



# The difference-based equivalent static load method: an improvement of the ESL method's nonlinear approximation quality

J. Triller<sup>1,2</sup> · R. Immel<sup>1</sup> · A. Timmer<sup>1,2</sup> · L. Harzheim<sup>1,2</sup>

Received: 17 September 2019 / Revised: 10 December 2020 / Accepted: 18 December 2020 / Published online: 2 April 2021  
© The Author(s) 2021

## Abstract

Nonlinear dynamic structural optimization is a real challenge, in particular for problems that require the use of explicit solvers, e.g., crash. Here, the number of design variables is typically very limited. A way to overcome this drawback is to use linear auxiliary load cases which are derived from nonlinear dynamic analysis results in order to enable the application of linear static response optimization. The equivalent static load method (ESLM) provides a well-defined procedure to create such linear auxiliary load cases. The main idea here is that after the selection of a number of representative time steps, a set of equivalent static loads (ESLs) is computed for each time step such that the resulting displacement field in the linear static analysis is identical to the respective field in the nonlinear dynamic analysis. Each set of ESLs defines an auxiliary load case, which is used in the linear static response optimization. The crucial point is that the finite element (FE)-model for each auxiliary load case describes the undeformed initial geometry. This can lead to insufficient approximation quality in the linear static system for highly nonlinear problems. To overcome this drawback, a difference-based extension of the ESL method called DiESL has been developed for nonlinear dynamic response optimization problems. Here, the FE-model for each auxiliary load case describes the deformed nonlinear geometry at the respective time, and the corresponding ESLs create only the displacement field leading to the deformed state of the subsequent ESL time step. Consequently, responses in each linear auxiliary load case (corresponding to a time step) are computed as the accumulated sum of the previous linear auxiliary load cases. Furthermore, the linear static response optimization problem consists not only of one but of  $n_T$  FE-models where  $n_T$  is the number of selected time steps. Such a multi-model optimization (MMO) can be solved with commercial FE solvers. It turns out that the DiESL approach leads to a significant improvement of the nonlinear approximation quality and faster convergence to the optimum when compared to standard ESLM. This will be demonstrated and discussed based on selected test examples.

**Keywords** DiESL · Equivalent static load method · Multi-model optimization · Crashworthiness · Explicit solvers

## 1 Introduction

Linear static response structural optimization is highly efficient and is therefore embedded in a considerable number of applications commonly used during the design process in industry. Many commercial codes are available in this area, enabling sizing, shape, and topology

optimization in an acceptable amount of time. The time saving can especially be attributed to the availability of (semi-) analytical sensitivity analysis enabling the application of very efficient gradient-based optimizers. In contrast to the well-established optimization based on linear analysis, the real challenge in optimization is the optimization of highly nonlinear dynamic systems. A prime example is the optimization of crash-related problems in automotive industry which is also the main objective of this paper. Furthermore, we restrict ourselves to the use of commercial crash solvers for the nonlinear dynamic analysis to ensure that the proposed DiESL (difference-based equivalent static load) method can be applied to real automotive crash problems. Worldwide, only explicit solvers are used in automotive crash analysis. Here, the most dominating part of the nonlinearities does not result

---

Responsible Editor: Seonho Cho

✉ J. Triller  
jens.triller@stud.tu-darmstadt.de

<sup>1</sup> Opel Automobile GmbH, Engineering Center Ruesselsheim, O/V Vehicle CAE, PC S1-01, D-65423 Ruesselsheim, Germany

<sup>2</sup> Technische Universität Darmstadt, Darmstadt, Germany

from material or geometric nonlinearities but from contact forces. They may not only change in magnitude but especially in location from time step to time step. As a consequence, it is very difficult to apply sensitivity analysis which may be the reason that no commercial code in this area is able to compute sensitivities enabling gradient-based optimization for crash. Instead, metamodel-based methods are often used. Examples for such metamodells are polynomials, radial basis functions (Powell 1992; Schaback and Wendland 2001; Schaback 2002), neural networks (Hornik et al. 1990, 1992; Waszczyszyn 1999), and kriging models (Matheron 1963; Cressie 1988, 1989, 1990). The drawback of the metamodel-based optimization is that the number of design variables is limited and applications with extremely high number of design variables such as topology optimization are impossible due to the large number of required designs that need to be evaluated in order to fit the metamodells.

A promising approach to circumvent the missing-sensitivity-problem of commercial crash solvers is to define linear auxiliary load cases enabling linear static response optimization. This means that the nonlinear dynamic optimization problem is solved by solving a sequence of linear static response optimization subproblems. The equivalent static load method (ESLM) provides a procedure to compute such auxiliary load cases. These auxiliary load cases are created by applying equivalent static loads (ESLs) to linear statics.

The ESLM has been successfully applied to various kinds of optimization like sizing, shape, free sizing, and topology optimization (Choi and Park 2002; Park et al. 2005; Lee et al. 2013b; Shin et al. 2007; Lee et al. 2007, 2013a; Jeong et al. 2008; Jang et al. 2012; Hong et al. 2010; Kim and Park 2010; Park 2011; Lee and Park 2015; Karev et al. 2018; Choi et al. 2018; Karev et al. 2019). Nevertheless, the ESLM has some limitations and disadvantages which are discussed in this paper. The main issues result from the fact that the ESLs are always calculated based on the undeformed initial geometry. The idea presented in this paper tries to overcome these issues by using a difference-based approach for calculating the ESLs. The focus is on solving nonlinear dynamic response optimization problems which require usage of explicit solvers.

This paper is structured in the following way: In Section 2, the ESLM approach is presented. In Section 3, a difference-based extension for calculating the ESLs called the DiESL method is introduced in detail. In that process, a general approach for handling structural responses is given. Furthermore, the implementation of the DiESL method is explained. Afterwards, the DiESL approach is tested and compared to the ESLM for two different examples in Section 4. Finally, a conclusion and an outlook are worked out.

## 2 The ESLM (equivalent static load method)

The ESLM has been introduced by Choi and Park (Choi and Park 2002). It provides a well-defined procedure to create linear auxiliary load cases for nonlinear dynamic response optimization problems which are used in optimization based on linear static analysis. The objective function and the constraints used in the linear static response optimization and in the nonlinear dynamic response optimization are identical. The only difference is that the nonlinear dynamic responses are approximated by linear static responses. For dynamic responses such as section forces, velocities, and accelerations, which are not defined in linear statics, it is necessary to find proper approximations. For velocity and acceleration, for instance, simple finite forward (Jeong et al. 2010) or central (Karev et al. 2018; Karev et al. 2019) differences between adjacent load cases (corresponding to adjacent time steps) can be used. The advantage of ESLM is that well-developed commercial software systems such as MSC NASTRAN, Altair OptiStruct, or VRAND GENESIS can be used for analysis and optimization and no development of own sensitivity analysis and optimization algorithm is necessary. In this sense, ESLM is not an optimization algorithm but a method to transform an optimization problem involving nonlinear and/or dynamic responses into one using responses from a set of linear static subcases.

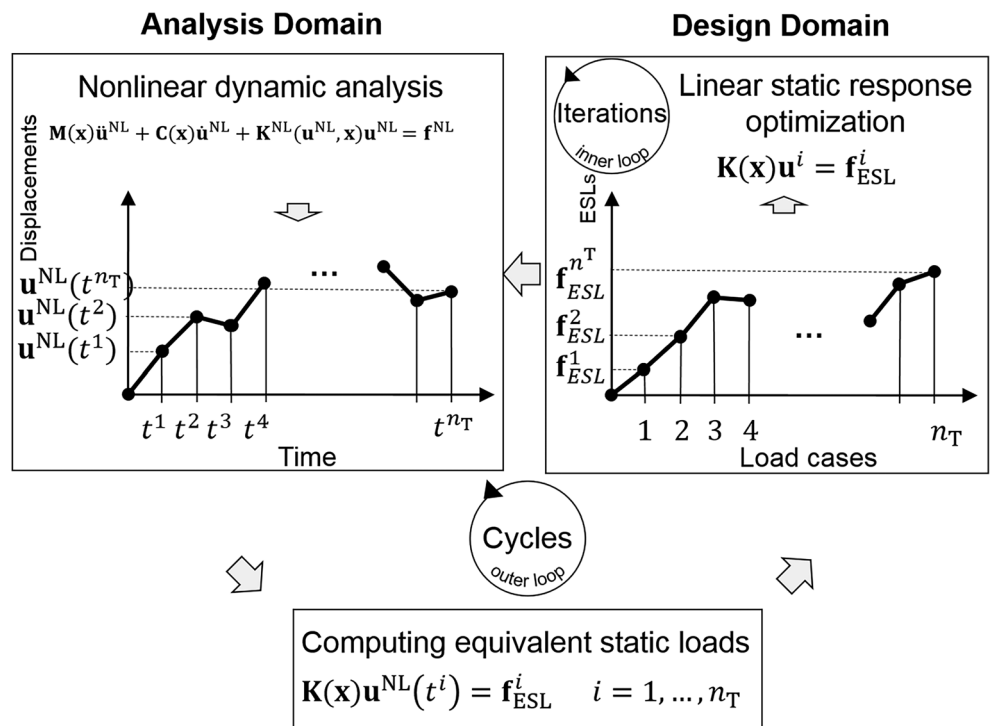
The procedure for computing the ESLs of the auxiliary load cases is now the following: In the first step, the user defines a set of  $n_T$  time steps  $t^i$ ,  $i = 1, \dots, n_T$ , to capture the behavior of the nonlinear problem with sufficient accuracy. The basic idea is now to calculate loads  $\mathbf{f}_{\text{ESL}}^i$  which yield the same displacement vector  $\mathbf{u}^i$  in linear statics as those obtained for the selected time steps  $t^i$  in the nonlinear dynamic analysis  $\mathbf{u}^{\text{NL}}(t^i)$  (Park 2011). Thus, the loads received are equivalent in the sense that they lead to same displacement fields.

In Fig. 1, the process of the ESLM is illustrated for a nonlinear dynamic problem. The process starts in the analysis domain with a nonlinear dynamic analysis from which the displacement vectors  $\mathbf{u}^{\text{NL}}(t^i)$  for  $n_T$  user-selected time steps  $t^i$  are obtained. The equivalent static loads  $\mathbf{f}_{\text{ESL}}^i$  are then calculated by multiplying the stiffness matrix  $\mathbf{K}(\mathbf{x})$  by the displacements fields

$$\mathbf{K}(\mathbf{x})\mathbf{u}^{\text{NL}}(t^i) = \mathbf{f}_{\text{ESL}}^i; \quad i = 1, \dots, n_T. \quad (1)$$

This leads to a set of ESLs and a corresponding linear auxiliary load case for each time step  $t^i$ . Based on these  $n_T$  load cases, linear static response optimization is performed in the design domain afterwards. All load cases are based on the same FE model (i.e., with the same initial geometry) and thus have the same stiffness matrix  $\mathbf{K}(\mathbf{x})$ . Hence, the optimization can be performed with a single optimization deck containing

**Fig. 1** Optimization process of ESLM and DiESL methods for a nonlinear dynamic problem (note that equations and symbols refer to ESLM and are different for DiESL)



$n_T$  subcases. After the optimization step (inner loop), nonlinear analysis has to be applied again with the updated design variables. In the following, the outer loop iterations are called *cycles* to distinguish them from the *iterations* of the inner loop in the linear static response subproblem.

Note that the displacement fields of the nonlinear and the linear analysis are identical only at the beginning of the optimization in the inner loop. At the end of the inner loop, the linear and the nonlinear dynamic responses no longer match perfectly because the linear auxiliary load cases are only an approximation of the nonlinear behavior of the system. If the difference is too high, the process is iterated (outer loop) until the difference is small enough and additional termination criteria (Shin et al. 2007; Jeong et al. 2010; Kim and Park 2010; Park 2011) are fulfilled. Furthermore, it can be expected that the difference between the linear and nonlinear dynamic responses increases with the length of the inner loop optimization path. If this difference is too high, the update in the outer loop may result in huge changes in the search directions and convergence is slowed down or even not guaranteed anymore. For this reason, commercial ESLM codes, such as VRAND ESLDYNA, limit the number of iterations and offer the application of move limits to restrict the length of the inner loop optimization path in each cycle.

ESLM has been successfully applied to different types of optimization and analyses (Park 2011) such as

- linear dynamic response optimization

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

- structural optimization for multi-body dynamic systems
- structural optimization for flexible multi-body dynamic systems
- nonlinear static response optimization

$$K^{NL}(x, u^{NL})u^{NL} = f \tag{3}$$

- nonlinear dynamic response optimization

$$M(x)\ddot{u}^{NL} + C(x)\dot{u}^{NL} + K^{NL}(x, u)u^{NL} = f(t) \tag{4}$$

The most challenging application is the optimization of crash problems in the field of nonlinear dynamic response optimization, which is the focus of this publication. The original crash optimization problem can be stated as

$$\min f(x, u^{NL}(x, t)) \tag{5a}$$

$$\text{s.t. } g_j(x, u^{NL}(x, t)) \leq 0; j = 1, \dots, m \tag{5b}$$

$$x^L \leq x \leq x^U; x \in \mathbb{R}^n \tag{5c}$$

Here,  $f(x)$  is the objective function,  $m$  the number of constraints  $g_j$ , and  $x^L$  and  $x^U$  are the lower and upper bounds of the design variable  $x$ , respectively. The displacement vector  $u^{NL}$  is the solution of

$$M(x)\ddot{u}^{NL}(t) + C(x)\dot{u}^{NL}(t) + K^{NL}(x, u^{NL}(t))u^{NL}(t) = f \tag{5d}$$

For crash problems, the external forces  $\mathbf{f}$  are typically zero (except for gravity) but the internal forces  $\mathbf{f}^{\text{int}} = -\mathbf{K}^{\text{NL}}\mathbf{u}^{\text{NL}}$  may change in magnitude and location very fast from time step to time step due to contact. Commercial codes are available for crash analysis such as LSTC LS-DYNA, ESI Pam-Crash, or Altair RADIOSS. They use an explicit time integration scheme and a penalty formulation for contacts. This means that the contacts are modeled by springs and a very time-consuming task in a crash code is the computation of contact areas and resulting contact forces. It results in an adding and removing of contact spring elements for closing or opening contacts, respectively. Of course, such codes can handle complicated nonlinear material models and geometric nonlinearities as well. Nevertheless, the most important or dominating nonlinearities result from the contacts.

The ESLM solves the following optimization problem:

$$\min f(\mathbf{x}, \mathbf{u}(\mathbf{x})) \quad (6a)$$

$$\text{s.t. } g_j(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq 0; j = 1, \dots, m \quad (6b)$$

$$\mathbf{x}^{\text{L}} \leq \mathbf{x} \leq \mathbf{x}^{\text{U}}; \mathbf{x} \in \mathbb{R}^n \quad (6c)$$

where  $\mathbf{u}$  is the solution of

$$\mathbf{K}(\mathbf{x})\mathbf{u}^i = \mathbf{f}_{\text{ESL}}^i; i = 1, \dots, n_T, \quad (6d)$$

the equivalent static loads  $\mathbf{f}_{\text{ESL}}^i$  for load case  $i$  are computed as

$$\mathbf{f}_{\text{ESL}}^i = \mathbf{K}(\mathbf{x})\mathbf{u}^{\text{NL}}(t^i); i = 1, \dots, n_T \quad (6e)$$

and  $\mathbf{u}^{\text{NL}}(t^i)$  is the solution of

$$\begin{aligned} \mathbf{M}(\mathbf{x})\ddot{\mathbf{u}}^{\text{NL}}(t) + \mathbf{C}(\mathbf{x})\dot{\mathbf{u}}^{\text{NL}}(t) + \mathbf{K}^{\text{NL}}(\mathbf{x}, \mathbf{u}^{\text{NL}}(t))\mathbf{u}^{\text{NL}}(t) \\ = \mathbf{f}(t) \end{aligned} \quad (6f)$$

for the selected time steps  $t^i$ .

It was stated in Park and Kang 2003 that the Karush-Kuhn-Tucker (KKT) conditions of the linear static subproblem and the original dynamic problem are identical if the ESLM satisfies specific termination criteria. Stolpe 2014 showed that the proof is incomplete and not correct. According to Stolpe, the most critical point is that the displacement sensitivities of the original problem and the subproblem are not necessarily identical. Therefore, Stolpe supposes an alternative algorithm where these sensitivities are identical by definition. However, the supposed algorithm requires the computation of sensitivities of the original dynamic problem and is therefore not applicable to optimization of crash problems. Furthermore, Stolpe et al. 2018 presented an example illustrating that ESLM may fail to find the optimal design of the dynamic response optimization problem. Nevertheless, the topic is still under discussion because Park and Lee claimed in their latest publication (Park and Lee 2019) that Stolpe's correction had now been taken into account and that the ESLM solution is a

KKT point if additional mathematical aspects are added. This discussion history demonstrates how difficult it is sometimes to prove convergence and optimality criteria for methods based on an engineering approach such as ESLM. The same holds for the DiESL approach presented in the next chapter and future will show if it is possible to prove if this method really guarantees to find the optimum of the original dynamic problem.

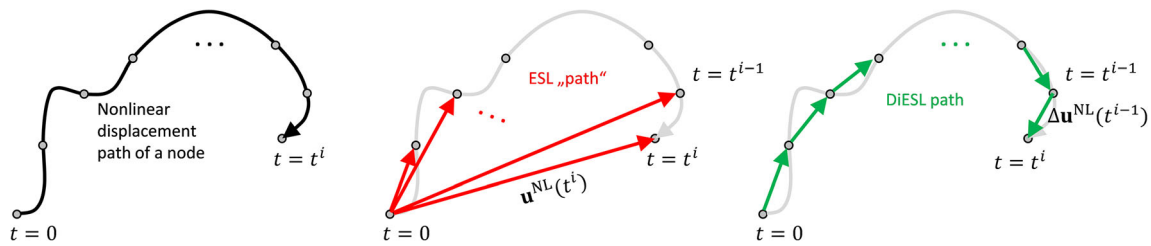
### 3 The DiESL (difference-based equivalent static loads) method

As mentioned above, ESLM is not really an optimization algorithm but a method to compute linear static auxiliary load cases to approximate nonlinear dynamic responses. The intention of DiESL is to improve this approximation quality as well as the resulting approximation quality of the sensitivities. It can be expected that the better the approximation quality, the higher the convergence speed of the optimization. It is intended to improve the ESLM by improving the approximation of the sensitivities.

To explain the idea of DiESL, we assume that a nonlinear dynamic FE analysis has been carried out. We now study the nonlinear spatial path of an arbitrary FEM node where its location at time  $t=0$  corresponds to the initial undeformed structure and the coordinate at  $t=t^i$  to the deformed structure at this time step (see Fig. 2, left). The main drawback of ESLM is that it always starts from the initial undeformed structure. This means that we always obtain a linear "path" from the location at  $t=0$  to the location at  $t=t^i$  if we want to compute the ESLs for displacement  $\mathbf{u}^{\text{NL}}(t^i)$  in linear static path (Fig. 2, middle). This "path" is in general completely different from the nonlinear path, causing the following issues:

- The ESLs in the linear static subproblem are completely different to the nodal forces in the original nonlinear dynamic problem. Typically, the ESLs are significantly higher.
- The strains in the linear static subproblem and the nonlinear dynamic problem are completely different.
- The stresses in the linear static subproblem and the nonlinear dynamic problem are completely different.
- The sensitivities in the linear static subproblem and the nonlinear dynamic problem may be completely different and may not even match in sign.

The idea of the difference-based equivalent static loads (DiESL) is now to improve the approximation quality of the subproblem by following the nonlinear displacement path. Here, the displacements of each node  $j$  at time step  $t^i$  used for computing the ESLs are not referring to the initial



**Fig. 2** Displacement path of an arbitrary node during the deformation of a structure (left) and the corresponding displacement  $\mathbf{u}^{NL}(t^i)$  (middle) and

$\Delta \mathbf{u}^{NL}(t^i)$  (right) used for the computation of the ESLs and DiESLs at time step  $t^i$ , respectively

coordinates  $\mathbf{r}_j(t=0)$ , but to the coordinates of the deformed geometry  $\mathbf{r}_j(t^{i-1})$  at the previous time step  $t^{i-1}$  (Fig. 2, right). As a consequence, the DiESL approach requires  $n_T$  FE models, one for each time step  $t^i, i=0, \dots, n_T-1$  (note:  $t^0$  corresponds to the first FE model with undeformed geometry at  $t=0$ ). In the following, we call each of these FE models at time step  $t^i$  the “ $i^{\text{th}}$  linear submodel” (LSM $^i$ ). An additional consequence is that the approximation of a nonlinear displacement at time  $t^i$  is no longer given by the displacement of load case  $i$  but as a sum of incremental displacements of different LSM $^i$ s. Each LSM $^i$  is created by modifying the coordinates of each node without changing the topology of the FE mesh.

The node coordinates of an LSM $^i$  are collected in the vector

$$\mathbf{r}^T(t^i) = (\mathbf{r}_1^T(t^i), \mathbf{r}_2^T(t^i), \dots, \mathbf{r}_{n_N}^T(t^i)) \tag{7}$$

which contains the coordinates  $\mathbf{r}_j(t^i)$  of all  $n_N$  nodes. Accordingly,  $\mathbf{r}(t^0)$  describes the coordinates of all nodes of the undeformed model. Furthermore, the vector

$$\mathbf{u}^{NL}(t)^T = (\mathbf{u}_1^{NL}(t)^T, \mathbf{u}_2^{NL}(t)^T, \dots, \mathbf{u}_{n_N}^{NL}(t)^T) \tag{8}$$

contains the nonlinear displacements of all  $n_N$  nodes at the time  $t$ . The coordinates of a linear submodel LSM $^i$  describing the deformed geometry at time  $t^i$  can therefore be calculated by

$$\mathbf{r}(t^i) = \mathbf{r}(t^0) + \mathbf{u}^{NL}(t^i). \tag{9}$$

Note that we used the ESLM displacement “path” to compute the coordinates of the deformed geometry in (9). Generally, this can also be done by summing up the incremental displacements  $\Delta \mathbf{u}^{NL}(t^i)$  (see Fig. 2, right) but this is more complicated and leads to the same result.

To determine the incremental equivalent static loads  $\Delta \mathbf{f}_{\text{DiESL}}^i$  in the linear submodel LSM $^i$ , the corresponding stiffness matrix  $\mathbf{K}^i = \mathbf{K}(\mathbf{x}, \mathbf{r}(t^i))$ , which depends on the design variables  $\mathbf{x}$  and the nodal coordinates  $\mathbf{r}(t^i)$  of LSM $^i$ , has to be multiplied by the vector of the incremental nonlinear displacements  $\Delta \mathbf{u}^{NL}(t^i)$

$$\mathbf{K}^i \Delta \mathbf{u}^{NL}(t^i) = \Delta \mathbf{f}_{\text{DiESL}}^i. \tag{10}$$

The incremental nonlinear displacements leading from  $\mathbf{r}(t^i)$  to  $\mathbf{r}(t^{i+1})$  are calculated by

$$\Delta \mathbf{u}^{NL}(t^i) = \mathbf{u}^{NL}(t^{i+1}) - \mathbf{u}^{NL}(t^i) \tag{11}$$

which is illustrated in Fig. 3 for an arbitrary node. Using the incremental equivalent static loads  $\Delta \mathbf{f}_{\text{DiESL}}^i$ , gradient-based linear static response optimization can now be performed similar as described before for the ESLM as illustrated in Fig. 1. But in contrast to the ESLM, which requires only a single FE model representing the undeformed structure of the underlying initial model, now  $n_T$  FE models have to be considered in one optimization run. Consequently, multi-model optimization (MMO) has to be applied where more than one FE model can be taken into account simultaneously in one linear static response optimization run. Here, the FE equation of linear statics

$$\mathbf{K}^i \Delta \mathbf{u}^i = \Delta \mathbf{f}_{\text{DiESL}}^i \tag{12}$$

is solved for each LSM $^i$ , which yields the incremental linear displacements  $\Delta \mathbf{u}^{iT} = (\Delta \mathbf{u}_1^{iT}, \Delta \mathbf{u}_2^{iT}, \dots, \Delta \mathbf{u}_{n_N}^{iT})$ .

Figure 4 illustrates the benefits of the DiESL approach based on the example of a three-point bending load case. ESLM can only capture the bending behavior of the beam but not the tensile stress that occurs in the deformed states. The reason is that ESLM always starts from the undeformed structure; hence, the stiffness matrix  $\mathbf{K}$  is assembled from only bending contributions for all beam elements. There are no tensile stiffness contributions which appear only after deformation. In contrast, DiESL captures the tensile stiffness contributions because it is based on deformed structures as illustrated on the right side of Fig. 4. Here, the dotted and the solid lines show the deformed shape at  $t=t^{i-1}$  and  $t=t^i$ , respectively. The displacements  $\Delta \mathbf{u}^{NL}(t^{i-1})$  describe the gap between the dotted and the solid line, and it is obvious that the smaller  $\Delta \mathbf{u}^{NL}(t^{i-1})$ , the more accurately the nonlinearities should be captured. Furthermore, it is expected that the nonlinear strains can be determined by linear theory with sufficient accuracy if



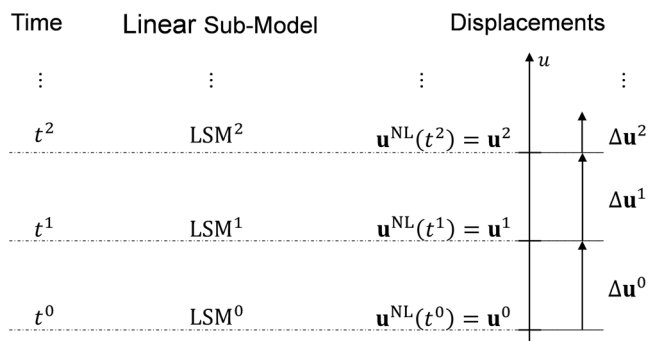


Fig. 3 Relationship between absolute and incremental displacements

one chooses the difference between time step  $t^i$  and  $t^{i-1}$  adequately small. Note that the linear strains of each LSM can be considered as “true strains” because they are based on deformed geometries. In addition, the DiESL approach can also capture the dependencies of the stiffness matrix on the displacements, because each stiffness matrix  $\mathbf{K}^i$  is now based on the deformed structure at time step  $t^i$ .

Summarizing, the DiESL approach solves the following optimization problem:

$$\min f(\mathbf{x}, \Delta \mathbf{u}^0(\mathbf{x}), \dots, \Delta \mathbf{u}^{n_T-1}(\mathbf{x})) \tag{13a}$$

$$\text{s.t. } g_j(\mathbf{x}, \Delta \mathbf{u}^0(\mathbf{x}), \dots, \Delta \mathbf{u}^{n_T-1}(\mathbf{x})) \leq 0; j = 1, \dots, m \tag{13b}$$

$$\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U; \mathbf{x} \in \mathbb{R}^n \tag{13c}$$

where  $\Delta \mathbf{u}^i$  is the solution of

$$\mathbf{K}^i(\mathbf{x}) \Delta \mathbf{u}^i = \Delta \mathbf{f}_{\text{DiESL}}^i; i = 0, \dots, n_T-1 \tag{13d}$$

in  $\text{LSM}^i$ , the equivalent static loads  $\Delta \mathbf{f}_{\text{ESL}}^i$  are computed as

$$\Delta \mathbf{f}_{\text{DiESL}}^i = \mathbf{K}^i(\mathbf{x})(\mathbf{u}^{\text{NL}}(t^{i+1}) - \mathbf{u}^{\text{NL}}(t^i)); i = 0, \dots, n_T-1 \tag{13e}$$

and  $\mathbf{u}^{\text{NL}}(t^i)$  is the solution of

$$\mathbf{M}(\mathbf{x}) \ddot{\mathbf{u}}^{\text{NL}}(t) + \mathbf{C}(\mathbf{x}) \dot{\mathbf{u}}^{\text{NL}}(t) + \mathbf{K}^{\text{NL}}(\mathbf{x}, \mathbf{u}^{\text{NL}}(t)) \mathbf{u}^{\text{NL}}(t) = \mathbf{f}(t) \tag{13f}$$

for the selected time steps  $t^i$ .

In the following, it is described in more detail how different kinds of structural responses are handled in the DiESL approach.

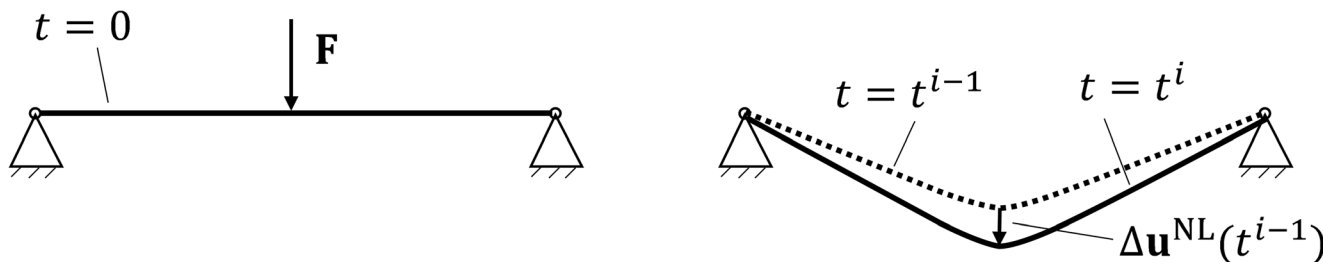


Fig. 4 Different states of deformation in three-point bending illustrating the benefits of the DiESL approach

### 3.1 Computation of displacements

During optimization of all LSMs using MMO, each LSM analysis yields incremental displacements. The total linear displacement of a node at time  $t^i$  is used as an approximation of the respective nonlinear displacement  $\mathbf{u}^i \approx \mathbf{u}^{\text{NL}}(t^i)$ . It can be computed recursively as

$$\mathbf{u}^i = \mathbf{u}^{i-1} + \Delta \mathbf{u}^{i-1} \tag{14}$$

where  $\Delta \mathbf{u}^{i-1}$  is the solution of (12) for  $\text{LSM}^{i-1}$ . Accordingly, the accumulated displacements can be calculated as

$$\mathbf{u}^i = \mathbf{u}^0 + \sum_{j=0}^{i-1} \Delta \mathbf{u}^j. \tag{15}$$

In general,  $\mathbf{u}^0 = \mathbf{0}$  applies. This accumulation is processed by the MMO, which handles all LSMs and accumulates their solutions  $\Delta \mathbf{u}^i$ .

### 3.2 Computation of strains and stresses

Strains and stresses can be accumulated in a similar way as the displacements. The procedure is shown in the following for strains. A strain component  $\varepsilon^i$  in  $\text{LSM}^i$  is calculated as

$$\varepsilon^i = \varepsilon^{i-1} + \alpha^{i-1} \Delta \varepsilon^{i-1}. \tag{16}$$

Here, the scaling factor  $\alpha$  is introduced to obtain a better description of nonlinearities such as geometric nonlinearity and plastic material behavior. The factor is calculated as

$$\alpha^{i-1} = \frac{\varepsilon^{\text{NL}}(t^i) - \varepsilon^{\text{NL}}(t^{i-1})}{\Delta \hat{\varepsilon}^{i-1}} \tag{17}$$

where  $\Delta \hat{\varepsilon}^{i-1}$  in (17) is the linear static strain value in  $\text{LSM}^{i-1}$  computed at the beginning of the linear static optimization (i.e., iteration 0) in each cycle. The nonlinear true strains  $\varepsilon^{\text{NL}}(t^i)$  and  $\varepsilon^{\text{NL}}(t^{i-1})$  are obtained from the nonlinear analysis. The accumulated strain can be calculated as

$$\varepsilon^i = \varepsilon^0 + \sum_{j=0}^{i-1} \alpha^j \Delta \varepsilon^j \tag{18}$$

where in general  $\varepsilon^0 = 0$  applies. Note that the linear static strain  $\varepsilon^i$  is also a true strain, at least in good approximation,

because it is computed based on the respective deformed geometries in each LSM. Hence, the correction factor  $\alpha$  for strains is not necessary if the time difference between adjacent LSMs is small enough. This is not necessarily true for stresses in the plastic area because the effective plastic yield stress tangent modulus differs strongly from the linear Young's modulus.

### 3.3 Computation of velocities and accelerations

Generally, velocities and accelerations can be computed in good approximation as finite differences of displacements between adjacent time steps if the time step in between is sufficiently small. In DiESL, this is simplified because the solution  $\Delta \mathbf{u}^i$  of LSM<sup>*i*</sup> is already a difference. Consequently, the velocity  $\dot{\mathbf{u}}^i$  at time step *t<sup>i</sup>* can simply be computed as forward difference by

$$\dot{\mathbf{u}}^i = \frac{\Delta \mathbf{u}^i}{\Delta t^i} \tag{19}$$

where  $\Delta t^i$  is the time difference between LSM<sup>*i*</sup> and LSM<sup>*i*+1</sup>. Hence, the acceleration can be calculated as

$$\ddot{\mathbf{u}}^i = \frac{\dot{\mathbf{u}}^i - \dot{\mathbf{u}}^{i-1}}{\Delta t^{i,i-1}}; \quad \Delta \ddot{\mathbf{t}}^{i,i-1} = \frac{\Delta t^i + \Delta t^{i-1}}{2}. \tag{20}$$

### 3.4 Implementation

In a first step, a Python code of the ESLM (Spohner 2018; Triller 2019) had been developed. In a second step, this code was extended to handle both ESL and DiESL methods. It uses the commercial solver LSTC LS-DYNA (LSTC 2015) for nonlinear dynamic analysis and OptiStruct (HyperWorks A 2012) for the computation of the ESLs and for the linear static response optimization. Furthermore, the same termination criteria were used for both methods ESLM and DiESL. The first criterion is that the design has to be feasible. Since the linear static subproblem only provides estimations for the actual problem, the convergence check has to be performed after the nonlinear dynamic analysis. However, the responses of the linear static analysis often differ from those of the nonlinear dynamic analysis. In some cases, to avoid unnecessary cycles with little changing design variables, a small constraint violation may be tolerated. Hence, the implemented first convergence criterion is that the maximum normalized constraint violation  $g_{\max}$  has to be smaller than a specified limit  $\delta$ :

$$g_{\max} \leq \delta \tag{21}$$

The second criterion is that the relative change of the objective function between two subsequent cycles *k* - 1 and *k* has to be smaller than a given value  $\epsilon$ :

$$\frac{|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k-1)})|}{|f(\mathbf{x}^{(k)})|} \leq \epsilon \tag{22}$$

The values  $\delta = 0.01$  and  $\epsilon = 0.01$  have been used for both test problems.

For the limitation of the length of the optimization path in the inner loop, in the current implementation of the ESL and DiESL method, the same strategy is used as in the commercial code ESLDYNA (VRAND 2012) based on two parameter sets. The first set controls the move limits of the current cycle *k*:

$$\begin{aligned} \hat{x}_i^{U,(k)} &= \min(x_i^U, x_i^{(k-1)} + D|x_i^{(k-1)}|) \\ \hat{x}_i^{L,(k)} &= \max(x_i^L, x_i^{(k-1)} - D|x_i^{(k-1)}|); D \in [0, 1] \end{aligned} \tag{23}$$

Parameter *D* controls the size of the current move limits in cycle *k* by means of two parameters *D*<sub>ini</sub> and  $\beta_{\text{red}}$ . The parameter *D*<sub>ini</sub> corresponds to the initial value of *D* in (23). The reduction factor  $\beta_{\text{red}} \in (0, 1]$  is used to control how *D* changes from cycle to cycle (outer loop)

$$\begin{aligned} D(1) &= D_{\text{ini}}; k = 1 \\ D(k) &= D(1)\beta_{\text{red}}^{k-1}; k > 1. \end{aligned} \tag{24}$$

In this publication, the parameter set *D*<sub>ini</sub> = 0.2 and  $\beta_{\text{red}}$  = 0.9 has been used in all test problems. The other parameter set controls the number of iterations (inner loop) by the parameters *max*<sub>iter<sub>ini</sub></sub> and *max*<sub>iter</sub> defining the maximum number of iterations per cycle in the first and in subsequent cycles (outer loop), respectively. For all examples shown below, the parameter set *max*<sub>iter<sub>ini</sub></sub> = *max*<sub>iter</sub> = 2 has been used. This means that each cycle contains 2 iterations.

Very often, the FE mesh topologies in the nonlinear dynamic and the linear model are not identical. In such a case, all the coordinates and responses are mapped from the nonlinear dynamic to the linear static mesh before applying the formulas described above. In the test problems below, this was not necessary because the mesh topologies in the LS-DYNA and the OptiStruct model were identical.

Due to the fact that DiESL does not start from a single undeformed initial model but from deformed structures related to the different LSMs, it may happen that the linear static response optimization terminates with an error during the calculation of the ESLs due to excessively deformed elements and thus poor element quality. In order to realize a robust application of DiESL, an automatic repair mechanism for the mesh of the affected LSMs has been developed by deleting the failed elements. Note that such failed elements typically do not appear in all but only in a few LSMs at very early cycles only. For this reason, the deleted elements of the previous cycle are added back

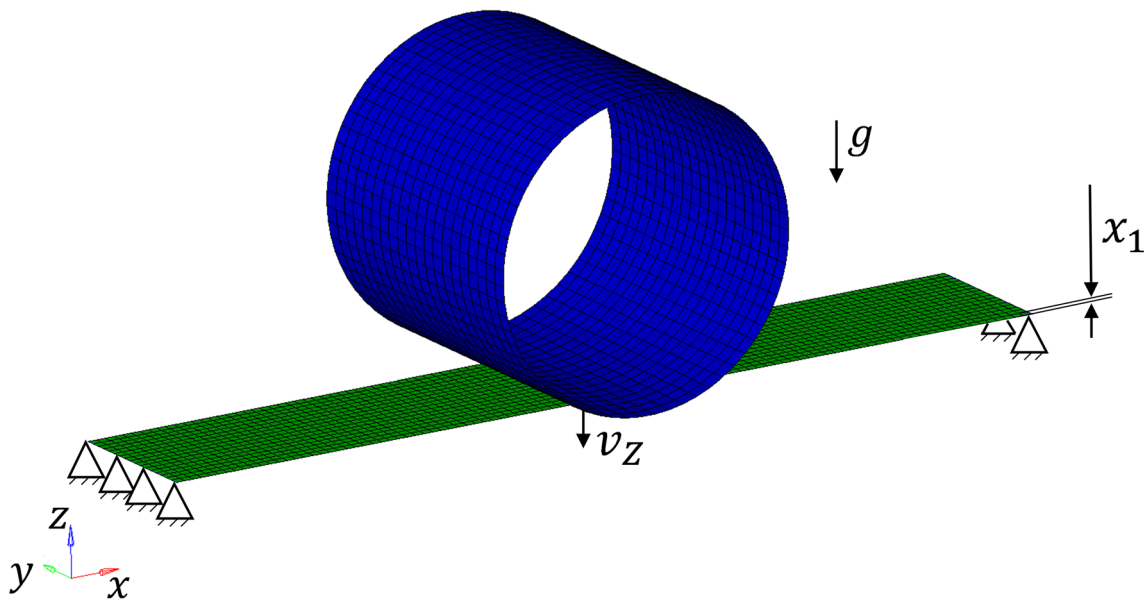


Fig. 5 Three-point bending test setup

at the beginning of each cycle and it is checked again if there are failed elements. Furthermore, it should be noted that in all applications in this paper, failed elements appeared only in the early cycles at 3 of the 20 DOE (Design of Experiments) points described in chapter 4.2. Here, some initial thicknesses were very low leading to huge deformations and thus distorted elements.

The overall flow scheme explained before is implemented as follows:

- Step 1: Set initial design variables and parameters ( $k=0$ ,  $\mathbf{x}^{(k=0)}$ ,  $\delta$ ,  $\epsilon$ ,  $D_{ini}$ ,  $\beta_{red}$ ,  $max_{iter_{ini}}$ ,  $max_{iter}$ )
- Step 2: Perform nonlinear dynamic analysis with  $\mathbf{x}^k$
- Step 3: If  $k > 0$ , check criteria of convergence for  $t^i$ : if (21) and (22) are satisfied then terminate the process.
- Step 4: If the mesh of the model used for nonlinear dynamic analysis and linear static response optimization is not

identical, map nonlinear displacements to linear static mesh.

- Step 5: Calculate the incremental displacements  $\Delta \mathbf{u}^{NL}(t^i)$  and the node coordinates  $\mathbf{r}^T(t^i)$  of all LSMs for all selected time steps  $t^i$
- Step 6: Check the element quality of each LSM FE mesh. If check was not successful, delete failed elements in respective LSM and repeat Step 6 for the remaining mesh.
- Step 7: Calculate the incremental equivalent static loads  $\Delta \mathbf{f}_{DIESEL}^i$
- Step 8: Update the move limits  $D$  according to (24)
- Step 9: Solve the linear static response optimization problem with the incremental equivalent static loads  $\Delta \mathbf{f}_{DIESEL}^i$
- Step 10: Update the design variables in the nonlinear model, set  $k = k + 1$ , and go to Step 2

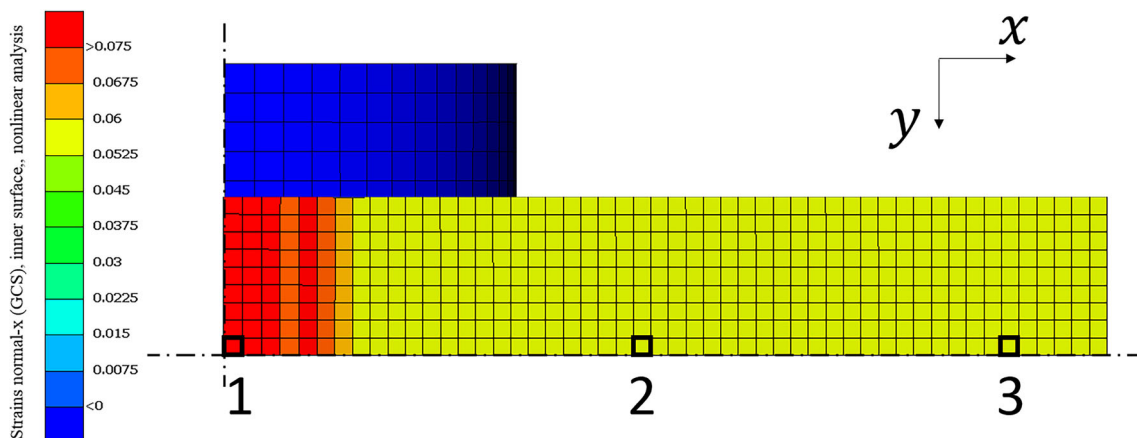


Fig. 6 Quarter FE model (bottom view in  $z$ -direction) of the three-point bending test used for linear and nonlinear analysis and three representative elements 1, 2, and 3



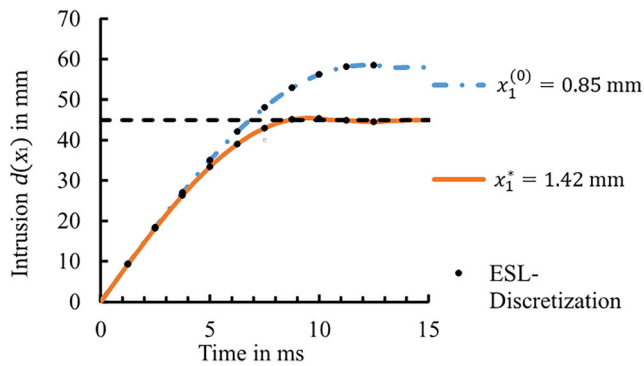


Fig. 7 Intrusion  $d(x_1)$  over time for the initial design and optimal design

Note that the scheme of the ESL algorithm is similar:  $\Delta \mathbf{f}_{\text{DIESL}}^i$  is replaced by  $\mathbf{f}_{\text{ESL}}^i$  above, and Steps 5 and 6 can be omitted.

### 4 Test problems

Two test problems for comparison of the two methods ESLM and DiESL are presented in this chapter. The first test problem is a simple 1-dimensional academic example with only one design variable for which the optimum can be determined graphically or by applying bisection. It has the advantage that the optimal solution can be computed easily in advance and that there are no local minima. Therefore, it is an ideal test example to compare the convergence speed of ESLM and DiESL without distorting influences such as becoming trapped in a local minimum. Furthermore, it is well suited for the approximation capability of structural responses such as strains. The second test problem is a more practical example because a simplified model for a crash side impact problem with 7 design variables is used. Both test problems represent sizing optimization problems in which the mass of the structure is to be minimized, while the intrusion of an impactor is constrained. A contact formulation for the contact between impactor and structure is used in the LS-DYNA and the

OptiStruct model for both test problems. Consequently, the displacement of the impactor is used as response for the constraint in each example. For both test problems the averaged run times per cycle are given in tables 3 and 4 in the appendix

For the second test problem, we compare both ESLM and DiESL results with the result of the commercial code LSTC LS-OPT (LSTC 2015). LS-OPT uses the Successive Response Surface Method (SRS) (Roux et al. 1998; Stander and Craig 2002) which is a metamodel-based approach. It starts with a DOE in the current subdomain, which is the whole design space at the beginning of the optimization. After computing all responses of interest at the sample points, the metamodels are created in the current subdomain for both objective function and constraints, and then, the optimization problem based on the metamodels is solved. The optimized point becomes the center of the new subdomain (panning). Additionally, the size of the subdomain is reduced (zooming) during the iterations. This procedure is repeated until termination criteria are fulfilled. We used the recommended default settings in LS-OPT, which are DOE creation with 50% oversampling and D-optimal samples (Roux et al. 1998) and a linear regression as metamodel.

#### 4.1 Three-point bending test

As shown in Fig. 5, a rigid cylindrical impactor with diameter = 100 mm,  $mass_{\text{impactor}} = 1.256 \text{ kg}$  and initial speed  $v_z = -7.5 \text{ m/s}$  strikes a rectangular sheet metal plate ( $length = 300 \text{ mm}$ ,  $width = 25.47 \text{ mm}$ ,  $thickness = x_1$ , Young's modulus  $E = 210 \text{ GPa}$ , density  $\rho = 7850 \text{ kg/m}^3$ , LS-DYNA element type = fully integrated shells) which has fixed translational degrees of freedom in  $x$ - and  $z$ -direction on both ends. Bilinear plastic material (tangent modulus = 0.6 GPa, yield stress = 0.25 GPa) is assigned to the sheet metal. The impactor is constrained to motion in  $z$ -direction only. Both impactor and plate are subjected to gravitation  $g = 9.81 \text{ m/s}^2$ .

The FE model used for modeling is shown in Fig. 6. In order to reduce the computational costs of the model, only a

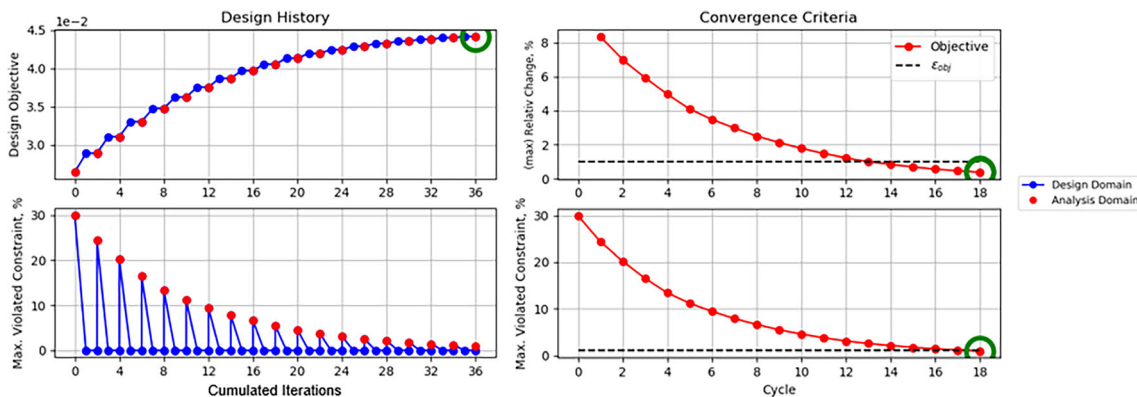
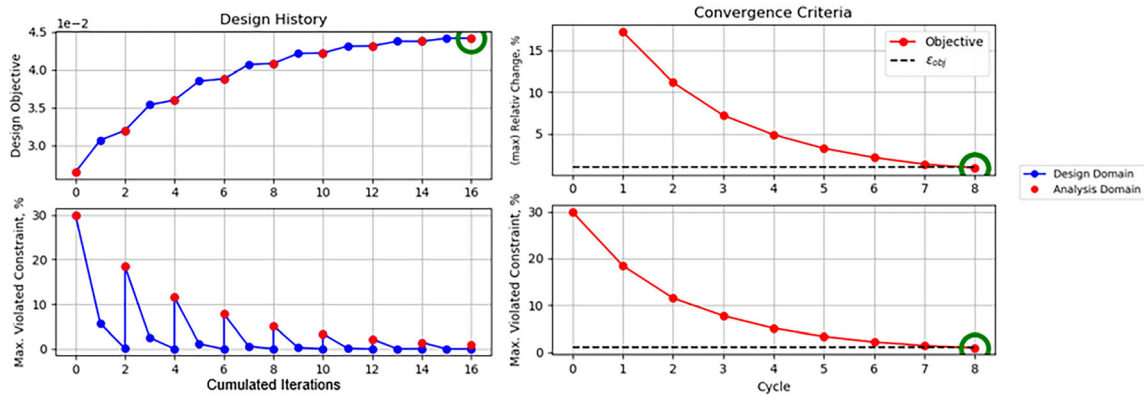


Fig. 8 Objective function and maximum relative constraint violation over cumulated iterations (left) and convergence criteria over cycles (right) using ESL



**Fig. 9** Objective function and maximum relative constraint violation over cumulated iterations (left) and convergence criteria over cycles (right) using DiESL

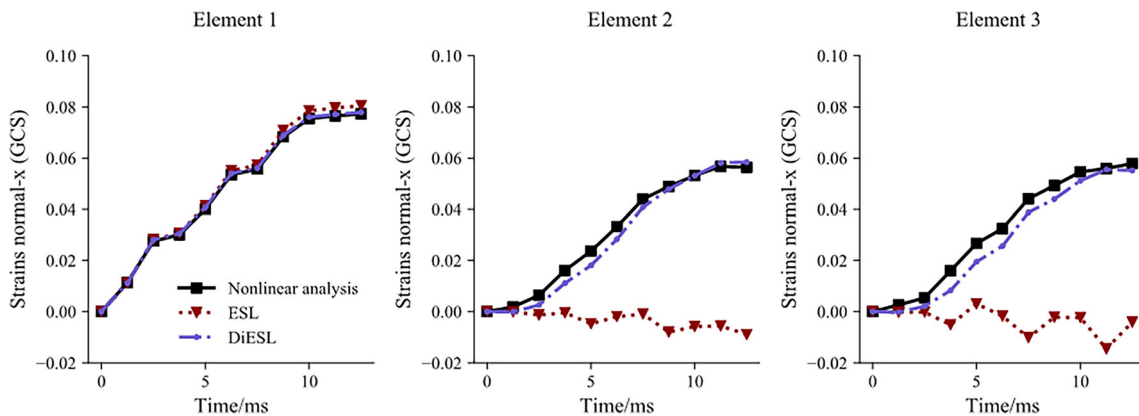
quarter of the whole model is used by applying symmetry conditions. Both linear and nonlinear dynamic analyses are based on the same mesh, and contact between impactor and plate is defined in both models. The plate’s elements are colored by the value of the normal strains in global  $x$ -direction as obtained from the nonlinear dynamic simulation at time  $t = 12.5$  ms for the initial configuration. The majority of the plate shows a relatively homogeneous tensile strain distribution and the superimposed bending strains on the left where the plate wraps around the rigid impactor. Three elements near the longitudinal symmetry line were chosen as representative elements for detailed strain analysis: (1) plate center near the impactor, (3) near the clamping (but not too close), and (2) in between.

The objective of the optimization problem is to minimize the mass of the sheet metal while the intrusion  $d$  of the impactor must not exceed 45 mm. The thickness of the sheet metal  $x_1 \in [0.85 \text{ mm}, 2 \text{ mm}]$  is the only design variable. Consequently, the optimization problem reads as

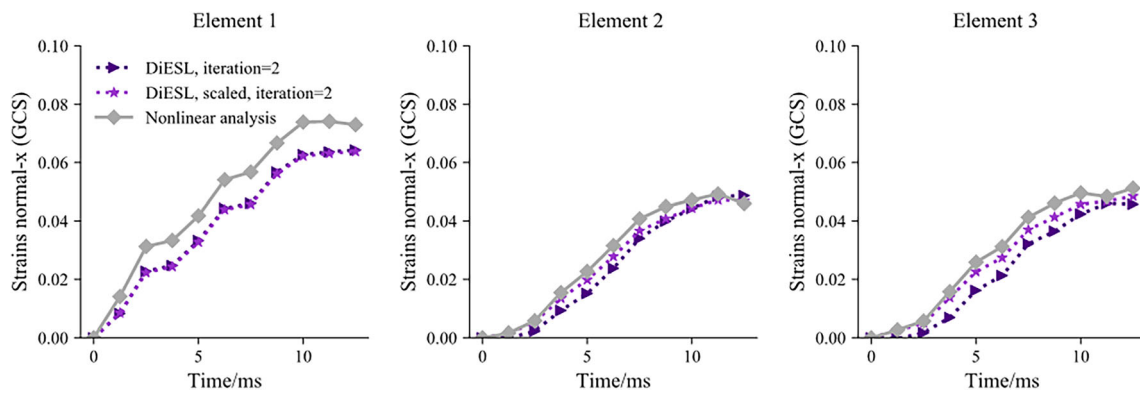
$$\begin{aligned} & \min \text{mass}(x_1) \\ \text{suchthat} \quad & d(x_1) \leq 45 \text{ mm} \\ & 0.85 \text{ mm} \leq x_1 \leq 2 \text{ mm} \end{aligned}$$

For the selection of ESL times, the following consideration had been made: For the initial value  $x_1^{(0)} = 0.85 \text{ mm}$ , the maximum intrusion  $d(x_1^{(0)}) \approx 60 \text{ mm}$ , as illustrated in Fig. 7, occurs at time  $t \approx 12.5 \text{ ms}$ . Hence, the design is infeasible. This means that the optimizer will increase the stiffness to become feasible. Increasing stiffness will cause the maximum deflection to move to smaller times. Therefore, the last ESL time can be set to 12.5 ms to ensure that the maximal intrusions are captured. In order to resolve the maximum intrusion and to capture the nonlinear displacement path (Fig. 2, left) properly, ten uniformly distributed ESL time steps from 1.25 to 12.5 ms have been used for ESLM and DiESL. This results in ten auxiliary load cases in ESLM and ten LSMs with one load case each in DiESL.

The optimization histories of the ESL and DiESL method are shown in Figs. 8 and 9, respectively. The objective function and the normalized maximum constraint violation are plotted in the left diagrams as a function of cumulated iterations. For example, this means for the selected values of  $m_{x_{\text{iter}_{\text{mi}}}} = m_{x_{\text{iter}}} = 2$  that the cumulated iteration 4 is the second iteration in the linear static response optimization in cycle 2 and the cumulated iteration 5 is the first iteration in the linear



**Fig. 10** Normal strains in the  $x$ -direction of three representative elements over time ( $\alpha = 1, x_1 = 0.85 \text{ mm}$ ) for the lower element surface before the linear static optimization



**Fig. 11** Normal strains in the  $x$ -direction based on the DiESL approach of three representative elements over time ( $x_1 = 0.103$  mm) for the lower element surface after linear static optimization (2 iterations)

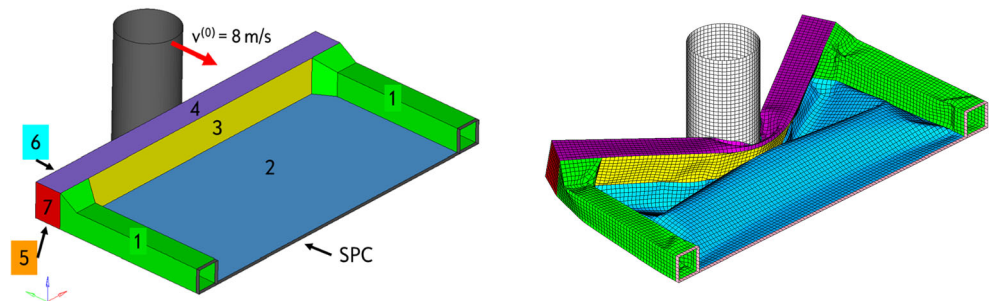
static response optimization in cycle 3. Blue circles show the values of each linear static response optimization iteration whereas red circles mark values obtained from nonlinear analyses of each cycle. The constraint violation decreases during optimization iterations until the allowed level of constraint violation is reached (1% in this example). Typically, the approximation error in the linear model can be seen at the beginning of each cycle (after every 2 iterations in our example, i.e., after iteration 2, 4, 6, ...) after recalculation of the responses by the nonlinear solver. The approximation error results in an increase of the maximum constraint violation at all even iteration numbers in the history diagram. In this case, there is no jump in the objective function, because mass can be calculated exactly in the subproblem for sizing optimization. The right diagrams plot the relative change of objective function and maximum constraint violation in the design domain over cycles together with the respective termination criteria  $\epsilon$  and  $\delta$ . We used cycles here and not cumulated iterations because these criteria are only tested once per cycle. It turns out that both methods find the optimum, but it can be seen that DiESL satisfies the termination criteria within fewer cycles than ESLM.

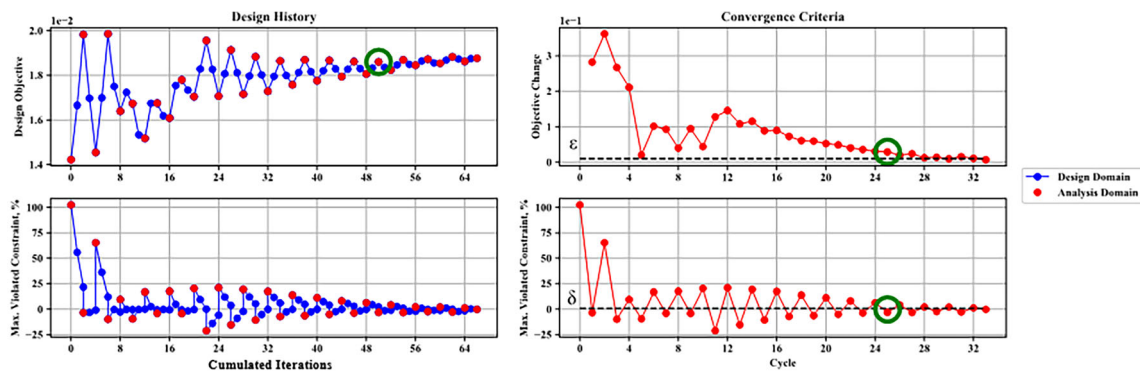
Figures 10 and 11 examine the normal strains in the  $x$ -direction in the three representative elements (see Fig. 6). In Fig. 10, each subfigure shows the strains from the nonlinear dynamic analysis as well as from the DiESL and ESL approach at the beginning of the linear static response optimization. The order of legend entries reflects the chronological

order of execution: first, the nonlinear dynamic analysis is executed. Then, the linear static analysis at the beginning of the optimization takes place (after having computed the ESLs). This means, the linear models merely reproduce the nonlinear results because no optimization has taken place yet. The first subplot shows the strains for the central element 1. Both ESLM and DiESL give excellent agreement with the nonlinear dynamic analysis. However, for elements 2 and 3, the ESLM strains no longer match with those from the nonlinear dynamic analysis, whereas the DiESL approach still provides a reasonable approximation. Note that the DiESL strains are not scaled according to Section 3.1 (i.e.,  $\alpha^l$  in (18) is set to a constant value 1) because scaled strains have to match exactly those of the nonlinear dynamic analysis by definition (i.e., at the beginning of the linear static response optimization). It is evident from Fig. 10 that for elements 2 and 3 the ESLM entirely fails to approximate the correct strains, in most cases, even the sign is reversed. The DiESL method is not highly accurate, either, but it approximates the strains much better than ESLM.

Figure 11 examines the strains at the end of a linear static response optimization (i.e., after  $max_{iter} = 2$  iterations). The nonlinear dynamic results are computed with the updated design. Again, the order of legend entries reflects the chronological order of execution. Similar to the previous findings, the strains of the DiESL method provide a good approximation for those of the nonlinear dynamic analysis. The scaling approach improves the accuracy, so that the strains of elements 2

**Fig. 12** FE model of the pole impact and labeling of design variables (left). Deformed FE model at state of maximum intrusion for initial thickness  $x_i^{(0)} = 0.8$  mm (right)





**Fig. 13** Objective function and maximum relative constraint violation over cumulated iterations (left) and convergence criteria over cycles (right) using ESLM. Note that the best design (green circle) is not obtained at the end of iterations

and 3 still match well with those from the nonlinear dynamic analysis. However, this is not the case for element 1. The reason may be that the poor displacement field approximation in iteration 2 (as seen from maximum constraint violation in iteration 2 in Fig. 9 left diagram) does affect the strains of element 1 primarily due to its position in the impact zone of the impactor. In fact, the intrusion of the impactor is underestimated and therefore the strains are as well.

## 4.2 Pole impact

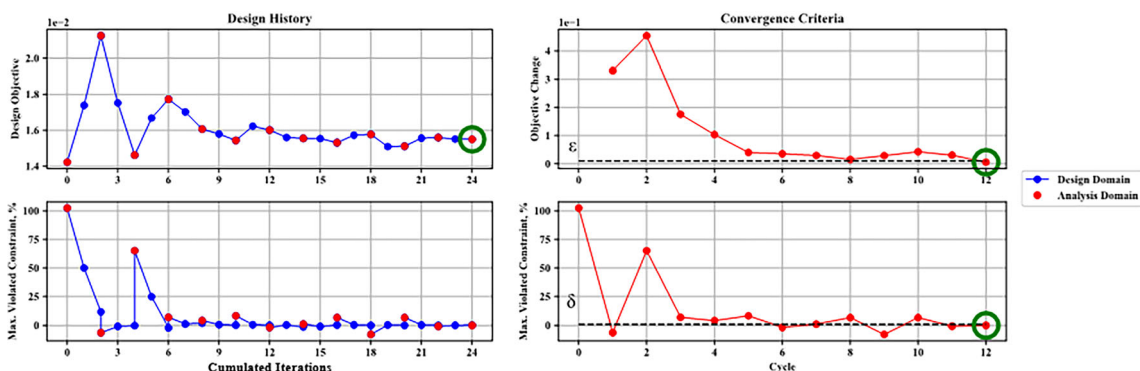
Figure 12 shows a rigid pole with initial speed  $v_y = -8$  m/s colliding with a frame structure. The translational degrees of freedom of the pole in  $x$ - and  $z$ -direction as well as all its rotations are locked. The structure is clamped along the distant edge in all 6 degrees of freedom using single-point constraints (SPC). The frame structure is made of steel (Young's modulus  $E = 210$  GPa, density  $\rho = 7850$  kg/mm<sup>3</sup>, LS-DYNA element type = Belytschko-Lin-Tsay shell elements, number of nodes = 12,813), and bilinear plastic material behavior is applied (tangent modulus = 0.6 GPa, yield stress = 0.25 GPa). The simulation end time is 100 ms. This captures the pole's maximum intrusion and part of its rebound. Consequently, the last ESL time step is selected as 100 ms.

As shown in Fig. 12, the design of the FE model is specified by seven different design variables each corresponding to a single sheet metal thickness (except  $x_1$  representing all 12 facets of both crossbars and  $x_7$  representing both end caps of rocker profile). As before, the same mesh is used for both linear and nonlinear analysis to avoid result mapping and a contact is defined between the frame structure and the pole. The objective is to minimize the mass of the frame structure while constraining the maximum intrusion of the pole in  $y$ -direction  $d(\mathbf{x})$ . Mathematically, the problem is given as

$$\begin{aligned} & \min \text{mass}(\mathbf{x}) \\ & \text{such that} \quad d(\mathbf{x}) \leq 200 \text{ mm} \\ & \quad 0.5 \text{ mm} \leq x_i \leq 3 \text{ mm}; i = 1, \dots, 7 \end{aligned}$$

In order to capture the maximum intrusion of the pole properly, 20 uniformly distributed ESL time steps from 5 to 100 ms have been used for both methods. The time increment  $\Delta t = 5$  ms is considered to be adequately small to neglect the error due to the time shift of the maximum intrusion  $d(\mathbf{x})$ . This results in 20 auxiliary load cases in ESL and 20 LSMs in DiESL.

In Figs. 13 and 14, the optimization history for an initial thickness of  $x_i^{(0)} = 0.8$  mm;  $i = 1, \dots, 7$  is shown. As before, DiESL satisfies the termination criteria within fewer cycles than



**Fig. 14** Objective function and maximum relative constraint violation over cumulated iterations (left) and convergence criteria over cycles (right) using DiESL



**Table 1** Optimization results using ESLM, DiESL, and LS-OPT (metamodel based approach)

Method	$x_1^*/\text{mm}$	$x_2^*/\text{mm}$	$x_3^*/\text{mm}$	$x_4^*/\text{mm}$	$x_5^*/\text{mm}$	$x_6^*/\text{mm}$	$x_7^*/\text{mm}$	$mass(\mathbf{x}^*)/\text{kg}$	Nonlinear dynamic analyses
ESL	0.513	0.707	0.5	1.172	2.703	2.95	0.511	18.75	33
DiESL	0.5	0.5	0.5	2.9	2.68	0.5	0.5	15.50	12
LS-OPT	0.5	0.5	0.5	2.63	3.0	0.5	0.69	15.50	131

ESLM. In Table 1, the resulting design variables are presented as well as the corresponding  $mass(\mathbf{x}^*)$  and the number of nonlinear dynamic analyses required. Additionally, the optimization result obtained by LS-OPT is given. It is remarkable that the best point for ESLM did not occur at the end of iterations. This means that ESLM converged to a point which is obviously not the nonlinear optimum. Obviously, the best point is found by accident and the convergence behavior is forced by the continuous reduction of the move limits only. In order to confirm whether the solution found by ESLM is an optimum of the nonlinear problem, an LS-OPT optimization run was executed using the ESLM solution as an initial design. It turned out that LS-OPT converged to a very different optimum. This shows that ESLM in this case was not capable of finding the actual optimum of the nonlinear problem. In contrast, DiESL and LS-OPT yield a similar design, which is significantly lighter than the one obtained from ESLM (note that due to their equal size and relative position with respect to the impactor, panels 4 and 5 have a similar effect on both mass and intrusion). This confirms that DiESL really found a nonlinear optimum in a good approximation. But it is also obvious that there is a big difference in the number of nonlinear dynamic analyses for DiESL and LS-OPT because the number of nonlinear dynamic analyses required by LS-OPT is more than 10 times as high as that required for DiESL. Note that the number of nonlinear analyses in LS-OPT scales with the number of design variables in each iteration due to the DOE. In contrast, only one nonlinear analysis per cycle is required in DiESL.

It is remarkable that five of seven design variables are at their bounds in the DiESL optimum, which is not the case for ESLM. In particular, the design variables  $x_4^*$  and  $x_6^*$  differ considerably. ESLM produces a more bending resistant design by increasing  $x_6^*$ , the thickness of the panel in contact with the pole. This can be attributed to the fact that every ESLM auxiliary load case refers to the undeformed initial structure and therefore is loaded with pure bending. No tensile loading occurs in panel 6 in any ESLM subcase. ESLM therefore increases the panel's thickness to compensate the missing

tensile contribution. In contrast, the DiESL auxiliary load cases take the tensile contributions into account because the stiffness matrix is assembled using the deformed geometry. Consequently, DiESL leads to a similar result as LS-Opt.

The results shown so far indicate that DiESL leads to better results with less computational effort compared to ESLM because the optimized mass  $mass_{\text{DiESL}}^* = 15.50$  kg is significantly smaller than  $mass_{\text{ESL}}^* = 18.75$  kg. However, it is well known that the optimization's course varies depending on the selected initial values of the design variables. In order to evaluate DiESL independently from the selected initial values, a multi-start optimization study was conducted: A DOE was created to generate 20 configurations of uniformly distributed design variables (space-filling design) used as initial values. The DOE was created as a Strength Two Orthogonal Array. These 20 configurations are optimized using ESLM, DiESL, and LS-OPT. Table 2 shows the average results over all configurations for both methods. The average results confirm the previous findings: the DiESL method performs significantly better than the ESLM both in terms of number of nonlinear dynamic analyses and in terms of the objective improvement.

## 5 Conclusions and outlook

In this paper, a difference-based extension of the ESLM is presented, which is called DiESL. Both methods are compared on the basis of two representative sizing optimization examples by minimizing mass with a displacement constraint on an impactor.

As expected, both methods produce the same optimal design for the three-point bending example. For DiESL, the results show a faster convergence to the optimum as well as a significant increase of the approximation quality for the true strains. This enables implementing a strain-based criterion of material failure in nonlinear dynamic response optimization for crash load cases.

**Table 2** Multi-start results averaged for ESLM, DiESL, and LS-OPT (metamodel-based approach)

Method	Average number of nonlinear dynamic analyses	Average mass*/kg
ESL	28.2	18.60
DiESL	13.35	15.67
LS-OPT	127.1	15.79



The second example is a pole with an initial velocity impacting a frame structure. The thicknesses of seven different parts are optimized. Here, DiESL reduces the number of nonlinear dynamic analyses required for convergence significantly. At the same time, DiESL leads to a considerably lighter design than the ESLM. The ESLM results in a more bending resistant design whereas DiESL produces a more tensile resistant one. To rule out that this result depends on the initial values of the design variables, a DOE-based multi-start optimization was executed in which the initial values of the design variables were varied. The results reconfirmed the previously determined observations. Additionally, the optimal design obtained with the DiESL method was confirmed to be an optimum of the nonlinear dynamic problem by solving the optimization using LS-OPT utilizing a metamodel-based approach. Furthermore, it was shown that the solution determined with ESLM is not an optimum of the nonlinear dynamic problem. However, the number of nonlinear dynamic analyses required when using the metamodel is about 10 times as high as that required for DiESL.

A critical point in DiESL is that due to excessively deformed elements and thus poor element quality in intermediate linear FE models, the linear solver was terminated with an error during the calculation of the ESL forces. In order to realize a robust application of DiESL, a solution has to be developed to automatically repair the mesh of the LSMs if such poor element quality occurs. This could, for example, be realized by deleting failed elements, splitting warped quad-elements into tria-elements, or by remeshing critical areas. The deletion approach has been successfully implemented and applied.

Summarizing, the test examples confirm the expectation that DiESL enables a significant increase in approximation quality of displacements and strains from nonlinear dynamic problems while simultaneously providing faster convergence. An extension of the methodology to other structural responses such as stresses and forces seems promising and would open up the way for other types of constraints. Additionally, the adaption of Young's modulus in the LSMs on element level addressing local plasticity should lead to a better stiffness representation and thus more realistic stresses. Furthermore, for optimization problems like those considered in this paper—where the maximum of a response is needed—the implementation of an automatic and adaptive selection of the ESL times at the beginning of each inner loop can help to make sure that the constrained response is well captured. The upper value of the ESL-time range can then be chosen as the time of the maximum of the response of interest plus a defined fraction (e.g., 10%). If there is more than one response needed then the maximum value of all is chosen. The optimal ESL time increments can be computed by minimizing the difference between the polygonal line of the DiESL approximation and the real response curve.

When using the ESLM, the optimization's convergence depends significantly on the chosen move limits. If the move limits are too big, the convergence is not guaranteed anymore. In case of DiESL, the approximation quality seems to be sufficiently accurate to justify larger move limits and therefore a reconsideration of the chosen move limit strategy.

Compared to a metamodel-based approach, the number of nonlinear dynamic analyses required is remarkably lower to optimize such nonlinear dynamic problems. Furthermore, the number of nonlinear dynamic analyses required for DiESL is not directly dependent on the number of design variables. Therefore, DiESL may enable topology optimization for nonlinear dynamic problems such as crash tests.

**Acknowledgments** Open Access funding enabled and organized by Projekt DEAL. This work was funded by the European Union under the Horizon 2020 program, grant agreement number 723893. We would like to thank the reviewers for their helpful comments and efforts towards improving this manuscript.

## Compliance with ethical standards

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

**Replication of results** With the exception of the information already given in the paper, no further information on the reproducibility of the results can be provided due to the confidentiality agreement of the Opel Automobile GmbH.

## Appendix

**Table 3** Runtimes per cycle for the three-point bending example for nonlinear dynamic analysis and linear static response optimization on the following platform: 48 CPU: Intel(R) Xeon(R) Silver 4116 CPU @ 2.10 GHz, CPU speed 2100 MHz, 49,368 MB RAM, 130,468 MB swap

Process	Elapsed Time/s	Cores applied
Nonlinear dynamic analysis	7.2	8
Linear static response optimization	<i>ESL</i>	4.0
	<i>DiESL</i>	4.1

**Table 4** Runtimes per cycle for side-impact example for nonlinear dynamic analysis and linear static response optimization on the following platform: 48 CPU: Intel(R) Xeon(R) Silver 4116 CPU @ 2.10 GHz, CPU speed 2100 MHz, 49,368 MB RAM, 130,468 MB swap

Process	Elapsed Time/s	Cores applied
Nonlinear dynamic analysis	146.5	8
Linear static response optimization	<i>ESL</i>	24.8
	<i>DiESL</i>	40.0

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

## References

- Choi WS, Park GJ (2002) Structural optimization using equivalent static loads at all time intervals. *Comput Methods Appl Mech Eng* 191: 2105–2122. [https://doi.org/10.1016/s0045-7825\(01\)00373-5](https://doi.org/10.1016/s0045-7825(01)00373-5)
- Choi WH, Lee Y, Yoon JM et al (2018) Structural optimization for roof crush test using an enforced displacement method. *Int J Automot Technol* 19:291–299. <https://doi.org/10.1007/s12239-018-0028-x>
- Cressie N (1988) Spatial prediction and ordinary kriging. *Math Geol* 20: 405–421
- Cressie N (1989) Geostatistics. *Am Stat* 43:197–202
- Cressie N (1990) The origins of kriging. *Math Geol* 22:239–252
- Hong EP, You BJ, Kim CH, Park GJ (2010) Optimization of flexible components of multibody systems via equivalent static loads. *Struct Multidiscip Optim* 40:549–562. <https://doi.org/10.1007/s00158-009-0384-2>
- Hornik K, Stinchcombe M, White H (1990) Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw* 3:551–560
- Hornik K, Stinchcombe M, White H (1992) Universal approximation of an unknown mapping and its derivatives. Blackwell, Inc, Cambridge
- HyperWorks A (2012) OptiStruct-12.0 user's guide. Altair Engineering, Troy
- Jang HH, Lee HA, Lee JY, Park GJ (2012) Dynamic Response Topology Optimization in the Time Domain Using Equivalent Static Loads. *AIAA J* 50: 226–234
- Jeong SB, Yi SI, Kan CD et al (2008) Structural optimization of an automobile roof structure using equivalent static loads. *Proc Inst Mech Eng Part D J Automob Eng*. <https://doi.org/10.1243/09544070JAUTO855>
- Jeong SB, Yoon S, Xu S, Park GJ (2010) Non-linear dynamic response structural optimization of an automobile frontal structure using equivalent static loads. *Proc Inst Mech Eng Part D J Automob Eng* 224:489–501
- Karev A, Harzheim L, Immel R, Erzgräber M (2018) Comparison of Different Formulations of a Front Hood Free Sizing Optimization Problem Using the ESL-Method. In: Schumacher A, Vietor T, Fiebig S, et al. (eds) *Advances in Structural and Multidisciplinary Optimization*. Springer International Publishing, Cham: 933–951
- Karev A, Harzheim L, Immel R, Erzgräber M (2019) Free sizing optimization of a front hood using the ESL method: overcoming challenges and traps. *Struct Multidiscip Optim*. <https://doi.org/10.1007/s00158-019-02285-9>
- Kim YI, Park GJ (2010) Nonlinear dynamic response structural optimization using equivalent static loads. *Comput Methods Appl Mech Eng* 199:660–676. <https://doi.org/10.1016/j.cma.2009.10.014>
- Lee H-A, Park G-J (2015) Nonlinear dynamic response topology optimization using the equivalent static loads method. *Comput Methods Appl Mech Eng* 283:956–970
- Lee H-A, Kim Y-I, Park G-J et al (2007) Structural optimization of a joined wing using equivalent static loads. *J Aircr* 44:1302–1308
- Lee JJ, Jung UJ, Park GJ (2013a) Shape optimization of the workpiece in the forging process using equivalent static loads. *Finite Elem Anal Des* 69:1–18. <https://doi.org/10.1016/j.finel.2013.01.005>
- Lee S-J, Lee H-A, Yi S-I et al (2013b) Design flow for the crash box in a vehicle to maximize energy absorption. *Proc Inst Mech Eng Part D J Automob Eng* 227:179–200
- LSTC (Livermore Software Technology Corporation) (2015) LS-OPT User's Manual, Version 5.2, Livermore
- Matheron G (1963) Principles of geostatistics. *Econ Geol* 58:1246–1266
- Park GJ (2011) Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Struct Multidiscip Optim* 43:319–337. <https://doi.org/10.1007/s00158-010-0530-x>
- Park GJ, Kang BS (2003) Validation of a structural optimization algorithm transforming dynamic loads into equivalent static loads. *J Optim Theory Appl* 118:191–200. <https://doi.org/10.1023/A:1024799727258>
- Park GJ, Lee Y (2019) Discussion on the optimality condition of the equivalent static loads method for linear dynamic response structural optimization. *Struct Multidiscip Optim* 59:311–316. <https://doi.org/10.1007/s00158-018-2059-3>
- Park KJ, Lee JN, Park GJ (2005) Structural shape optimization using equivalent static loads transformed from dynamic loads. *Int J Numer Methods Eng* 63:589–602. <https://doi.org/10.1002/nme.1295>
- Powell MJD (1992) The theory of radial basis function approximation in 1990, *Advances in Numerical Analysis II: Wavelets, Subdivision, and Radial Functions* (WA Light, ed.). Oxford University Press, Oxford: 105–210
- Roux WJ, Stander N, Haftka RT (1998) Response surface approximations for structural optimization. *Int J Numer Methods Eng* 42:517–534
- Schaback R (2002) Stability of radial basis function interpolants. *Approximation Theory X: Wavelets, Splines, and Applications*: pp 433–440. <http://num.math.uni-goettingen.de/schaback/research/papers/SoRBFI.pdf>
- Schaback R, Wendland H (2001) Characterization and construction of radial basis functions. *Multivar Approx Appl* 1–24. <https://doi.org/10.1017/CBO9780511569616.002>
- Shin MK, Park KJ, Park GJ (2007) Optimization of structures with non-linear behavior using equivalent loads. *Comput Methods Appl Mech Eng* 196:1154–1167. <https://doi.org/10.1016/j.cma.2006.09.001>
- Spohner A (2018) Erstellung und Implementierung eines Optimierungsprogramms zur Wandstärkenoptimierung unter Verwendung der "Equivalent Static Load"-Methode [Unpublished master thesis]. Technical University Darmstadt
- Stander N, Craig KJ (2002) On the robustness of a simple domain reduction scheme for simulation-based optimization. *Eng Comput* 19(4): 431–450
- Stolpe M (2014) On the equivalent static loads approach for dynamic response structural optimization. *Struct Multidiscip Optim*. <https://doi.org/10.1007/s00158-014-1101-3>

- Stolpe M, Verbart A, Rojas-Labanda S (2018) The equivalent static loads method for structural optimization does not in general generate optimal designs. *Struct Multidiscip Optim* 58:139–154. <https://doi.org/10.1007/s00158-017-1884-0>
- Triller J (2019) Improvement of the Equivalent Static Load Method Based on a Difference Approach [Unpublished master thesis]. Technical University Darmstadt
- VRAND (Vanderplaats R&D, Inc) (2012) ESLDYNA Documentation, Gardenbrook
- Waszczyszyn Z (1999) Fundamentals of artificial neural networks. In: Waszczyszyn Z (ed) *Neural Networks in the Analysis and Design of Structures*. Springer Vienna, pp 1–51. [https://doi.org/10.1007/978-3-7091-2484-0\\_1](https://doi.org/10.1007/978-3-7091-2484-0_1)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.