrogate elements on complex structures for static and dynamic simulations, even when trained on simple models. Consequently, any structure can be considered, as long as the interface regions connecting the surrogate elements and the standard finite element regions are identical. The main contribution of this work is the presentation of a method that is highly flexible in its applicability. It fills the gap in surrogate modeling on the system level and on the element integration level, as it is with constitutive laws. The surrogate elements can be interpreted as substructures that are not dependent on boundary conditions and load cases in the offline generation process.

In future work, our aim is to incorporate the ability to model different material properties of linear materials. Also, it should be evaluated how the proposed approach scales for even larger problems, for example in simulations with very fine meshes where the surrogate elements replace much bigger regions. The ANNs input could consist of two vectors, one representing displacements and the other capturing material properties, such as the elasticity modulus corresponding to the chosen material model. Regarding accuracy of the method, it could be further enhanced by implementing more precise training procedures and datasets. Furthermore, the influence of *artificial degrees of freedom* on the accuracy of the surrogate modeling should be assessed in detail. It is also our intention to apply surrogate elements to geometrically nonlinear problems to fully exploit the potential of the presented method with non-constant input vectors. Furthermore, this approach could be extended to enable the replacement of specific regions of FE models that deal with contact.

References

Allen, M.S., Rixen, D., van der Seijs, M., Tiso, P., Abrahamsson, T. and Mayes, R.L. (2020), *Model Reduction Concepts and Substructuring Approaches for Linear Systems*, Springer, Cham, pp. 25-73, doi: 10.1007/978-3-030-25532-9_3.

Badia, S., Li, W. and Martín, A.F. (2024), "Finite element interpolated neural networks for solving forward and inverse problems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 418, 116505, doi: 10.1016/j.cma.2023.116505.

Bathe, K.-J. (2007), *Finite Element Procedures*, Prentice-Hall of India, New Delhi.

Beck, J.V. and Woodbury, K.A. (1998), "Inverse problems and parameter estimation: integration of measurements and analysis", *Measurement Science and Technology*, Vol. 9 No. 6, pp. 839-847, doi: 10.1088/0957-0233/9/6/001.

Berg, J. and Nyström, K. (2021), "Neural networks as smooth priors for inverse problems for PDEs", *Journal of Computational Mathematics and Data Science*, Vol. 1, 100008, doi: 10.1016/j.jcmds.2021.100008.

Bishara, D., Xie, Y., Liu, W.K. and Li, S. (2023), "A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials", *Archives of Computational Methods in Engineering*, Vol. 30 No. 1, pp. 191-222, doi: 10.1007/s11831-022-09795-8.

Cammarata, A., Sinatra, R. and Maddìo, P.D. (2019), "Static condensation method for the reduced dynamic modeling of mechanisms and structures", *Archive of Applied Mechanics*, Vol. 89 No. 10, pp. 2033-2051, doi: 10.1007/s00419-019-01560-x.

Capuano, G. and Rimoli, J. (2018), "Smart finite elements: a novel machine learning application", *Computer Methods in Applied Mechanics and Engineering*, Vol. 345, pp. 363-381, doi: 10.1016/j.cma.2018.10.046.

Chamekh, A., Bel Hadj Salah, H. and Hambli, R. (2009), "Inverse technique identification of material parameters using finite element and neural network computation", *The International Journal of Advanced Manufacturing Technology*, Vol. 44 No. 1, pp. 173-179, doi: 10.1007/s00170-008-1809-6.

Chen, J., Schäfer, F., Huang, J. and Desbrun, M. (2021), "Multiscale Cholesky preconditioning for ill-conditioned problems", *ACM Transactions on Graphics*, Vol. 40 No. 4, pp. 1-13, doi: 10.1145/3450626.3459851.

Chung, E., Efendiev, Y. and Hou, T.Y. (2023), *Multiscale Model Reduction*, 1st ed., Springer, Cham, doi: 10.1007/978-3-031-20409-8_1.

Craig, R.R. and Bampton, M.C.C. (1968), "Coupling of substructures for dynamic analyses", *AIAA Journal*, Vol. 6 No. 7, pp. 1313-1319, doi: 10.25143.4741.

Czech, C., Lesjak, M., Bach, C. and Duddeck, F. (2022), "Data-driven models for crashworthiness optimisation: intrusive and non-intrusive model order reduction techniques", *Structural and Multidisciplinary Optimization*, Vol. 65 No. 7, 190, doi: 10.1007/s00158-022-03282-1.

Deng, S., Hosseinmardi, S., Wang, L., Apelian, D. and Bostanabad, R. (2024), "Data-driven physics-constrained recurrent neural networks for multiscale damage modeling of metallic alloys with process-induced porosity", *Computational Mechanics*, Vol. 74 No. 1, pp. 191-221, doi: 10.1007/s00466-023-02429-1.

Fachinotti, V.D., Albanesi, A.E. and Flores, F.G. (2020), "Inverse finite element analysis using a simple reduced integration hexahedral solid-shell element", *Finite Elements in Analysis and Design*, Vol. 178, 103440, doi: 10.1016/j.finel.2020.103440.

Franke, M. and Wagner, M. (2024), "Transient surrogate modeling of modally reduced structures with discontinuous loads and damping", *Archive of Applied Mechanics*, Vol. 94 No. 6, pp. 1647-1666, doi: 10.1007/s00419-024-02601-w.

Fresca, S., Gobat, G., Fedeli, P., Frangi, A. and Manzoni, A. (2022), "Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures", *International Journal for Numerical Methods in Engineering*, Vol. 123 No. 20, pp. 4749-4777, doi: 10.1002/nme.7054.

Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, MA, doi: 10.1007/s10710-017-9314-z.

Guyan, R.J. (1965), "Reduction of stiffness and mass matrices", *AIAA Journal*, Vol. 3 No. 2, p. 380, doi: 10.2514/3.2874.

Higham, N.J. (1988), "Computing a nearest symmetric positive semidefinite matrix", *Linear Algebra and Its Applications*, Vol. 103, pp. 103-118, doi: 10.1016/0024-3795(88)90223-6.

Huang, M., Cui, T., Liu, C., Du, Z., Zhang, J., He, C. and Guo, X. (2023), "A problem-independent machine learning (PIML) enhanced substructure-based approach for large-scale structural analysis and topology optimization of linear elastic structures", *Extreme Mechanics Letters*, Vol. 63, 102041, doi: 10.1016/j.eml.2023.102041.

Hughes, T.J.R. (1987), *The Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ.

Ingrid, L. and Rottner, T. (1999), "Krylov subspace methods for structural finite element analysis", *Parallel Computing*, Vol. 25 No. 7, pp. 861-875, doi: 10.1016/S0167-8191(99)00022-8.

Jung, J., Yoon, K. and Lee, P.S. (2020), "Deep learned finite elements", *Computer Methods in Applied Mechanics and Engineering*, Vol. 372, 113401, doi: 10.1016/j.cma.2020.11340.

Karapiperis, K., Stainier, L., Ortiz, M. and Andrade, J. (2021), "Data-driven multiscale modeling in mechanics", *Journal of the Mechanics and Physics of Solids*, Vol. 147, 104239, doi: 10.1016/j.jmps.2020.104239.

Kingma, D. and Ba, J. (2014), "Adam: a method for stochastic optimization", *International Conference on Learning Representations*.

Kirchdoerfer, T. and Ortiz, M. (2016), "Data-driven computational mechanics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 304, pp. 81-101, doi: 10.1016/j.cma.2016.02.001.

Kneifl, J., Fehr, J., Brunton, S.L. and Kutz, J.N. (2024), "Multi-hierarchical surrogate learning for explicit structural dynamical systems using graph convolutional neural networks", *Computational Mechanics*, Vol. 75 No. 3, pp. 1115-1135, doi: 10.1007/s00466-024-02553-6.

Koeppe, A., Bamer, F. and Markert, B. (2020), "An intelligent nonlinear meta element for elastoplastic continua: deep learning using a new Time-distributed residual U-Net architecture", *Computer Methods in Applied Mechanics and Engineering*, Vol. 366, 113088, doi: 10.1016/j.cma.2020.113088.

Krishnamoorthy, A. and Anita, S. (2016), "Soil–structure interaction analysis of a FPS-isolated structure using finite element model", *Structures*, Vol. 5 No. 5, pp. 44-57, doi: 10.1016/j.istruc.2015.08.003.

Kudela, J. and Matousek, R. (2022), "Recent advances and applications of surrogate models for finite element method computations: a review", *Soft Computing*, Vol. 26 No. 24, pp. 13709-13733, doi: 10.1007/s00500-022-07362-8.

Le Guennec, Y., Brunet, J.P., Daim, F.Z., Chau, M. and Tourbier, Y. (2018), "A parametric and non-intrusive reduced order model of car crash simulation", *Computer Methods in Applied Mechanics and Engineering*, Vol. 338, pp. 186-207, doi: 10.1016/j.cma.2018.03.005.

Lewis, A.S. (2003), "The mathematics of eigenvalue optimization", *Mathematical Programming*, Vol. 97 No. 1, pp. 155-176, doi: 10.1007/s10107-003-0441-3.

Mahdiabadi, M.K., Tiso, P., Brandt, A. and Rixen, D.J. (2021), "A non-intrusive model-order reduction of geometrically nonlinear structural dynamics using modal derivatives", *Mechanical Systems and Signal Processing*, Vol. 147, 107126, doi: 10.1016/j.ymssp.2020.107126.

Meethal, R.E., Kodakkal, A., Khalil, M., Ghantasala, A., Obst, B., Bletzinger, K.U. and Wüchner, R. (2023), "Finite element method-enhanced neural network for forward and inverse problems", *AMSES*, Vol. 10 No. 1, 6, doi: 10.1186/s40323-023-00243-1.

Nguyen, L.T.K., Rambausek, M. and Keip, M.A. (2020), "Variational framework for distance-minimizing method in data-driven computational mechanics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 365, 112898, doi: 10.1016/j.cma.2020.112898.

Nguyen, P.C.H., Choi, J.B., Udaykumar, H.S. and Baek, S. (2023), "Challenges and opportunities for machine learning in multiscale computational modeling", *Journal of Computing and Information Science in Engineering*, Vol. 23 No. 6, doi: 10.1115/1.4062495.

Oishi, A. and Yagawa, G. (2017), "Computational mechanics enhanced by deep learning", *Computer Methods in Applied Mechanics and Engineering*, Vol. 327, pp. 327-351, doi: 10.1016/j.cma.2017.08.040.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T. and Lin, Z. (2019), "Pytorch: an imperative style, high-performance deep learning library", *33rd Annual Conference on Neural Information Processing Systems*, Vancouver, available at: https://arxiv.org/abs/1912.01703

Pourazarm, B., Vahdani, S. and Farjoodi, J. (2011), "Reduced stiffness method for nonlinear analysis of structural frames", *Scientia Iranica*, Vol. 18 No. 2, pp. 181-189, doi: 10.1016/j.scient.2011.03.027.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019), "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational Physics*, Vol. 378, pp. 686-707, doi: 10.1016/j.jcp.2018.10.045.

Römer, U., Hartmann, S., Tröger, J.A., Anton, D., Wessels, H., Flaschel, M. and De Lorenzis, L. (2024), "Reduced and all-at-once approaches for model calibration and discovery in computational solid mechanics", *Applied Mechanics Reviews*, Vol. 77 No. 4, doi: 10.1115/1.4066118.

Sato, A.K., Martins, T.C. and Tsuzuki, M.S.G. (2023), "GPU implementation of an incomplete Cholesky conjugate gradient solver for a FEM-generated system using full kernel consolidation", *Soft Computing*, Vol. 27 No. 14, pp. 9307-9320, doi: 10.1007/s00500-023-08125-9.

Schilders, W.A.D., Vorst, H.A. and Rommes, J. (2008), *Model Order Reduction: Theory, Research Aspects and Applications*, 1st ed., Springer, Berlin; Heidelberg.

Wagner, M. (2022), *Lineare Und Nichtlineare FEM: Eine Einführung Mit Anwendungen in Der Umformsimulation Mit LS-DYNA*, 3rd ed., Springer, Wiesbaden.

Wilson, E.L. (1976), "The static condensation algorithm", *International Journal for Numerical Methods in Engineering*, Vol. 8 No. 1, pp. 198-203, doi: 10.1002/nme.1620080115.

Wriggers, P. (2008), *Nonlinear Finite Element Methods*, Springer, Berlin.

Zhang, D., Lu, L., Guo, L. and Karniadakis, G.E. (2019), "Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems", *Journal of Computational Physics*, Vol. 397, 108850, doi: 10.1016/j.jcp.2019.07.048.

Zienkiewicz, O.C., De Gago, J. and Kelly, D. (1983), "The hierarchical concept in finite element analysis", *Computers and Structures*, Vol. 16 No. 1, pp. 53-65, doi: 10.1016/0045-7949(83)90147-5.

Corresponding author
Markus Franke can be contacted at: markus.franke@oth-regensburg.de