Electrical Engineering (2018) 100:2635–2647
https://doi.org/10.1007/s00202-018-0716-6

ORIGINAL PAPER

Robust shape optimization of electric devices based on deterministic optimization methods and finite-element analysis with affine parametrization and design elements

Ion Gabriel Ion[1,2] · Zeger Bontinck[1,2] · Dimitrios Loukrezis[1,2] · Ulrich Römer[1,3] · Oliver Lass[1,4] · Stefan Ulbrich[1,4] · Sebastian Schöps[1,2] · Herbert De Gersem[1,2,5]

Received: 11 March 2018 / Accepted: 13 July 2018 / Published online: 9 August 2018

Abstract

In this paper, gradient-based optimization methods are combined with finite-element modeling for improving electric devices. Geometric design parameters are considered by piecewise affine parametrizations of the geometry or by the design element approach, both of which avoid remeshing. Furthermore, it is shown how to robustify the optimization procedure, that is, how to deal with uncertainties on the design parameters. The overall procedure is illustrated by an academic example and by the example of a permanent-magnet synchronous machine. The examples show the advantages of deterministic optimization compared to standard and popular stochastic optimization procedures such as particle swarm optimization.

Keywords Finite-element analysis · Genetic algorithms · Gradient methods · Electric machines · Optimization methods · Particle swarm optimization · Permanent-magnet machines · Quadratic programming

1 Introduction

In almost all electric design procedures, numerical optimization is employed as one of the last design steps in order to optimize the device's performance and efficiency, to minimize its weight and size and to save on material and manufacturing costs. Often, the quality of this optimization step indirectly determines the success of the product and, hence, the market position of the company. The reliability, accuracy and computational cost of the numerical optimization procedure becomes in itself a subject of competition. This paper illustrates that shape optimization can be improved substantially when finite-element (FE) analysis procedures are equipped with piecewise affine parametrization or design elements, such that well-performing deterministic optimization methods become applicable.

Impressive technical improvements have been achieved by numerical optimization on the basis of magnetic equivalent circuits or 2D and 3D FE models. All have led to highly optimized designs, for example, for permanent-magnet synchronous machines (PMSMs) in automotive applications. Since three decades, FE-based optimization has been addressed in several text books (see, e.g., [12]) and hundreds of journal articles (see, e.g., [14] and the references therein). Although originally, gradient-based methods were

✉ Herbert De Gersem
degersem@ce.tu-darmstadt.de;
degersem@temf.tu-darmstadt.de;
herbert.degersem@kuleuven-kulak.be

Ulrich Römer
u.roemer@tu-braunschweig.de

Stefan Ulbrich
ulbrich@mathematik.tu-darmstadt.de

[1] Graduate School Computational Engineering, Technische Universität Darmstadt, Dolivostraße 15, 64293 Darmstadt, Germany

[2] Institut für Theorie Elektromagnetischer Felder, Technische Universität Darmstadt, Schlossgartenstraße 8, 64289 Darmstadt, Germany

[3] Institute of Dynamics and Vibrations, TU Braunschweig, Schleinitzstraße 20, 38106 Braunschweig, Germany

[4] Chair of Nonlinear Optimization, Department of Mathematics, Technische Universität Darmstadt, Dolivostraße 15, 64293 Darmstadt, Germany

[5] Wave Propagation and Signal Processing Research Group, KU Leuven – Kulak, Etienne Sabbelaan 53, 8500 Kortrijk, Belgium

preferred (see, e.g., [48,56,58]), already for more than two decades, stochastic algorithms are more popular (see, e.g., [19,33]). The majority of the proposed procedures opt for *stochastic* or *population-based* optimization methods, such as genetic algorithms and particle swarm optimization (see, e.g., [34]), because they allow to use FE solvers as a black box, they can easily consider geometric parameters, their parallelization is straightforward and they are more likely to find the global optimum. Stochastic algorithms have been used for robust optimization, have been applied together with surrogate modeling and have been extended to multi-objective optimization problems [3,23]. In particular for PMSMs, optimization with stochastic methods became the method of choice [2,9,51].

The trend toward stochastic optimization combined with FE analysis continues without restraint, as is illustrated by the number of according contributions at recent conferences. This paper partially counteracts this tendency by turning back to deterministic optimization algorithms. Deterministic optimization methods are known to converge faster than stochastic optimization methods, albeit possibly to a local optimum. Moreover, the analysis of gradient-based methods is more mature, allowing for a rigorous control of mesh discretization errors, for instance. The main drawback of many deterministic methods is, however, the necessity to provide derivatives, which is particularly cumbersome when optimization according to geometric parameters is pursued. This drawback is here addressed explicitly and is alleviated by piecewise affine parametrizations of the geometry or by the design element approach. The overall deterministic optimization routine is shown to outperform the most popular stochastic algorithms by factors. Moreover, the optimization method will be robustified to include uncertainties on the design parameters.

The paper is structured as follows: Sect. 2 recalls the basics of mathematical optimization. It clearly distinguishes between deterministic methods (Sect. 2.3) and particle swarm optimization as a relevant representative of stochastic methods (Sect. 2.4). Furthermore, an extension to robust optimization is discussed in Sect. 2.5. Section 3 deals with FE analysis of magnetodynamic fields. The core parts of the paper are Sect. 3.3.1 about affine parametrization and Sect. 3.3.2 about design elements, both facilitating and improving the calculation of derivatives with respect to geometric parameters. The superior performance of gradient-type deterministic optimization is illustrated for a benchmark example in Sect. 4 and for a PMSM in Sect. 5. Conclusions are formulated in Sect. 6.

2 Constrained optimization

2.1 Constrained optimization problem

The optimization is carried out with respect to I *design parameters* $\mathbf{p} = (p_1, p_2, \ldots, p_I)$ belonging to the *admissible set* $\mathcal{P}_{\mathrm{ad}} = \{\mathbf{p} \in \mathbb{R}^I | G_m(\mathbf{p}) \le 0, m = 1, \ldots, M\}$, where $G_m(\mathbf{p})$ denote the constraints. The design parameters can be any continuous variables, for example material constants, excitation parameters and geometric sizes or positions. The constraints limit the admissible range of these parameters, for example to preserve the topology of the geometry or to set physical and operational constraints. Discrete design parameters are not considered in this work, although many methods apply, for example as part of a branch-and-bound technique, to mixed-integer optimization problems as well [21].

The optimization goal is represented by the *objective function* $J(\mathbf{p})$ returning a scalar value for every set of design parameters. Relevant quantities are, for example force, torque, current, efficiency, weight, temperature or a combination thereof. When N objective functions $J_n(\mathbf{p})$, $n = 1, \ldots, N$ are relevant, a possible approach is to combine them with user-defined weight factors α_n into a single cost function $J(\mathbf{p}) = \sum_{n=1}^{N} \alpha_n J_n(\mathbf{p})$. The optimization problem then reads

$$\underset{\mathbf{p}\in\mathbb{R}^I}{\text{minimize}} \quad J(\mathbf{p}), \tag{1a}$$

$$\text{subject to} \quad G_m(\mathbf{p}) \le 0, \quad m = 1, \ldots, M. \tag{1b}$$

In this work, the evaluation of $G_m(\mathbf{p})$ and/or $J(\mathbf{p})$ involves a FE analysis of the device. Hence, the computational performance of the overall approach is heavily determined by the number of FE-solver calls.

2.2 Optimization methods

The selection of a particular optimization method consists of four essentially independent choices (see also Table 3 in [18]).

- Problem (1) considers a single optimization goal. For a *multi-objective* optimization problem, a Pareto front is calculated such that the relative importance of the optimization goals can be fixed later on [7,12]. This paper does not further consider multi-objective optimization. Nonetheless, the developed techniques are applicable to multi-objective optimization as well.
- A distinction is made between *global* optimization and *local* optimization, where the first strives to a global optimum, whereas the second may run into a local one. This paper is limited to methods for local optimization. In practice, if a global optimum is required, the methods

may be repeated for different start values, or could be embedded as part of a global optimization scheme [24].

– Especially when the evaluation of the objective function is computationally expensive, it is recommended to carry out the optimization method on the basis of a *surrogate* model (*indirect* optimization methods). Such a simplified model can be obtained by expert knowledge on the application [59], by design space reduction [17], by a response surface methodology [17] or by space mapping [29] or manifold mapping [15]. Here, a *direct* optimization procedure is used. All ideas presented here can, however, be used in combination with indirect optimization approaches as well [30].
– The result from a *nominal* optimization is a set of optimized design parameters leading to an optimum of the objective function. The optimum may, however, become irrelevant when it is highly sensitive to uncertainties in the design parameters. One speaks about *robust* optimization when the optimization is carried out taking such uncertainties into account. In this paper, both nominal and robust optimization methods are considered. An approach for *robustification* is discussed in Sect. 2.5.
– Two families of basic optimization methods exist: *deterministic* and *stochastic* methods. Among the stochastic methods, *genetic algorithms* [35], *differential evolution* [38] and *particle swarm optimization* (PSO) [27] are well known.

This paper motivates the use of a gradient-based deterministic method for nominal and robust optimization and compares it with a standard particle swarm technique.

2.3 Gradient-based deterministic method

This work proposes to solve (1) by standard sequential quadratic programming (SQP) with damped Broyden–Fletcher–Goldfarb–Shanno (BFGS) updates for the Hessian approximation [22,40]. This method establishes locally a second-order convergence, which means that

$$|J(\mathbf{p}_{k+1}) - J(\mathbf{p}_{\mathrm{opt}})| \le C|J(\mathbf{p}_k) - J(\mathbf{p}_{\mathrm{opt}})|^2 \tag{2}$$

for $C > 0$ and k the iteration step, which should be sufficiently large. The method, however, requires knowledge about the sensitivities of the objective function with respect to the design parameters, in particular, $\nabla_{\mathbf{p}} J(\mathbf{p})$ or, alternatively, a locally quadratic approximation of the objective function [44]. Many FE solution and post-processing routines do not provide this information, especially when geometric design parameters are involved. Therefore, one is tempted to approximate the sensitivities by finite differences as in, e.g., [48]. This is, however, known to be particularly cumbersome because of the limited accuracy of the finite differences [58]. Even when relying on gradient-free deterministic methods (e.g., [44,45]), artifacts caused by FE analysis may hamper the convergence of the optimization routines. Eventually, as apparently the only option, deterministic optimization algorithms are abandoned in favor of stochastic approaches. This paper, however, sticks to gradient-based deterministic methods by complementing the FE simulation procedure with sensitivity information. The problems caused by the presence of geometric parameters are alleviated by introducing piecewise affine parametrization (see Sect. 3.3.1) or, alternatively, design elements (see Sect. 3.3.2) to the FE procedure.

2.4 Particle swarm optimization

Particle swarm optimization (PSO) [27] belongs to the broad class of Stochastic algorithms and is particularly popular for optimizing electric machines (see, e.g., [2,3,9,23,34]). In PSO, a set of Q particles indicated by $q = 1, \ldots, Q$ moves through the admissible set in the design space in search of an optimum. At each iteration step k, the algorithm evaluates the objective function $J(\mathbf{p})$ in every particle position $\mathbf{p}_{k,q}$. The newly obtained values are compared to the previous best values in the individual particle histories and the best value of the entire swarm. The corresponding best sets are denoted by $\hat{\mathbf{p}}_q$ and $\hat{\mathbf{p}}_{\mathrm{swarm}}$, respectively. The velocities of the particles are updated according to

$$\mathbf{v}_q \leftarrow \underbrace{\omega_0 \mathbf{v}_q}_{1)} + \underbrace{\omega_1 \mathbf{N}_1(\hat{\mathbf{p}}_q - \mathbf{p}_{k,q})}_{2)} + \underbrace{\omega_2 \mathbf{N}_2(\hat{\mathbf{p}}_{\mathrm{swarm}} - \mathbf{p}_{k,q})}_{3)}, \tag{3}$$

where ω_0, ω_1 and ω_2 are swarm characteristic constants and $\mathbf{N}_1$ and $\mathbf{N}_2$ are two random diagonal matrices with elements in [0, 1] generated independently and uniformly for each particle at every step, representing the *free will* of the swarm. The components of the velocity update are:

1. Maintain a part of the current velocity;
2. Head toward the particle's best found point ($\hat{\mathbf{p}}_q$);
3. Head toward the swarm's best found point ($\hat{\mathbf{p}}_{\mathrm{swarm}}$).

If at some iteration there is a particle that leaves the admissible set, its position is projected on the boundary of the admissible set. Initially, all particles are randomly and uniformly distributed in the admissible set and the initial velocities are set to 0. The particle swarm is a gradient-free method and works for non-smooth functions as well. The iteration ends when a maximum number of iterations is reached, or when the majority of the particles are close enough to the best point $\hat{\mathbf{p}}_{\mathrm{swarm}}$:

$$\frac{1}{Q}\sum_{q=1}^{Q}\|\hat{\mathbf{p}}_{\text{swarm}}-\mathbf{p}_{k,p}\|_2<\epsilon, \tag{4}$$

with a user-defined tolerance ϵ, or if there is no further change in the global best point $\hat{\mathbf{p}}_{\text{swarm}}$ over N_{stall} consecutive iterations.

2.5 Robust optimization

In a *nominal* optimization procedure, one is looking for the minimum value of an objective function. However, during manufacturing, small deviations can occur on the parameters. As a consequence, the optimal solution may become suboptimal in reality. *Robust* optimization searches for an optimum that is not too much affected by the expected parameter deviations [41,60].

One possibility is to optimize such that the worst-case scenario within a stochastic set of possibilities around the optimal design parameters is the best possible. The robust counterpart of (1) adopting a worst-case scenario is

$$\underset{\mathbf{p}\in\mathbb{R}^I}{\text{minimize}}\quad \max_{\boldsymbol{\delta}\in U} J(\mathbf{p}+\boldsymbol{\delta}), \tag{5a}$$

$$\text{subject to}\quad \max_{\boldsymbol{\delta}\in U} G_m(\mathbf{p}+\boldsymbol{\delta})\leq 0,\quad m=1,\dots,M. \tag{5b}$$

Here, the uncertainty set for the deviations $\boldsymbol{\delta}$ is defined by

$$\begin{aligned} U &:= \left\{\boldsymbol{\delta}\in\mathbb{R}^n \mid \delta_i^{\mathrm{l}}\leq\delta_i\leq\delta_i^{\mathrm{u}},\ i=1,\dots,n\right\}\\ &= \left\{\boldsymbol{\delta}\in\mathbb{R}^n \mid \|\mathbf{D}^{-1}\boldsymbol{\delta}\|_\infty\leq 1\right\}, \end{aligned} \tag{6}$$

where $\mathbf{D}$ is a scaling matrix and where $\delta_i^l=-\delta_i^u$.

The nested optimization problem formulated by (5) is hard to solve. A numerically feasible optimization problem is obtained by approximating the max problem, i.e., by applying a first-order Taylor approximation of the objective function and the constraints with respect to $\mathbf{p}$ [13]:

$$J(\mathbf{p}+\boldsymbol{\delta})\approx J(\mathbf{p})+\nabla_{\mathbf{p}}J(\mathbf{p})\cdot\boldsymbol{\delta}; \tag{7}$$

$$G_m(\mathbf{p}+\boldsymbol{\delta})\approx G_m(\mathbf{p})+\nabla_{\mathbf{p}}G_m(\mathbf{p})\cdot\boldsymbol{\delta}, \tag{8}$$

for $m=1,\dots,M$. Inserting this approximation into (5), one obtains the linear approximation of the robust optimization problem:

$$\underset{\mathbf{p}\in\mathbb{R}^I}{\text{minimize}}\quad J(\mathbf{p})+\|\mathbf{D}\nabla_{\mathbf{p}}J(\mathbf{p})\|_1, \tag{9a}$$

$$\text{subject to}\quad G_m(\mathbf{p})+\|\mathbf{D}\nabla_{\mathbf{p}}G_m(\mathbf{p})\|_1\leq 0, \tag{9b}$$

for $m=1,\dots,M$. A dual norm $||\cdot||_*$ is defined by

$$\|\cdot\|_*:\quad \mathbb{R}^I\rightarrow\mathbb{R}$$

$$\mathbf{g}\mapsto\|\mathbf{g}\|_*:=\max_{\|\boldsymbol{\delta}\|\leq\mathbf{1}}\mathbf{g}^\top\boldsymbol{\delta}. \tag{10}$$

In this particular case, one can use the property that the dual of $\|\mathbf{D}^{-1}\cdot\|_\infty$ is given by $\|\mathbf{D}\cdot\|_1$.

A further problem is introduced by the fact that the norms are not differentiable, which leads to a non-smooth optimization problem. A differentiable problem is obtained by introducing $M+1$ *slack* variables $\xi_0,\dots,\xi_M$ and reformulate (9) as

$$\underset{\mathbf{p}\in R^I,\xi_0,\dots,\xi_M\in\mathbb{R}^I}{\text{minimize}}\quad J(\mathbf{p})+\mathbb{V}^\top\xi_0, \tag{11a}$$

$$\text{subject to}\quad G_m(\mathbf{p})+\mathbb{V}^\top\xi_m\leq 0, \tag{11b}$$

$$-\xi_0\leq\mathbf{D}\nabla_{\mathbf{p}}J(\mathbf{p})\leq\xi_0, \tag{11c}$$

$$-\xi_m\leq\mathbf{D}\nabla_{\mathbf{p}}G_m(\mathbf{p})\leq\xi_m, \tag{11d}$$

where $m=1,\dots,M$ and $\mathbb{V}=[1,\dots,1]^\top\in\mathbb{R}^I$. This optimization problem can now be efficiently solved numerically. Additionally to the quantities introduced in the previous section, now also second-order sensitivities with respect to the design parameters are required. This approach can be generalized to use a quadratic approximation with respect to $\mathbf{p}$ as worked out in [30].

3 Finite-element model

The behavior of the devices under consideration is determined by magnetic field phenomena and is simulated using a FE model.

3.1 Magnetoquasistatic formulation

The magnetoquasistatic (MQS) subset of Maxwell's equations is considered. The design parameters $\mathbf{p}$ influence the material distribution represented by the reluctivity $\nu(\mathbf{p})$ and the conductivity $\sigma(\mathbf{p})$, as well as the excitations, represented by the applied current density $\mathbf{J}_{\text{src}}(\mathbf{p})$ in current carrying conductors and the magnetizing field strength $\mathbf{H}_{\text{m}}(\mathbf{p})$ of the present permanent magnets. The MQS formulation in terms of the magnetic vector potential $\mathbf{A}(\mathbf{p})$ reads

$$\begin{aligned} &\nabla\times(\nu(\mathbf{p})\nabla\times\mathbf{A}(\mathbf{p}))+\sigma(\mathbf{p})\frac{\partial\mathbf{A}(\mathbf{p})}{\partial t}\\ &\quad=\mathbf{J}_{\text{src}}(\mathbf{p})-\nabla\times\mathbf{H}_{\text{m}}(\mathbf{p}), \end{aligned} \tag{12}$$

and is complemented with adequate boundary conditions. Equation 12 encompasses the case of linear, nonlinear and remanent magnetic materials expressed by

$$\mathbf{H}(\mathbf{p})=\nu(\mathbf{p})\mathbf{B}(\mathbf{p}), \tag{13}$$

$$\mathbf{H}(\mathbf{p})=\nu(\mathbf{p},\mathbf{B}(\mathbf{p}))\mathbf{B}(\mathbf{p}), \tag{14}$$

$$\mathbf{H}(\mathbf{p}) = \mathbf{H}_\mathrm{m}(\mathbf{p}) + \nu(\mathbf{p})\mathbf{B}(\mathbf{p}) \tag{15}$$

respectively. $\mathbf{H}(\mathbf{p})$ and $\mathbf{B}(\mathbf{p}) = \nabla \times \mathbf{A}(\mathbf{p})$ are the magnetic field strength and magnetic flux density. In the nonlinear setting, the formulation is treated by the Newton method, which is equivalent to using a linearized material relation $\mathbf{H}(\mathbf{p}) = \mathbf{H}_\mathrm{m}^{(k)}(\mathbf{p}) + \overline{\overline{\nu}}^{(k)}(\mathbf{p})\mathbf{B}(\mathbf{p})$ and updating the tensorial differential permeability $\overline{\overline{\nu}}^{(k)}(\mathbf{p})$ and the magnetizing field strength $\mathbf{H}_\mathrm{m}^{(k)}(\mathbf{p})$ between the successive Newton steps k [28].

3.2 Finite-element discretization

The magnetic vector potential is discretized by lowest-order Nédélec edge shape functions $\mathbf{w}_j(x, y, z)$,

$$\mathbf{A}(\mathbf{p}) \approx \sum_{j=1}^{N_\mathrm{dof}} a_j(\mathbf{p})\mathbf{w}_j, \tag{16}$$

where $a_j(\mathbf{p})$ are the degrees of freedom and N_dof is the number of degrees of freedom. In the 3D case, the shape functions are associated with the edges of a tetrahedral mesh. In the 2D Cartesian case, the edge shape functions are aligned with the z-axis and are constructed from the nodal shape functions $N_j(x, y)$ associated with the nodes of a 2D mesh:

$$\mathbf{w}_j(x, y) = \frac{N_j(x, y)}{l_z}\mathbf{e}_z, \tag{17}$$

where l_z is the length of the device in z-direction. In both cases, the discretization procedure leads to the system of equations

$$\mathbf{K}_\nu(\mathbf{p})\mathbf{a}(\mathbf{p}) + \mathbf{M}_\sigma(\mathbf{p})\frac{\mathrm{d}\mathbf{a}(\mathbf{p})}{\mathrm{d}t} = \mathbf{j}_\mathrm{src}(\mathbf{p}) + \mathbf{j}_\mathrm{m}(\mathbf{p}), \tag{18}$$

where

$$K_{\nu,i,j}(\mathbf{p}) = \int_{V_D} \nu(\mathbf{p})\nabla \times \mathbf{w}_j \cdot \nabla \times \mathbf{w}_i \,\mathrm{d}V; \tag{19}$$

$$M_{\sigma,i,j}(\mathbf{p}) = \int_{V_D} \sigma(\mathbf{p})\mathbf{w}_j \cdot \mathbf{w}_i \,\mathrm{d}V; \tag{20}$$

$$j_{\mathrm{src},i}(\mathbf{p}) = \int_{V_D} \mathbf{J}_\mathrm{src}(\mathbf{p}) \cdot \mathbf{w}_i \,\mathrm{d}V; \tag{21}$$

$$j_{\mathrm{m},i}(\mathbf{p}) = -\int_{V_D} \mathbf{H}_\mathrm{m}(\mathbf{p}) \cdot \nabla \times \mathbf{w}_i \,\mathrm{d}V, \tag{22}$$

and where V_D is the computational domain [36]. In the 2D case, $V_D = S_D \times [0, l_z]$ where S_D is the cross section of the device. Equation 18 is further discretized in time by, for example, an implicit Runge–Kutta method, linearized by the Newton–Raphson method and solved by a solution method for large sparse systems of equations [10,28].

3.3 Geometry parametrization

In the following, designs will be optimized with respect to geometric parameters. At first sight, the changing geometry necessitates the reconstruction of the computational mesh. This would, however, lead to unacceptably high computation times. Moreover, the unavoidable changes in mesh topology would introduce numerical noise which could mask the true sensitivity of the quantities of interest on the geometric parameters. Two different types of parametrizations are presented in the following. *Affine parametrization* or *affine decomposition* (see, e.g., [47]) is particularly appealing in the context of model order reduction and well suited for parallelization. However, complex transformations cannot be represented exactly by this approach. One may turn to (empirical) interpolation methods, but they introduce additional approximation errors. This is not the case for the second parametrization, which is based on the well-established concept of *design elements* [6] in combination with non-uniform rational B-splines (NURBS). Here, the mapping will not be affine and more effort is needed for the update of the FE matrices and vectors. In either case, good results can be obtained for many shape optimization problems by one of the two methods, with moderate implementation effort. It should also be mentioned that nonparametric approaches to shape optimization [11] present a viable alternative and have already been applied for electric machines [16]. There, however, advanced techniques for both derivation and implementation are needed.

The geometry is decomposed in a domain V_D^0 that is unaffected from the geometric parameters and domains $V_D^\ell(\mathbf{p})$, $\ell = 1, \ldots, L$ subject to geometry changes according to the geometric parameters $\mathbf{p}$. The FE matrices $\mathbf{K}_\nu(\mathbf{p})$ and $\mathbf{M}_\sigma(\mathbf{p})$ and vectors $\mathbf{j}_\mathrm{src}(\mathbf{p})$ and $\mathbf{j}_\mathrm{m}(\mathbf{p})$ can be partitioned accordingly:

$$\mathbf{K}_\nu(\mathbf{p}) = \mathbf{K}_\nu^0 + \sum_{\ell=1}^{L} \mathbf{K}_\nu^\ell(\mathbf{p}), \tag{23}$$

and similarly for $\mathbf{M}_\sigma(\mathbf{p})$, $\mathbf{j}_\mathrm{src}(\mathbf{p})$ and $\mathbf{j}_\mathrm{m}(\mathbf{p})$. Reference geometries $\hat{V}_D^\ell$, $\ell = 1, \ldots, L$ and $\hat{V}_D^0 = V_D^0$ are defined, as well as maps $f_\mathbf{p}^\ell : \hat{V}_D^\ell \to V_D^\ell(\mathbf{p})$, $\hat{\mathbf{r}} \mapsto f_\mathbf{p}^\ell(\hat{\mathbf{r}}) = \mathbf{r}$, which depend on $\mathbf{p}$.

3.3.1 Affine parametrization

This section discusses the idea of decomposing (a part of) the mesh such that geometrical changes can be represented by piecewise affine maps [30,47]. The affine maps will be referred to by $f_{\mathbf{p},\mathrm{aff}}^\ell$.

As an example, an affine map for the 2D case is developed (Fig. 1). Let $\mathbf{r}_j = \left[x_j \ y_j\right]^\top$, $j = 1, 2, 3$ denote the nodes of a deformed triangle in the domain $V_D^\ell(\mathbf{p})$ and $\hat{\mathbf{r}}_j = \left[\hat{x}_j \ \hat{y}_j\right]^\top$,

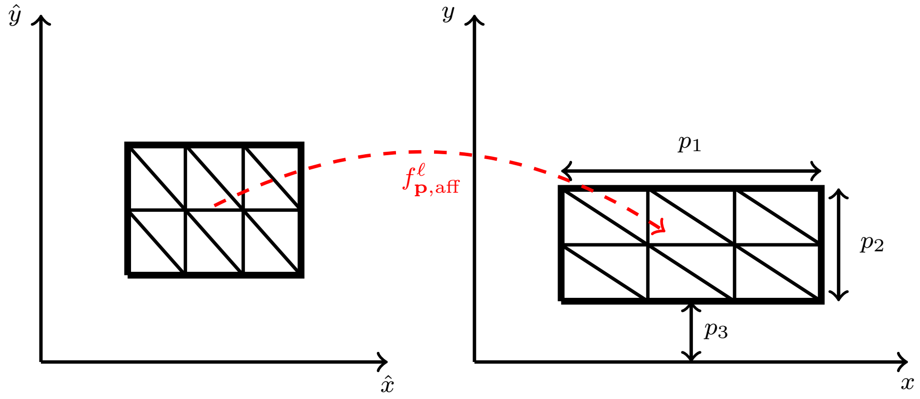

Fig. 1 Affine maps between a deformed element and a reference element

$j = 1, 2, 3$ the nodes of the corresponding triangle in the reference domain $\hat{V}^{\ell}_{D}$. The transformation from the reference triangle to the deformed triangle $\hat{\mathbf{r}} \mapsto f^{\ell}_{\mathbf{p},\mathrm{aff}}(\hat{\mathbf{r}}) = \mathbf{r}$ is given by the affine map

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \times \frac{1}{\hat{S}} \begin{bmatrix} \hat{x}_2\hat{y}_3 - \hat{x}_3\hat{y}_2 & \hat{y}_2 - \hat{y}_3 & \hat{x}_3 - \hat{x}_2 \\ \hat{x}_3\hat{y}_1 - \hat{x}_1\hat{y}_3 & \hat{y}_3 - \hat{y}_1 & \hat{x}_1 - \hat{x}_3 \\ \hat{x}_1\hat{y}_2 - \hat{x}_2\hat{y}_1 & \hat{y}_1 - \hat{y}_2 & \hat{x}_2 - \hat{x}_1 \end{bmatrix} \begin{bmatrix} 1 \\ \hat{x} \\ \hat{y} \end{bmatrix}, \tag{24}$$

with $\hat{S}$ the cross-sectional area of the reference triangle. In short, this becomes a linear transformation of the form

$$\mathbf{r} = \mathbf{N}^{\ell}_{1}(\mathbf{p}) + \mathbf{T}^{\ell}(\mathbf{p})\hat{\mathbf{r}} \qquad \text{where} \quad \hat{\mathbf{r}} \in \hat{V}^{\ell}_{D}. \tag{25}$$

In 3D, affine parametrization is organized using a tetrahedral decomposition of the computational domain. The key of affine parametrization is that the Jacobian of the map,

$$J^{\ell}_{\mathrm{aff}}(\mathbf{p}) = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} & \frac{\partial x}{\partial \hat{z}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} & \frac{\partial y}{\partial \hat{z}} \\ \frac{\partial z}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{y}} & \frac{\partial z}{\partial \hat{z}} \end{bmatrix}, \tag{26}$$

is constant on each subdomain $V^{\ell}_{D}(\mathbf{p})$. In the integrations in (19)–(22), the volume integrations now have to be carried out according to $\mathrm{d}V = \vartheta^{\ell}_{0}(\mathbf{p})\,\mathrm{d}\hat{V}$, where $\vartheta^{\ell}_{0}(\mathbf{p}) = |J^{\ell}_{\mathrm{aff}}(\mathbf{p})|$ denotes the determinant of the Jacobian. Hence,

$$\mathbf{M}_{\sigma}(\mathbf{p}) = \vartheta^{\ell}_{0}(\mathbf{p})\hat{\mathbf{M}}_{\sigma}; \tag{27}$$

$$\mathbf{j}_{\mathrm{src}}(\mathbf{p}) = \vartheta^{\ell}_{0}(\mathbf{p})\hat{\mathbf{j}}_{\mathrm{src}}, \tag{28}$$

where $\hat{\mathbf{M}}_{\sigma}$ and $\hat{\mathbf{j}}_{\mathrm{src}}$ are assembled for the reference geometry only once. Additionally, the affine maps affect the differential operators in (19) and (22). A bit of calculation is needed to work out the transformed differential operators and the scalar products component-wise. For the 2D Cartesian case, the results are

$$\mathbf{K}^{\ell}_{\nu}(\mathbf{p}) = \vartheta^{\ell}_{1}(\mathbf{p})\hat{\mathbf{K}}^{\ell}_{\nu,xx} + \vartheta^{\ell}_{2}(\mathbf{p})\hat{\mathbf{K}}^{\ell}_{\nu,yy} + \vartheta^{\ell}_{3}(\mathbf{p})\hat{\mathbf{K}}^{\ell}_{\nu,xy} + \vartheta^{\ell}_{4}(\mathbf{p})\hat{\mathbf{K}}^{\ell}_{\nu,yx}; \tag{29}$$

$$\mathbf{j}^{\ell}_{\mathrm{m}}(\mathbf{p}) = \vartheta^{\ell}_{5}(\mathbf{p})\hat{\mathbf{j}}^{\ell}_{\mathrm{m},x} + \vartheta^{\ell}_{6}(\mathbf{p})\hat{\mathbf{j}}^{\ell}_{\mathrm{m},y}, \tag{30}$$

where the matrix factors $\hat{\mathbf{K}}^{\ell}_{\nu,xx}$, $\hat{\mathbf{K}}^{\ell}_{\nu,yy}$, $\hat{\mathbf{K}}^{\ell}_{\nu,xy}$, $\hat{\mathbf{K}}^{\ell}_{\nu,yx}$, and the vector factors $\hat{\mathbf{j}}^{\ell}_{\mathrm{m},x}$ and $\hat{\mathbf{j}}_{\mathrm{m},y}$ are assembled for the reference geometry in advance. Hence, the assembly of new FE matrices and vectors can be avoided during the optimization procedure. The functions $\vartheta^{\ell}_{q}(\mathbf{p})$ are simple scalar functions in terms of the design parameters and are evaluated for each model instantiation.

3.3.2 Design element approach

For many geometry optimization tasks, the domain underlying geometry changes cannot be decomposed in triangles or tetrahedra with straight edges and faces, which excludes the use of affine parametrization. NURBS are a more general way to represent geometries and are widely used in CAD systems. Therefore, it seems natural to use the control points (and weights) of NURBS curves as design parameters [6,49]. This approach has received considerable attention in recent years as new approaches, incorporating NURBS geometries into FE analysis, have emerged. Isogeometric analysis [25] and the NURBS-enhanced FE method [50] are important examples. Here, NURBS are only used for the geometry parametrization. A triangular (tetrahedral) mesh is generated once and deformed using the concept of design elements [6,26].

In the following, for simplicity, the two-dimensional case is considered solely. A generic NURBS curve of degree p is given as

$$\mathbf{C}(\hat{x}) = \sum_{i} R^{p}_{i}(\hat{x})\mathbf{P}_{i}, \tag{31}$$

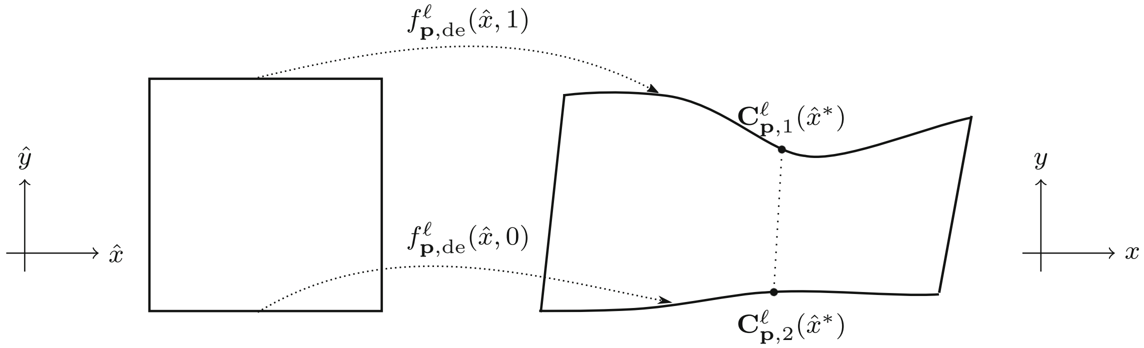
$f^{\ell}_{\mathbf{p},\mathrm{de}}(\hat{x},1)$
$f^{\ell}_{\mathbf{p},\mathrm{de}}(\hat{x},0)$
$\mathbf{C}^{\ell}_{\mathbf{p},1}(\hat{x}^*)$
$\mathbf{C}^{\ell}_{\mathbf{p},2}(\hat{x}^*)$
$\hat{y}$
$\hat{x}$
y
x

Fig. 2 Reference domain $[0, 1]^2$ and design element patch

where $\mathbf{P}_i$ refers to a control point and the rational spline R_i^p is defined in terms of B-splines N_i^p and weights w_i as

$$R_i^p(\hat{x}) = \frac{N_i^p(\hat{x})w_i}{\sum_j N_j^p(\hat{x})w_j}. \tag{32}$$

In total, L design elements are considered, each of which is represented by two NURBS curves $\mathbf{C}^{\ell}_{\mathbf{p},1}$ and $\mathbf{C}^{\ell}_{\mathbf{p},2}$, each depending on the geometric parameters $\mathbf{p}$. More precisely, a design element is defined by a map $f^{\ell}_{\mathbf{p},\mathrm{de}} : \hat{V}^{\ell}_D = [0,1]^2 \to V^{\ell}_D(\mathbf{p})$ given as

$$f^{\ell}_{\mathbf{p},\mathrm{de}}(\hat{x},\hat{y}) = \mathbf{C}^{\ell}_{\mathbf{p},1}(\hat{x})\hat{y} + \mathbf{C}^{\ell}_{\mathbf{p},2}(\hat{x})(1-\hat{y}). \tag{33}$$

Hence, design elements are obtained by successively connecting points of each NURBS curve by a straight line, as depicted in Fig. 2. The affine parametrization may result in unstructured representations where such a patch structure is missing. For each node (x_i, y_i) in $V^{\ell}_D(\mathbf{p})$, its position in the reference domain $[0,1]^2$ is computed in advance by solving

$$(\hat{x}_i, \hat{y}_i) \in \mathrm{argmin}_{(\hat{x},\hat{y})} \left| f^{\ell}_{\mathbf{p},\mathrm{de}}(\hat{x},\hat{y}) - (x_i, y_i) \right|, \tag{34}$$

for example with the Newton–Raphson method. Then, the mesh can be easily deformed by applying the parameter-dependent map $f^{\ell}_{\mathbf{p},\mathrm{de}}$ to all nodes $(\hat{x}_i, \hat{y}_i)$.

The transformation of the FE matrices and vectors is more involved compared to the affine parametrization described in Sect. 3.3.1. Each entry of the mass matrix is transformed as

$$\mathbf{M}_{\sigma,i,j}(\mathbf{p}) = \int_{\hat{V}_D} \hat{\sigma}\hat{\mathbf{w}}_j \cdot \hat{\mathbf{w}}_i |J^{\ell}_{\mathrm{de}}(\mathbf{p})| \,\mathrm{d}\hat{V}, \tag{35}$$

where it is important to emphasize that $|J^{\ell}_{\mathrm{de}}(\mathbf{p})|$ is not constant on each design element. A similar expression is obtained for $\mathbf{j}_{\mathrm{src}}(\mathbf{p})$, whereas the conforming transformation of the curl operator yields

$$\mathbf{K}_{\nu,i,j}(\mathbf{p}) = \int_{\hat{V}_D} \frac{\hat{\nu}}{|J^{\ell}_{\mathrm{de}}(\mathbf{p})|} J^{\ell}_{\mathrm{de}}(\mathbf{p})\nabla \times \hat{\mathbf{w}}_j \cdot J^{\ell}_{\mathrm{de}}(\mathbf{p})\nabla \times \hat{\mathbf{w}}_i \,\mathrm{d}\hat{V}, \tag{36}$$

$$\mathbf{j}^{\ell}_{\mathrm{m},i}(\mathbf{p}) = \int_{\hat{V}_D} \hat{\mathbf{H}}_{\mathrm{m}}(\mathbf{p}) \cdot J^{\ell}_{\mathrm{de}}(\mathbf{p})\nabla \times \hat{\mathbf{w}}_i \,\mathrm{d}\hat{V}. \tag{37}$$

In (36) and (37), the dependence of the integration domain on the geometry changes was eliminated. Because the Jacobian $J^{\ell}_{\mathrm{de}}(\mathbf{p})$ can be expressed as a function of the geometric parameters p_i, the analytical derivative of the system matrix and of the right-hand side with respect to the geometric parameters can be determined.

3.4 Sensitivities

After differentiating the FE system, a new linear system for the derivatives of the degrees of freedom with respect to the geometric parameters is obtained:

$$\mathbf{K}_{\nu}\mathbf{s}_i = \frac{\partial}{\partial p_i}(\mathbf{j}_{\mathrm{src}} + \mathbf{j}_{\mathrm{m}}) - \frac{\partial \mathbf{K}_{\nu}}{\partial p_i}\mathbf{a}, \quad \text{for} \quad i = 1, \ldots, I, \tag{38}$$

where $\mathbf{s}_i(\mathbf{p}) = \frac{\partial \mathbf{a}(\mathbf{p})}{\partial p_i}$ are the sensitivities of the FE solution. To calculate $\mathbf{s}_i$, I equations of the form (38) have to be solved. In the case of affine parametrization, derivatives of $\mathbf{K}_{\nu}$ are easily calculated from (23) and (29) using expressions for $\frac{\partial \vartheta^{\ell}(\mathbf{p})}{\partial p_i}$ which are known analytically as derivatives of the functions $\vartheta^{\ell}(\mathbf{p})$. The expressions become more involved when NURBS are involved, yet closed-form formulas also exist in this case.

The optimization algorithm requires the derivatives of the objective function with respect to each of the design parameters. Often, the objective function does not explicitly depend on the design parameters, i.e., $J(\mathbf{p}) = \tilde{J}(\mathbf{a}(\mathbf{p}))$. In this case, the derivatives are given as

$$\frac{\partial J(\mathbf{p})}{\partial p_i} = \nabla_{\mathbf{a}}\tilde{J}(\mathbf{a}(\mathbf{p})) \cdot \mathbf{s}_i(\mathbf{p}), \quad \text{for} \quad i = 1, \ldots, I. \tag{39}$$

For a large number of parameters, an adjoint method should be used instead [57].

4 Example 1: Die press mold

As a first example, a die press mold for radially magnetizing a segment of sintered magnetic powder (SMP) is considered [54]. This problem has been proposed as testing electromagnetic analysis methods (TEAM) benchmark problem 25 [53] and has been used in numerous papers for comparing optimization algorithms. The vast majority of these publications apply and compare *stochastic* optimization methods [32,52], possibly combined with surrogate models [8], uncertainty quantification [39], multi-objective optimization or a combination of them [31]. Only a few papers (see, e.g., [1,4]) choose deterministic methods, again possibly combined with surrogate models [20], uncertainty quantification [55] or multi-objective optimization. This paper addresses one of the main drawbacks of deterministic methods, which is the consideration of geometric parameters. For this example, the design element approach is used.

The SMP segment is arranged between a cylindrical inner pole and a more generally shaped outer pole (Fig. 3). The original TEAM-25 problem considers an outer pole with an elliptical inner surface. Here, the inner surface is described by a spline. This is motivated by the fact that splines are currently the basic building block for mechanical processing. The considered design parameters are then chosen to be

p_1: radius of the inner yoke;

p_2, p_3: semiaxis of ellipse between points i and j;

p_4: x-coordinate of points m and k.

Both the circle and the ellipse are exactly represented by NURBS curves. The relation between the geometric parameters and the NURBS control points is given in the Appendix.

The optimization aims at a homogeneous, radially oriented magnetic flux density of $B_{\text{goal}} = 0.35$ T inside the SMP segment. The objective function $J(\mathbf{p})$ is defined as the mean-squared error between the simulated magnetic field and the goal at 9 sample points equidistantly distributed along the arc with radius r_{smp} between points $e = (r_{\text{smp}}, 0)$ and $f = (r_{\text{smp}} \cos \varphi_f, r_{\text{smp}} \sin \varphi_f)$:

$$J(\mathbf{p}) = \sum_{k=1}^{9} \|\mathbf{B}(r_{\text{smp}} \cos \varphi_k, r_{\text{smp}} \sin \varphi_k; \mathbf{p}) - B_{\text{goal}} \mathbf{e}_k\|_2^2, \tag{40}$$

where $\varphi_k = \varphi_f \frac{k-1}{8}$ and $\mathbf{e}_k = (\cos \varphi_k, \sin \varphi_k)$. The optimization problem yields:

$$\underset{\mathbf{p}}{\text{minimize}} \quad J(\mathbf{p}), \tag{41a}$$

$$\text{subject to} \quad \mathbf{p} \in \mathcal{F}, \tag{41b}$$

where the admissible set is defined as:

$$\mathcal{F} = [5.1, 9] \times [16, 18] \times [14.5, 16] \times [9.5, 13] \text{ mm}.$$

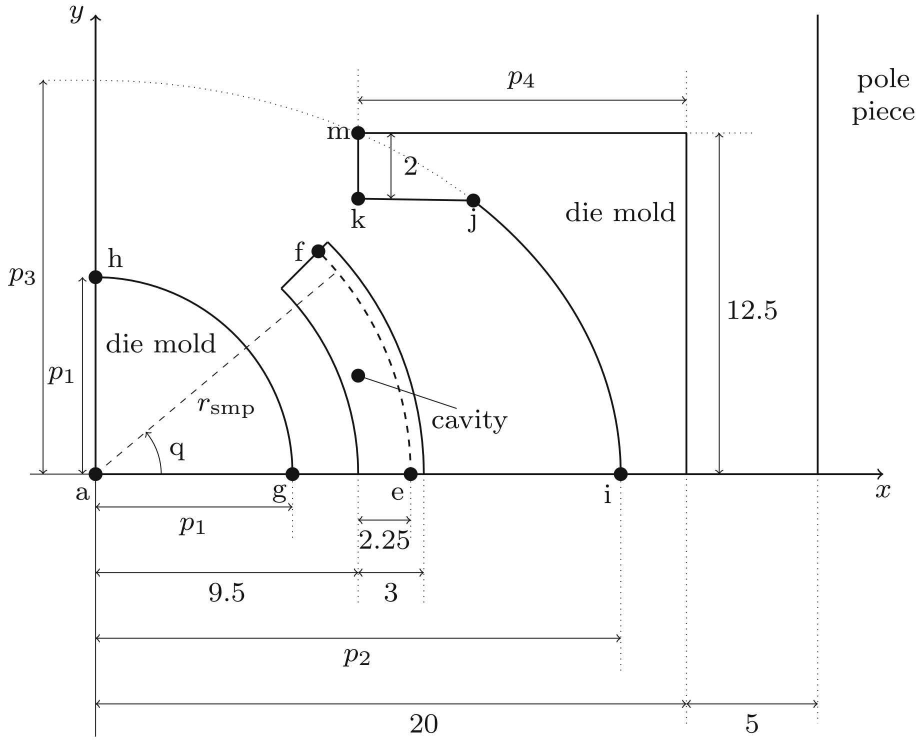

Fig. 3 TEAM Problem 25: Cross section of the inner part of the die press showing the SMP ring, the inner yoke and the outer yoke (all measures in mm). A horizontal magnetic flux is exerted on the configuration by an outer magnetic circuit (not shown)

Table 1 Results from the optimization of the die press mold with particle swarm optimization (PSO), trust region (TR) (with MATLAB®'s `fmincon`) and an own implementation of sequential quadratic programming (SQP) combined with the design element approach

Method	Minimizer $\hat{\mathbf{p}}_{\mathrm{min}}$ (in mm)	Minimum (in T^2)	Iteration count	Function calls		Time (in s)
				f()	∇f()	
PSO	$\begin{pmatrix}5.1000\\16.0000\\16.0000\\9.5000\end{pmatrix}$	1.413498	7	280	N/A	56.63
SQP (`fmincon`)	$\begin{pmatrix}5.1000\\16.0000\\16.0000\\9.5000\end{pmatrix}$	1.413498	4	7	7	31.61
SQP (own implementation)	$\begin{pmatrix}5.1000\\16.0000\\16.0000\\9.4999\end{pmatrix}$	1.413498	2	3	2	12.84

For the gradient, the derivatives of J with respect to the geometry parameters p_i are needed. Before applying the chain rule on (40), the derivatives of the degrees of freedom with respect to the geometry parameters $\partial_{p_i}\mathbf{a}$ are calculated as described in Sect. 3.3.2.

The performance of a standard algorithm for particle swarm optimization (PSO), of the sequential quadratic programming (SQP) method implemented in MATLAB®'s `fmincon` function [43] and of an own implementation of SQP is compared in Table 1. Both SQP implementations use the analytical gradients, the BFGS formula for updating the Hessian and a sufficient decrease condition in a merit function. For the PSO, a set of 40 particles is considered and the implementation is multi-threaded, while the gradient-based methods are single-thread implementations. The termination criterion for the PSO algorithm is the number of stall iterations, which was set to 5. The PSO actually finds the optimum after 2 iterations. This is because the optimum is at a vertex of the box-shaped domain and all the particles leaving the admissible region are projected onto the boundary. All three methods converge to the same optimum. The deterministic algorithms are substantially faster than PSO, even though PSO exploits parallelization. On the same machine, an evaluation of the objective function $J(\mathbf{p})$ is performed in 1.65 s, an analytical evaluation of the gradient $\nabla J(\mathbf{p})$ in 4.69 s and a numerical evaluation of the gradient $\nabla_{\mathrm{num}} J(\mathbf{p})$ using a forward difference quotient in 7.48 s. All tests were done on a 64 GB RAM Intel® Xeon® E5-2630 v4 machine.

5 Example 2: Permanent-magnet synchronous machine (PMSM)

5.1 Design parameters

The second example is a 3-phase 6-pole permanent-magnet (PM) synchronous machine (PMSM) borrowed from [42] (Fig. 4) and already studied as an optimization example in [5]. The stator features two slots per pole and per phase with a conventional distributed double-layer winding. The rotor contains a buried rare-earth magnet. The yoke parts are laminated. The design parameters are

p_1: width of the PM;
p_2: thickness of the PM;
p_3: distance from the PM to the rotor surface.

5.2 Objective function

The optimization goal is to minimize the size $S_{\mathrm{pm}} = p_1 p_2$ of PM material while preserving a prescribed electromotive force E_0. The electromotive force (EMF) $E_0(\mathbf{p})$ is post-processed from a magnetostatic solution of a 2D FE model of the PMSM using the *loading* method proposed in [46]. For that purpose, the FE solution of the z-component of the magnetic vector potential is sampled at a circle (or in the case of a partial machine model, an arc) in the PMSM's air gap, yielding $A_z(r_{\mathrm{ag}}, \varphi) \approx \hat{A}_{z,\mathrm{eff}}\sqrt{2}\sin(N_p\varphi - \varphi_{\mathrm{d}})$, where $N_p = 3$ is the pole-pair number, $\hat{A}_{z,\mathrm{eff}}$ is the rms magnitude of fundamental harmonic component and φ_{d} is the angle of the PMSM's *direct* axis. The EMF is then found from

$$E_0 = 2\hat{A}_{z,\mathrm{eff}}\,\omega_{\mathrm{syn}} N_{\mathrm{w}} k_{\mathrm{w},1}, \tag{42}$$

where ω_{syn} is the synchronous speed and N_{w} is the number of windings per phase. The winding factor is

$$k_{\mathrm{w},\nu} = \frac{\sin\left(q\nu\frac{\alpha_{\mathrm{el}}}{2}\right)}{q\sin\left(\nu\frac{\alpha_{\mathrm{el}}}{2}\right)} \cdot \sin\left(\nu\frac{\pi}{2}\frac{\tau_{\mathrm{c}}}{\tau_{\mathrm{p}}}\right) \cdot \frac{\sin\left(\nu\frac{\varepsilon}{2}\right)}{\nu\frac{\varepsilon}{2}}, \tag{43}$$

where q is the number of coil sides per phase belt, α_{el} is the electric angle between two slots, τ_{c} is the coil pitch, τ_{p} is the pole pitch and ε is the electric skew angle [37,46].

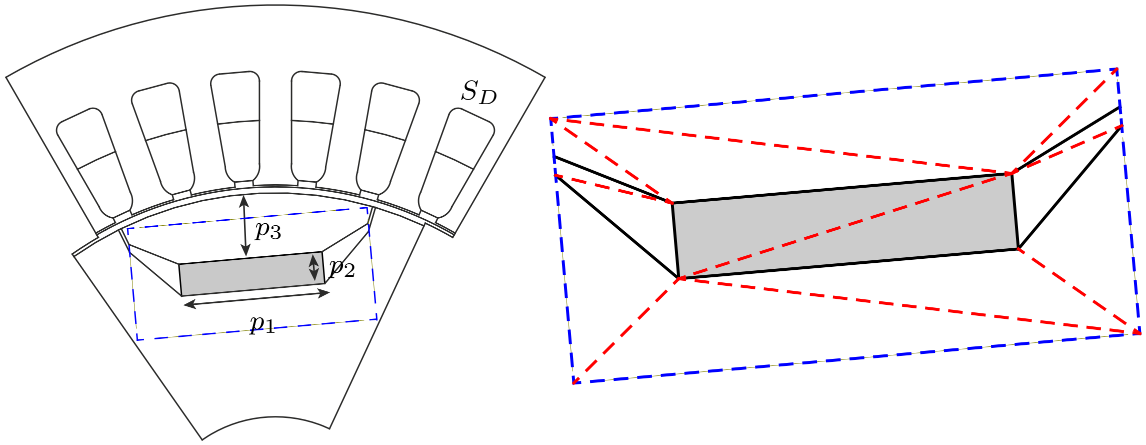

Fig. 4 Cross section of one pole of the machine with the permanent magnet depicted in gray and the region of the affine parametrization indicated by the dashed box. On the right-hand side, the triangulation into L subdomains is shown by the dashed-dotted lines. The figure is adapted from [5]

5.3 Optimization problem

The optimization problem reads

$$\underset{\mathbf{p}\in\mathbb{R}^3}{\text{minimize}} \quad J(\mathbf{p}) = p_1 p_2, \tag{44a}$$

$$\text{subject to} \quad G(\mathbf{p}) = \begin{bmatrix} p_1^{\mathrm{l}} - p_1 \\ p_2^{\mathrm{l}} - p_2 \\ p_3^{\mathrm{l}} - p_3 \\ p_3 - p_3^{\mathrm{u}} \\ p_2 + p_3 - 15\text{ mm} \\ 3p_1 - 2p_3 - 50\text{ mm} \\ E_{\mathrm{d}} - E_0(\mathbf{p}, \mathbf{a}(\mathbf{p})) \end{bmatrix} \leq 0. \tag{44b}$$

The first four constraints are related to the lower (p^{l}) and upper (p^{u}) bounds of $\mathbf{p}$: $\left(p_1^{\mathrm{l}}, p_2^{\mathrm{l}}, p_3^{\mathrm{l}}\right) = (1, 1, 5)$ mm and $\left(p_1^{\mathrm{u}}, p_2^{\mathrm{u}}, p_3^{\mathrm{u}}\right) = (\infty, \infty, 14)$ mm. To ensure the validity of the affine parametrization (intersections are not allowed), the fifth constraint is added. The sixth constraint is a design constraint enforcing that each PM has to keep a sufficient distance to the rotor surface, especially for wide PMs. The last constraint expresses the requirement to fulfill the prescribed EMF. Since the EMF is post-processed from the FE solution, the optimization problem actually has a PDE constraint.

5.4 Results

The results for 5 different optimization methods are collected in Table 2.

1. The first optimization run is carried out with the genetic algorithm implemented in MATLAB®.
2. The second optimization run is carried out with MATLAB®'s PSO implementation. To circumvent the restriction to box-shaped parameter domains, the admissible set is enforced by a penalty turn. The new objective function reads
$$\begin{aligned} J_{\mathrm{pen}}(\mathbf{p}) = J(\mathbf{p}) + 2J(\mathbf{p})\big(&f\,(\max(p_2 + p_3 - 15, 0)) \\ &+ f\,(\max(3p_1 - 2p_3 - 50, 0)) \\ &+ f\,(\max(g(x), 0))\big), \end{aligned} \tag{45}$$
where $f(t) = e^{(4t^{0.1})} - 1$ was chosen heuristically such that J_{pen} grows exponentially if one of the constraints is violated. The function J_{pen} was called 4740 times, but was organized as to only evaluate the nonlinear constraint if all other constraints were satisfied. The number of particles was set to 30, the maximum number of stall iterations to $N_{\mathrm{stall}} = 15$ and the function change tolerance to 10^{-6}. The PSO characteristic constants are chosen to be $\omega_0 = 0.5$ and $\omega_1 = \omega_2 = 1.49$. The algorithm took 157 iterations before termination.
3. The third optimization is carried out with an own PSO implementation, for the original objective function $J(\mathbf{p})$ and applying the nonlinear constraints directly. Here, it is assumed that the admissible set is convex such that points inside the convex hull formed by all previous points do not need to be checked. Fifty particles were used. Termination was enforced after maximally $N_{\mathrm{it,max}} = 100$ steps or when $N_{\mathrm{stall,max}} = 15$ stall iterations were observed.
4. The fourth run was done with the deterministic method described in Sect. 2.3, relying upon FE simulations equipped with an affine parametrization of the geometry as described in Sect. 3.3.1.
5. The fifth run was done with the deterministic method for robust optimization expressed by (11) in Sect. 2.5, again with affine parametrization of the geometry.

The three stochastic algorithms were run on a 64 GB RAM Intel® Xeon® E5-2630 v4 machine. Both deterministic algo-

Table 2 Numerical results obtained for a $\delta = 0.2$ mm [5]

	p_1 (mm)	p_2 (mm)	p_3 (mm)	S_{pm} (mm^2)	E_0 (V)	FE slv calls	Time (s)
Initial design	19.00	7.00	7.00	133	30.370	–	–
Genetic algorithm	21.04	2.98	6.56	62.80	30.370	≈ 6760	520.5
PSO with penalty term	20.60	3.09	5.91	63.71	30.370	≈ 3470	267.16
PSO, own implementation	21.08	2.98	6.63	62.80	30.370	1765	217.52
SQP, nominal optimization	21.07	2.98	6.61	62.80	30.370	34	2.0
SQP, robust optimization	20.88	3.73	6.82	77.87	31.086	48	5.9

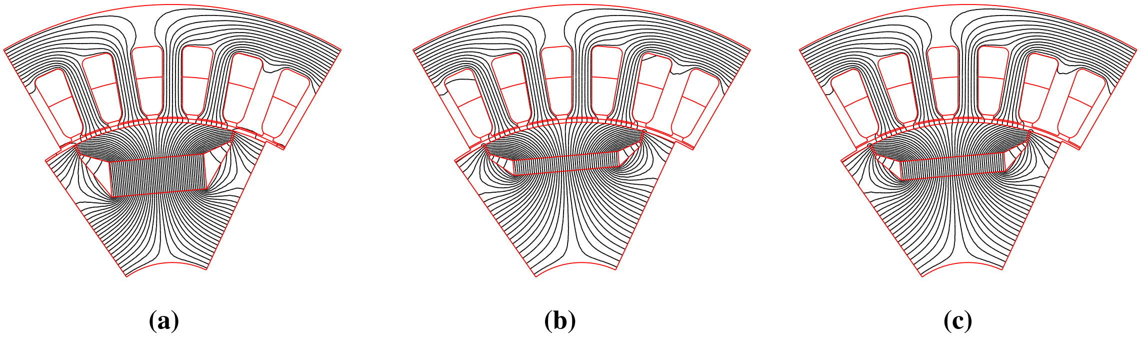

Fig. 5 Initial and optimized geometries together with the magnetic flux distribution at no-load. The figures are adapted from [5]. **a** Initial geometry, **b** nominal optimum, **c** robust optimum

rithms were run on a 16 GB RAM Intel® Core$^{\mathrm{TM}}$ with i7-5820K processors (3.30 GHz).

The results of all optimization procedures are compared with the values of the initial design (Fig. 5). All routines achieve a substantial decrease in the PM size from 133 mm^2 up to about 63 mm^2. The price for robustness is a slightly larger size of about 77 mm^2. The deterministic methods outperform the stochastic ones by two orders of magnitude. This impressively illustrates the major message of this paper stating that deterministic optimization methods accompanied by FE analysis providing gradients with respect to geometric parameters should be favored over stochastic methods, at least for the here considered class of problems.

6 Conclusion

Affine parametrization and design element approaches are capable of parametrizing the geometry of finite-element models such that accurate derivatives with respect to geometric parameters become available. This alleviates one of the major drawbacks of gradient-type deterministic optimization methods. For the example of a die mold press, standard sequential programming combined with the design element approach outperforms particle swarm optimization by more than a factor ten. The second example illustrates the applicability of gradient-type robust optimization combined with an affine parametrization of the geometry for a permanent-magnet synchronous machine. Supported by the substantial improvement in computational efficiency, this paper stands up for a revival of deterministic methods for numerical optimization in electrotechnical design procedures.

Acknowledgements This work is supported by the German BMBF in the context of the SIMUROM project (grant nr. 05M2013) and the PASIROM project (grant nr. 05M2018), by the 'Excellence Initiative' of the German Federal and State Governments and by the Centre and Graduate School Computational Engineering at TU Darmstadt.

Appendix

The dependence between the geometry parameters and the NURBS representation of the die press model is as follows. For the ellipse arc, the control points and weights are

$$\mathbf{P_0} = \begin{pmatrix} L_2 \\ 0 \end{pmatrix}, \quad \mathbf{P_2} = \begin{pmatrix} L_2 \cos\alpha \\ L_3 \sin\alpha \end{pmatrix},$$

$$\mathbf{P_1} = \mathbf{P_2} + \begin{pmatrix} -\lambda L_2 \sin\alpha \\ \lambda L_3 \cos\alpha \end{pmatrix},$$

$$w_0 = w_2 = 1, \quad w_1 = \cos\frac{\alpha}{2},$$

where

$$\alpha = \operatorname{asin}\left(\frac{10.5\text{mm}}{p_3}\right), \quad \lambda = \frac{-p_2 + p_2 \cos\alpha}{p_2 \sin\alpha}.$$

The corresponding knots are $\mathcal{K} = \{0, 0, 0, 1, 1, 1\}$, and the degree of the basis functions is $p = 2$.

For the circular arc, the control points are

$$\mathbf{P_0} = \begin{pmatrix} p_1 \\ 0 \end{pmatrix}, \quad \mathbf{P_1} = \begin{pmatrix} p_1 \\ p_1 \end{pmatrix}, \quad \mathbf{P_2} = \begin{pmatrix} 0 \\ p_1 \end{pmatrix},$$

with the constant weights

$$w_0 = w_2 = 1, \quad w_1 = \sqrt{2}/2.$$

The degree of the basis functions is $p = 2$. The corresponding knots are $\mathcal{K} = \{0, 0, 0, 1, 1, 1\}$. The deformation of the mesh inside one design element region V_D^ℓ with N_V vertices is given by

$$(x_i, y_i) = \mathbf{C}_1^\ell(\hat{x}_i; \mathbf{p})\hat{y}_i + \mathbf{C}_2^\ell(\hat{x}_i; \mathbf{p})(1 - \hat{y}_i), \tag{46}$$

where (x_i, y_i) are the coordinates of the vertices of the deformed mesh and $(\hat{x}_i, \hat{y}_i)$ are the coordinates in the reference domain $[0, 1]^2$.

References

1. Alotto P, Nervi M (2001) An efficient hybrid algorithm for the optimization of problems with several local minima. Int J Numer Meth Eng 50:847–868
2. Bash ML, Pekarek SD (2011) Modeling of salient-pole wound-rotor synchronous machines for population-based design. IEEE Trans Energy Convers 26(2):381–392
3. Baumgartner U, Magele C, Renhart W (2004) Pareto optimality and particle swarm optimization. IEEE Trans Magn 40(2):1172
4. Berkani M, Giurgea S, Espanet C, Coulomb JL, Kieffer C (2013) Study on optimal design based on direct coupling between a FEM simulation model and L-BFGS-B algorithm. IEEE Trans Magn 49(5):2149–2152
5. Bontinck Z, Lass O, Schöps S, De Gersem H, Ulbrich S, Rain O (2018) Robust optimization formulations for the design of an electric machine. IET Sci Meas Technol (accepted). arXiv:1712.01536
6. Braibant V, Fleury C (1984) Shape optimal design using B-splines. Comput Methods Appl Mech Eng 44(3):247–267. https://doi.org/10.1016/0045-7825(84)90132-4
7. Brisset S, Vizireanu D, Brochet P (2008) Design and optimization of a nine-phase axial-flux PM synchronous generator with concentrated winding for direct-drive wind turbine. IEEE Trans Ind Appl 44(3):707–715. https://doi.org/10.1109/TIA.2008.921379
8. Canova A, Gruosso G, Repetto M (2003) Magnetic design optimization and objective function approximation. IEEE Trans Magn 39(5):2154–2162
9. Cassimere BN, Sudhoff SD (2009) Population-based design of surface-mounted permanent-magnet synchronous machines. IEEE Trans Energy Convers 24(2):338–346
10. Clemens M (2005) Large systems of equations in a discrete electromagnetism: formulations and numerical algorithms. IEE Proc Sci Meas Technol 152(2):50–72. https://doi.org/10.1049/ip-smt:20050849
11. Delfour M, Zolésio JP (2011) Shapes and geometries: metrics, analysis, differential calculus, and optimization, 2 edn. In: Advances in design and control. SIAM, Philadelphia. https://doi.org/10.1137/1.9780898719826
12. Di Barba P (2010) Multiobjective shape design in electricity and magnetism, lecture notes in electrical engineering. Springer, Berlin
13. Diehl M, Bock HG, Kostina E (2006) An approximation technique for robust nonlinear optimization. Math Program 107(1–2):213–230. https://doi.org/10.1007/s10107-005-0685-1
14. Duan Y, Ionel DM (2013) A review of recent developments in electrical machine design optimization methods with a permanent-magnet synchronous motor benchmark study. IEEE Trans Ind Appl 49(3):1268–1275. https://doi.org/10.1109/TIA.2013.2252597
15. Echeverría D, Lahaye D, Encica L, Lomonova EA, Hemker PW, Vandenput AJA (2006) Manifold-mapping optimization applied to linear actuator design. IEEE Trans Magn 42(4):1183–1186. https://doi.org/10.1109/TMAG.2006.870969
16. Gangl P, Amstutz S, Langer U (2016) Topology optimization of electric motor using topological derivative for nonlinear magnetostatics. IEEE Trans Magn 52(3):1–4. https://doi.org/10.1109/TMAG.2015.2496172
17. Gillon F, Brochet P (2000) Screening and response surface method applied to the numerical optimization of electromagnetic devices. IEEE Trans Magn 36(4):1163–1167. https://doi.org/10.1109/20.877647
18. Graeb HE (2007) Analog design centering and sizing. Springer, Dordrecht
19. Hameyer K, Kasper M (1993) Shape optimization of a fractional horse-power DC-motor by stochastic methods. In: Hernandez S, Brebbia C (eds) Computer aided optimimum design of structures III: optimization of structural systems and applications, pp 15–30
20. Hemker PW, Echeverría D (2007) A trust-region strategy for manifold-mapping optimization. J Comput Phys 224:464–475
21. Hemker T, von Stryk O, De Gersem H, Weiland T (2008) Mixed-integer nonlinear design optimization of a superconductive magnet with surrogate functions. IEEE Trans Magn 44(6):1110–1113
22. Hinze M, Kunkel M, Vierling M (2011) POD model order reduction of drift-diffusion equations in electrical networks. In: Benner P, Hinze M, ter Maten EJW (eds) Model reduction for circuit simulation. Lecture notes in electrical engineering, vol 74. Springer, Netherlands, pp 177–192
23. Ho SL, Yang S, Ni Guangzheng LEWC, Wong HC (2005) A particle swarm optimization-based method for multiobjective design optimizations. IEEE Trans Magn 41(5):1756–1759. https://doi.org/10.1109/TMAG.2005.846033
24. Horst R, Pardalos PM, Thoai NV (2000) Introduction to global optimization. In: Nonconvex optimization and its applications, vol 48. Springer
25. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195. https://doi.org/10.1016/j.cma.2004.10.008
26. Imam MH (1982) Three-dimensional shape optimization. Int J Numer Methods Eng 18(5):661–673. https://doi.org/10.1002/nme.1620180504
27. Kennedy J, Eberhart R (1995) Particle swarm optimization. IEEE Int Conf Neural Netw 4:1942–1948
28. Koch S, De Gersem H, Weiland T, Fischer E, Moritz G (2008) Transient 3D finite element simulations of the SIS100 magnet considering anisotropic, nonlinear material models for the ferromagnetic yoke. IEEE Trans Appl Supercond 18(2):1601–1604

29. Koziel S, Leifsson L (2013) Surrogate-based modeling and optimization: applications in engineering. Springer, Berlin
30. Lass O, Ulbrich S (2017) Model order reduction techniques with a posteriori error control for nonlinear robust optimization governed by partial differential equations. SIAM J Sci Comput 39(5):S112–S139. https://doi.org/10.1137/16M108269X
31. Lebensztajn L, Rondini Marretto CA, Caldora Costa M, Coulomb JL (2004) Kriging: a useful tool for electromagnetic device optimization. IEEE Trans Magn 40(2):1196–1199
32. Lei G, Liu C, Zhu J, Guo Y (2015) Techniques for multilevel design optimization of permanent magnet motors. IEEE Trans Energy Convers 30(4):1574–1584
33. Lok CL, Vengadaesvaran B, Ramesh S (2017) Implementation of hybrid pattern search-genetic algorithm into optimizing axial-flux permanent magnet coreless generator (AFPMG). Electr Eng 99:751–761. https://doi.org/10.1007/s00202-016-0443-9
34. Ma C, Qu L (2015) Multiobjective optimization of switched reluctance motors based on design of experiments and particle swarm optimization. IEEE Trans Energy Convers 30(3):1144–1153
35. Man KF, Tang KS, Kwong S (2012) Genetic algorithms: concepts and designs. Springer, Berlin
36. Monk P (2003) Finite element methods for Maxwell's equations. Oxford University Press, Oxford
37. Müller G, Vogt K, Ponick B (1974) Berechnung elektrischer Maschinen, 6th edn. Wiley-VCH, Weinheim
38. Neri F, Tirronen V (2010) Recent advances in differential evolution: a survey and experimental analysis. Artif Intell Rev 33(1–2):61–106. https://doi.org/10.1007/s10462-009-9137-2
39. Nishida N, Takahashi Y, Wakao S (2008) Robust design optimization approach by combination of sensitivity analysis and sigma level estimation. IEEE Trans Magn 44(6):998–1001
40. Nocedal J, Wright SJ (2006) Numerical optimization. Springer series in operations research and financial engineering, 2nd edn. Springer, New York
41. Omekanda AM (2006) Robust torque and torque-per-inertia optimization of a switched reluctance motor using the Taguchi methods. IEEE Trans Ind Appl 42(2):473–478
42. Pahner U, Mertens R, De Gersem H, Belmans RJM, Hameyer K (1998) A parametric finite element environment tuned for numerical optimization. IEEE Trans Magn 34(5):2936–2939. https://doi.org/10.1109/20.717685
43. Powell MJD (1978) A fast algorithm for nonlinearly constrained optimization calculations, vol 630. Springer, Berlin, pp 144–157
44. Powell MJD (2002) UOBYQA: unconstrained optimization by quadratic approximation. Math Program B92:555–582. https://doi.org/10.1007/s101070100290
45. Powell MJD (2009) The BOBYQA algorithm for bound constrained optimization without derivatives. In: Technical report DAMTP 2009/NA06. Cambridge University Press
46. Rahman MA, Zhou P (1991) Determination of saturated parameters of PM motors using loading magnetic fields. IEEE Trans Magn 27(5):3947–3950. https://doi.org/10.1109/20.104967
47. Rozza G, Huynh DBP, Patera AT (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Arch Comput Methods Eng 15(3):229–275. https://doi.org/10.1007/s11831-008-9019-9
48. Russenschuck S (1990) Mathematical optimization techniques for the design of permanent magnet synchronous machines based on numerical field calculation. IEEE Trans Magn 26(2):638–641
49. Ryu JS, Yao Y, Koh CS, Yun S, Kim DS (2005) Optimal shape design of 3-D nonlinear electromagnetic devices using parameterized design sensitivity analysis. IEEE Trans Magn 41(5):1792–1795. https://doi.org/10.1109/TMAG.2005.845982
50. Sevilla R, Fernández-Méndez S, Huerta A (2008) NURBS-enhanced finite element method (nefem). Ph.D. Thesis, Universitat Politècnica de Catalunya (2008)
51. Sizov GY, Zhang P, Ionel DM, Demerdash NAO, Rosu M (2013) Automated multi-objective design optimization of PM AC machines using computationally efficient FEA and differential evolution. IEEE Trans Ind Appl 49(5):2086–2096
52. Sonoda S, Takahashi N, Kawagishi K, Nishida N, Wakao S (2007) Application of stepwise multiple regression to design optimization of electric machine. IEEE Trans Magn 43(4):1609–1612
53. Takahashi N (1997) Optimization of die press model (TEAM workshop problem 25). Technical report, International Compumag Society. http://www.compumag.org/jsite/images/stories/TEAM/problem25.pdf. Accessed 11 Mar 2018
54. Takahashi N, Ebihara K, Yoshida K, Nakata T, Ohashi K, Miyata K (1996) Investigation of simulated annealing method and its application to optimal design of die mold for orientation of magnetic powder. IEEE Trans Magn 32(3):1210–1213. https://doi.org/10.1109/20.497461
55. Takahashi RHC, Ramírez JA, Vasconcelos JA, Saldanha RR (2001) Sensitivity analysis for optimization problems solved by stochastic methods. IEEE Trans Magn 37(5):3566–3569. https://doi.org/10.1109/20.952663
56. Takorabet N, Laporte B, Vinsard G (1997) On the optimization of linear induction devices. Electr Eng 80:221–226
57. Tröltzsch F (2009) Optimale Steuerung Partieller Differentialgleichungen: Theorie. Springer, Verfahren und Anwendungen
58. Weeber K, Hoole SRH (1992) Geometric parametrization and constrained optimization techniques in the design of salient pole synchronous machines. IEEE Trans Magn 28(4):1948–1960
59. Wrobel R, Lukaniszyn M, Jagiela M, Latawiec K (2003) A new approach to reduction of the cogging torque in a brushless motor by skewing optimization of permanent magnets. Electr Eng 85:59–69. https://doi.org/10.1007/s00202-002-0144-4
60. Yoon SB, Jung IS, Hyun DS, Hong JP, Kim YJ (1999) Robust shape optimization of electromechanical devices. IEEE Trans Magn 35(3):1710–1713. https://doi.org/10.1109/20.767356

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