Slawomir Koziel · Leifur Leifsson

Simulation-Driven Design by Knowledge-Based Response Correction Techniques

Springer

Simulation-Driven Design by Knowledge-Based Response Correction Techniques

Slawomir Koziel • Leifur Leifsson

Simulation-Driven Design by Knowledge-Based Response Correction Techniques

Springer

Slawomir Koziel
Engineering Optimization
& Modeling Center
Reykjavik University
Reykjavik, Iceland

Leifur Leifsson
Department of Aerospace Engineering
Iowa State University
Ames, Iowa
USA

ISBN 978-3-319-30113-6 ISBN 978-3-319-30115-0 (eBook)
DOI 10.1007/978-3-319-30115-0

Library of Congress Control Number: 2016937931

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

Computer simulation models have become the most important design tools in the majority of engineering disciplines. They are used for verification purposes but also directly in the design process, in particular, for design optimization. Although desirable, and in many situations indispensable, simulation-driven design may be quite challenging. A fundamental bottleneck is the high computational cost of the simulation models. This often makes the use of conventional optimization algorithms prohibitive as these techniques require—especially for higher-dimensional parameter spaces—a large number of objective function evaluations. The high computational cost issue can be alleviated to some extent by utilization of adjoint sensitivities or automated differentiation. In general, however, robust, automated, and efficient optimization of expensive computational models remains an open problem.

Probably the most promising approach to computationally efficient handling of expensive computational models is surrogate-based optimization (SBO) where direct optimization of the expensive (high-fidelity) simulation model is replaced by iterative construction, enhancement, and re-optimization of its cheap representation, referred to as a surrogate model. A number of SBO techniques have evolved over the recent years. The most important difference between them lies in the surrogate model construction. The two major types of surrogate models can be distinguished: data-driven models obtained by approximating sampled high-fidelity simulation data, and physics-based ones constructed by suitable correction of the underlying low-fidelity model. Because the low-fidelity models embed system-specific knowledge, usually a small number of high-fidelity simulations are sufficient to configure a reliable surrogate. The most straightforward way of constructing the surrogate model from the underlying low-fidelity model is to correct its response. Various response correction techniques and related SBO algorithms have been proposed in the literature. Some of these methods are based on simple analytical formulas; others aim at tracking the response changes of the low-fidelity model, thus ensuring better generalization capability, although at the cost of increased formulation complexity and more restrictive assumptions regarding applicability of a given method.

This book focuses on surrogate-based optimization techniques exploiting response-corrected physics-based surrogates. For the sake of keeping the text self-contained, relevant background material is also covered to some extent, including the basics of classical and surrogate-assisted optimization, both based on data-driven and physics-based surrogates. The main part of the book provides an exposition of parametric and non-parametric response correction techniques as well as discusses practical issues on the construction and trade-offs for low-fidelity models. Furthermore, we consider related methods and discuss various aspects of physics-based SBO such as the adaptively adjusted design specification approach, feature-based optimization, response correction methods enhanced by adjoint sensitivities, as well as multi-objective optimization. Finally, we outline applications of response correction techniques for physics-based surrogate modeling. The book contains a large number of practical design cases from various engineering disciplines including electrical engineering, antenna design, aerospace engineering, and hydrodynamics, as well as recommendations for practical use of the presented methods.

Reykjavik, Iceland Slawomir Koziel
Ames, IA, USA Leifur Leifsson

Contents

Chapter 1
Introduction

Computer simulations have become the principal design tool in many engineering disciplines. They are commonly used for verification purposes, but also, more and more often, directly utilized within the design process, e.g., to adjust geometry and/or material parameters of the system of interest so that given performance requirements are satisfied. As a matter of fact, simulation-driven design has become a necessity for a growing number of devices and systems, where traditional approaches (e.g., based on design-ready theoretical models) are no longer adequate. One of the reasons is the increasing level of complexity of engineering systems, as well as various system- and component-level interactions, which have to be taken into account in the design process (Koziel and Ogurtsov 2014a). A reliable evaluation of the system performance can only be obtained (apart from physical measurements of the fabricated prototype) through high-fidelity computer simulations (typically, these simulations are computationally expensive).

Perhaps the most common design task involving multiple evaluations of the computational model is parametric optimization, in which a sequence of candidate designs are produced so as to converge to a solution that is optimal with respect to a predefined objective function (Sobester and Forrester 2015; Koziel et al. 2013a). Other tasks involving numerous simulations include statistical design (e.g., Monte Carlo analysis; Styblinski and Oplaski 1986), construction of behavioral models (Couckuyt et al. 2012), or uncertainty quantification (Hosder 2012; Allaire and Willcox 2014). A wide range of specialized algorithms have been developed to perform each of these tasks (Nocedal and Wright 2000; Yang 2010; Conn et al. 2009; Gorissen et al. 2010). Some of these methods are generic, i.e., applicable to a number of problems in different areas (Nocedal and Wright 2000), and others are more problem specific (e.g., Koziel et al. 2013a).

Unfortunately, in many practical cases, simulation-driven design is a very challenging problem. A fundamental bottleneck is the high computational cost of the simulation models utilized in many areas of engineering and science. Depending on the complexity of the computational model, simulation times may vary from seconds (e.g., for simple electromagnetic (EM) simulations of

S. Koziel, L. Leifsson, *Simulation-Driven Design by Knowledge-Based Response Correction Techniques*, DOI 10.1007/978-3-319-30115-0_1

two-dimensional models, e.g., planar microwave filters), through minutes (e.g., computational fluid dynamics (CFD) simulations of two-dimensional airfoil profiles), to hours and days (CFD simulations of three-dimensional structures such as aircraft wings, or EM simulations of integrated photonic components), or even weeks (simulations of complex 3D structures such as a full aircraft or a ship, evaluation of climate models). This often makes the use of conventional optimization algorithms prohibitive as these techniques require—especially for higher dimensional parameter spaces—a large number of objective function evaluations. Another issue is the numerical noise that is inherent to simulation models. The noise might be, among others, a result of terminating the simulation process before it fully converges or it may be related to adaptive meshing utilized by certain solvers (in which case, very small changes of the structure geometry lead to considerable changes of the mesh topology, and, consequently, to noticeable changes of the simulation results). The noise is particularly an issue for gradient-based methods that normally require smoothness of the objective function. The problem of the high computational cost can be alleviated to some extent by means of techniques such as adjoint sensitivities (Director and Rohrer 1969; Pironneau 1984; Jameson 1988; Papadimitriou and Giannakoglou 2008; El Sabbagh et al. 2006), or automated differentiation (Griewank 2000; Bischof et al. 2008). For many problems, these methods allow for a fast evaluation of gradients of the figures of interest at small additional computational effort (often only one additional simulation) regardless of the number of designable parameters. Thus, the benefits of adjoints are particularly evident for higher dimensional problems. Adjoint sensitivities are currently available in some commercial simulation packages in various areas (electromagnetic solvers: CST 2011; HFSS 2010; computational fluid dynamics solvers: ANSYS Fluent 2015; Star-CCM+ 2015), as well as some noncommercial codes (e.g., Stanford University Unstructured; Palacios et al. 2013).

In general, robust, automated, and computationally efficient optimization of expensive computational models remains an open problem. It should be mentioned here that—given the challenges of simulation-driven design mentioned in the previous paragraph—the most common way of utilizing computer simulations in design work is an interactive one. Although the details may vary in various fields, the basic flow is that an engineer is making predictions about possibly better designs (e.g., parameter setups) using his or her experience and insight, and utilizing simulation tools to verify these predictions. A typical realization would be to select (based on insight) a few parameters and to sweep them within certain ranges (also set up by experience), looking for an optimum. Parameter sweeps are normally executed one parameter at a time, and iterated until a satisfactory design is identified. For an experienced user, especially the one who solved a number of similar problems before, such a procedure usually leads to reasonable results; however, it might be very laborious and time consuming. Obviously, it does not guarantee optimum results, and it often fails when the user does not have a previous experience with similar class of problems, the number of relevant parameters is large, or optimum parameter setups are counterintuitive (which occurs for many contemporary structures and systems, such as miniaturized microwave and antenna components, Koziel et al. 2013b).

Despite the aforementioned difficulties, automated simulation-driven optimization is highly desirable. Simplified models and engineering insight can, in most cases, only lead to initial designs that have to be further tuned. Finding optimum designs normally require simultaneous adjustments of multiple parameters. Thus, numerical optimization techniques are indispensable. A large variety of conventional optimization techniques are available, including gradient-based methods (Nocedal and Wright 2000), e.g., conjugate-gradient, quasi-Newton, sequential quadratic programming, and derivative free (Kolda et al. 2003), e.g., Nelder-Mead, pattern-search techniques, as well as global optimization methods, including a popular class of population-based metaheuristics, e.g., genetic algorithms (Goldberg 1989), evolutionary algorithms (Back et al. 2000), ant systems (Dorigo and Gambardella 1997), particle swarm optimizers (Kennedy 1997), and many others (cf. Yang 2010). A common issue, from the simulation-driven design standpoint, is a large number of objective function evaluations required by majority of these methods, from dozens (for low-dimensional problems solved with gradient-based or pattern-search methods), through hundreds (for medium-size problems), to thousands, and tens of thousands (when using metaheuristics). Certain technologies, such as the aforementioned adjoint sensitivity and its commercial availability, alleviate this difficulty to some extent. Another factor is the continuous increase of available computational power (such as faster processors, parallel and distributed computing, and hardware acceleration). However, its impact is reduced by ever-growing need for more accurate (and therefore more expensive) simulations, including the necessity of simulating more complex systems.

Perhaps the most promising approach to realize computationally efficient optimization of expensive computational models is surrogate-based optimization (SBO) (Forrester and Keane 2009; Queipo et al. 2005; Booker et al. 1999; Koziel et al. 2011; Bandler et al. 2004a). The fundamental concept of SBO is to replace direct optimization of the expensive (high-fidelity) simulation model by means of an iterative construction, enhancement, and re-optimization of its cheap representation, referred to as a surrogate model. The surrogate is utilized as a prediction tool that enables the identification of promising locations of the search space (Forrester and Keane 2009; Koziel and Leifsson 2013a). New candidate solutions obtained through such a prediction are then verified by referring to the high-fidelity model. This additional data is also used to update the surrogate model and, consequently, improve its local (or global, depending on the SBO algorithm version) accuracy. The most important prerequisite of the SBO paradigm is that the surrogate is significantly faster than the high-fidelity model. It is critical because most operations in the SBO process are executed at the surrogate model level (in particular, its optimization is usually performed by means of conventional tools), and its numerous evaluations could become a major contributor to the overall optimization cost if the model is not sufficiently fast. At the same time, in many SBO algorithms, the high-fidelity model is only evaluated once per iteration, and—for a well-working

SBO process—the number of iteration is relatively small. As a result, the computational cost of the SBO-assisted design process may be significantly lower than for most of the conventional optimization methods.

Despite conceptual similarity, a number of rather distinct SBO techniques have evolved over the recent years. The most important difference between various SBO methods lies in the construction of the surrogate model. There are two major types of surrogate models. The first one comprises response surface approximation models constructed from sampled high-fidelity simulation data (Simpson et al. 2001a, b). A large variety of approximation (and interpolation) techniques are available, including artificial neural networks (Haykin 1998), radial basis functions (Wild et al. 2008), Kriging (Forrester and Keane 2009), support vector regression (Smola and Schölkopf 2004), Gaussian process regression (Angiulli et al. 2007; Jacobs 2012), or multidimensional rational approximation (Shaker et al. 2009), to name just the most popular ones. These data-driven surrogates are versatile, and if the design space is sampled with sufficient density the resulting model becomes a reliable representation of the system or device of interest so that it can be used for design purposes. In some cases, given a sufficiently accurate approximation surrogate, optimization process can be completely detached from the high-fidelity model (i.e., the optimum design can be found just by optimizing a surrogate without further reference to the simulation model). Unfortunately, computational overhead related to data acquisition and construction of such accurate approximations may be significant. Depending on the number of designable parameters, the number of training samples necessary to ensure usable (not to mention very high) accuracy might be hundreds, thousands, or even tens of thousands. Moreover, the number of samples quickly grows with the dimensionality of the problem (the so-called curse of dimensionality). As a consequence, globally accurate approximation modeling can only be justified in case of multiple-use library models of specific components described by a limited number of parameters. It is not suitable for ad hoc (one-time) optimization of systems evaluated through expensive simulations.

Iteratively improved approximation surrogates are becoming prevalent for global optimization (Gorissen et al. 2010). Various ways of incorporating new training points into the model (so-called infill criteria) exist, including exploitative models (i.e., models oriented towards improving the design in the vicinity of the current one), explorative models (i.e., models aiming at improving global accuracy), as well as model with balanced exploration and exploitation (Forrester and Keane 2009). Generally, these classes of techniques are often referred to as efficient global optimization (EGO) methods (Jones et al. 1998) or surrogate-assisted evolutionary algorithms (SAEAs) (Jin 2011; Lim et al. 2010).

Physics-based models constitute another class of surrogates whose adaptation has been growing recently due to the fact that SBO methods exploiting such models tend to be computationally more efficient than those exploiting approximation models. At the same time, surrogate construction is rather problem specific, and the related optimization algorithms generally require more complex implementation (Bandler et al. 2004a; Koziel and Leifsson 2013a). Physics-based surrogates are constructed from underlying low-fidelity (or coarse) models of the structures or systems being studied. Perhaps their biggest advantage is the fact that because the low-fidelity models embed system-specific knowledge, usually a small number of high-fidelity simulations are sufficient to configure a reliable surrogate. Low-fidelity models can be obtained in various ways: (1) as analytical models (in practice, a set of design-ready equations offering a considerably simplified description of the system); (2) by simulating the system at a different level (e.g., in microwave engineering: equivalent circuit representation evaluated using circuit theory rules versus full-wave electromagnetic simulation, Bandler et al. 2004a); and (3) lower fidelity or lower resolution simulation (e.g., simulation with coarser discretization of the structure and/or relaxed convergence criteria, Koziel and Ogurtsov 2014a, b). It should also be mentioned that because low-fidelity models are typically simulation ones, their evaluation time cannot—in many cases—be neglected and the aggregated computational cost of the low-fidelity model simulation may be significant (sometimes, a major) contribution to the overall cost of the SBO process (Leifsson et al. 2014a, b, c). Also, one of the important considerations is finding a proper balance between the accuracy and the speed of the low-fidelity model (Koziel and Ogurtsov 2012a), which may affect the SBO algorithm performance.

One of the most popular (although not the simplest in terms of formulation) SBO approaches using physics-based surrogates is space mapping (SM) (Bandler et al. 2004a, b, c; Koziel et al. 2008a). On the other hand, the most straightforward way of constructing the surrogate model from the underlying low-fidelity model is correcting its response. Various response correction techniques and related SBO algorithms can be found in the literature, including AMMO (Alexandrov and Lewis 2001), multi-point correction (Toropov 1989), manifold mapping (Echeverria and Hemker 2005), adaptive response correction (Koziel et al. 2009), or shape-preserving response prediction (Koziel 2010a). Some of the above methods are based on simple analytical formulas, and others (mostly non-parametric ones) aim at tracking the response changes of the low-fidelity model, thus ensuring better generalization capability, although at the cost of increased formulation complexity and more restrictive assumptions regarding applicability of a given method.

This book focuses on surrogate-based optimization techniques exploiting response-corrected physics-based surrogates. At the same time, for the sake of keeping the text self-contained, related background material is also covered to some extent, including the basics of classical and surrogate-based optimization. We begin, in Chap. 2, by formulating the simulation-driven optimization problem and discussing its major challenges. In Chap. 3, we briefly discuss conventional numerical optimization techniques, both gradient-based and derivative-free methods, including population-based metaheuristics. Chapter 4 is an introduction to surrogate-based optimization, where we discuss—on a generic level—the SBO design workflow as well as various aspects of surrogate-based optimization. In the same chapter, an outline of surrogate modeling is given, both regarding function approximation and physics-based models. In Chap. 5, we introduce surrogate modeling through a response correction, outline parametric and non-parametric correction techniques, as well as discuss some practical issues on the construction and trade-offs for low-fidelity models. Chapters 6 and 7 provide formulation and application examples of several specific response correction techniques, both parametric (Chap. 6) and non-parametric (Chap. 7) ones. The remaining chapters discuss various methods and aspects of physics-based SBO such as the adaptively adjusted design specification approach (Chap. 8), optimization using response features (Chap. 9), response correction methods enhanced by adjoint sensitivities (Chap. 10), multi-objective optimization exploiting response correction (Chap. 11), and response correction techniques for physics-based surrogate modeling (Chap. 12). The book is concluded in Chap. 13 where we formulate recommendations for the readers interested in applying the presented algorithms and techniques in their design work and discuss possible future developments concerning automation of simulation-driven optimization. The book is illustrated with a large number of practical design cases from various engineering disciplines including electrical engineering, antenna design, aerospace engineering, hydrodynamics, and others.

Chapter 2
Simulation-Driven Design

In this chapter, the simulation-driven design task is formulated as a nonlinear minimization problem. Among others, we introduce the notation used throughout the book, discuss typical design objectives and constraints, as well as describe common challenges, mostly related to the high computational cost of evaluating simulation models of the devices and systems under consideration. A brief outline of conventional numerical optimization methods is provided in Chap. 3. The concept of surrogate-based optimization is discussed, including various types of surrogate models, in Chap. 4.

2.1 Formulation of the Optimization Problem

We will denote the response of a high-fidelity (or fine) simulation model of a device or system under design by $\boldsymbol{f}(\boldsymbol{x})$. Typically, $\boldsymbol{f}$ will represent an evaluation of the performance characteristics of interest (see Sect. 2.2 for specific examples). The vector $\boldsymbol{x} = [x_1\ x_2\ \ldots\ x_n]^T$ represents the designable parameters to be adjusted (e.g., related to the geometry and/or the material).

In many situations, individual components of the vector $\boldsymbol{f}(\boldsymbol{x})$ will be considered and we will use the notation $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x})\ f_2(\boldsymbol{x})\ \ldots\ f_m(\boldsymbol{x})]^T$, where $f_k(\boldsymbol{x})$ is the kth component of $\boldsymbol{f}(\boldsymbol{x})$. In particular, we might have $\boldsymbol{f}(\boldsymbol{x}) = [f(\boldsymbol{x},t_1)\ f(\boldsymbol{x},t_2)\ \ldots\ f(\boldsymbol{x},t_m)]^T$, where t_k, $k = 1, \ldots, m$, parameterizes the components of $\boldsymbol{f}(\boldsymbol{x})$; for example, t might represent time, frequency, or a relevant geometry variable (such as the airfoil thickness). In some cases, the response $\boldsymbol{f}(\boldsymbol{x})$ may actually consist of several scalars and/or vectors representing the performance characteristics of interest.

The simulation-driven design task is usually formulated as a nonlinear minimization problem of the following form (Koziel and Leifsson 2013a):

S. Koziel, L. Leifsson, *Simulation-Driven Design by Knowledge-Based Response Correction Techniques*, DOI 10.1007/978-3-319-30115-0_2

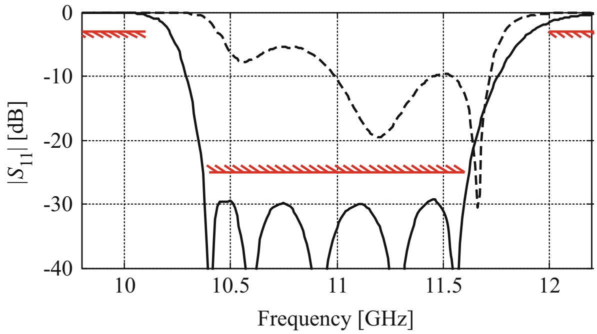

Fig. 2.1 Illustration of minimax design specifications, here, $|S_{11}| \leq -25$ dB for frequencies from 10.4 to 11.6 GHz, and $|S_{11}| \geq -3$ dB for frequencies lower than 10.1 GHz and higher than 12.0 GHz, marked with *thick horizontal lines*. An example waveguide filter response that does not satisfy our specifications (*dashed line*) (specification error, i.e., maximum violation of the requirements is about + 22 dB at 10.4 GHz), and another response that does satisfy the specifications (*solid line*) (specification error, i.e., the minimum margin of satisfying the requirements is about 1.5 dB at 12 GHz)

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} U(\boldsymbol{f}(\boldsymbol{x})) \tag{2.1}$$

Here, U is the scalar merit function encoding the design specifications, whereas $\boldsymbol{x}^*$ is the optimum design to be found. The composition $U(\boldsymbol{f}(\boldsymbol{x}))$ will be referred to as the objective function. The function U is implemented so that a better design $\boldsymbol{x}$ corresponds to a smaller value of $U(\boldsymbol{f}(\boldsymbol{x}))$.

In some cases, definition of the objective function is pretty straightforward (e.g., minimization of the drag coefficient of an airfoil, and the maximization of the heat transfer of a heat exchanger). Thus, in many cases, the objective function can be defined in a norm sense as $U(\boldsymbol{f}(\boldsymbol{x})) = ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}^*||$, where $\boldsymbol{f}^*$ is a target response.

In several areas, optimization problems are formulated in a minimax sense with upper (and/or lower) specifications. Figure 2.1 shows an example of minimax specifications corresponding to typical requirements for microwave filters, here, $|S_{11}| \leq -25$ dB for 10.4–11.6 GHz, and, at the same time, $|S_{11}| \geq -3$ dB for frequencies lower than 10.1 GHz and higher than 12.0 GHz. The value of $U(\boldsymbol{f}(\boldsymbol{x}))$ (also referred to as the minimax specification error) corresponds to the maximum violation of the design specifications within the frequency bands of interest.

To simplify notation, we will occasionally use the symbol $\boldsymbol{f}(\boldsymbol{x})$ as an abbreviation for $U(\boldsymbol{f}(\boldsymbol{x}))$.

In practical situations, the problem (2.1) is almost always constrained. The following types of constraints can be considered:

- Lower and upper bounds for the design variables, i.e., $l_k \leq x_k \leq u_k$, $k = 1, \ldots, n$.
- Inequality constraints, i.e., $c_{ineq.l}(\boldsymbol{x}) \leq 0$, $l = 1, \ldots, N_{ineq}$, where N_{ineq} is the number of constraints.
- Equality constraints, i.e., $c_{eq.l}(\boldsymbol{x}) = 0$, $l = 1, \ldots, N_{eq}$, where N_{eq} is the number of constraints.

Design constraints may be introduced to make sure that the device or system that is to be evaluated by the simulation software is physically valid (e.g., linear dimensions assuming nonnegative values). Also, constraints can be introduced to ensure that the physical dimensions (length, width, area, and volume) or other characteristics (e.g., cost) of the structure do not exceed certain assumed values. In some cases, evaluation of the constraints may be just as expensive as evaluating the objective function itself.

Another important aspect of simulation-driven design is that the majority of practical problems are multi-objective ones; that is, there is more than one performance characteristic that should be considered at a time. Typically, objectives conflict with each other so that the improvement of one is only possible at the expense of degrading the others. In many cases, a priori preference articulation is possible, so that only one specific objective is to be explicitly optimized, whereas the others can be handled through design constraints. This allows for solving the problem through single-objective formulation. Sometimes, however, this is neither possible nor convenient, e.g., when acquiring knowledge about possible trade-offs between competing objectives is important. In those situations a genuine *multi-objective* optimization is necessary. Typically, a solution to multi-objective design is represented as the so-called Pareto front (Tan et al. 2005), which represents the set of the best possible designs which are non-commensurable in the conventional (single-objective) sense.

A popular approach to handle multiple objectives is aggregation, using, for example, the weighted sum method (Tan et al. 2005), where

$$U(\boldsymbol{f}(\boldsymbol{x})) = \sum_{k=1}^{N_{obj}} w_k U_k(\boldsymbol{f}(\boldsymbol{x})) \tag{2.2}$$

Here, $U_k(\boldsymbol{f}(\boldsymbol{x}))$ represents the kth objective, whereas w_k, $k = 1, \ldots, N_{obj}$, are weighting factors (usually, convex combinations are considered, with $0 \le w_k \le 1$ and $\sum_k w_k = 1$). It should be emphasized that combining objectives into a single scalar function only allows for finding a single solution, i.e., one point on the Pareto front. Other solutions can be obtained (with some restrictions) by optimizing an aggregated objective function such as (2.2) with different setups of the weighting factors.

A discussion of genuine multi-objective optimization that aims at identifying the entire set of solutions representing the Pareto front exceeds the scope of this book. See for example Deb (2001). Some information, in the context of specific application examples, can be found in Chap. 11.

2.2 Design Challenges

The main focus of this book is on the optimization of simulation models. Perhaps the major challenge is the fact that realistic (in particular, accurate) simulations of real-world devices and systems tend to be computationally expensive. When

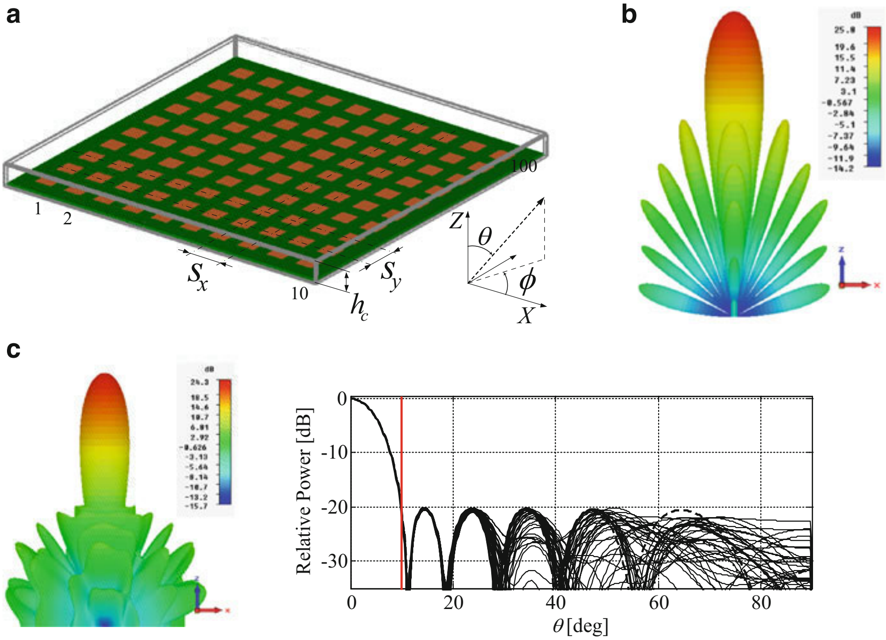

Fig. 2.2 Microstrip array antenna (Koziel and Ogurtsov 2015a): (**a**) Geometry: the structure contains 100 independent element (radiators); radiator geometry, element spacings, as well as excitation amplitudes and phases are designable parameters (over 200 in total); electromagnetic simulation of the structure takes about 1 h; (**b**) radiation pattern: the objective is to control the main beam width while minimizing radiation in other directions (or reduce the so-called side lobes); (**c**) radiation response of the optimized structure (note lower side lobe levels)

combined with other issues such as a large number of designable parameters, complex objective function landscapes, numerical noise, as well as multiple objectives, the simulation-driven optimization becomes a very challenging problem which—in many cases—is essentially unsolvable with conventional techniques unless massively parallel computations are utilized. In this section, we give a few examples of typical simulation-based optimization problems in various engineering areas. The presented problems are concerned with single devices rather than systems (the latter being even more challenging from computational point of view).

Figure 2.2 shows an example of a typical array antenna, where one of the major tasks is the synthesis of the radiation pattern, in particular the reduction of the side lobes (cf. Fig. 2.2b–c). Evaluating a high-fidelity electromagnetic (EM) simulation model of the array antenna can take about 1 h. At the same time, the number of designable parameters is large (in this case the parameters are related to the element geometries, element spacings, locations of feeds, and excitation amplitudes and phases, and are over 200 in total). Moreover, global optimization is necessary due to a rugged objective landscape. Optimization of the structure using conventional methods, such as gradient-based search (not to mention population-based

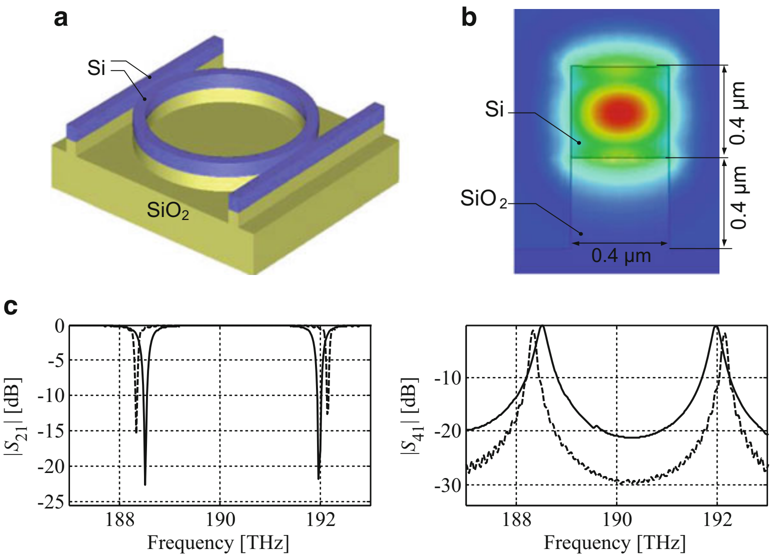

Fig. 2.3 Microring resonator (Koziel and Ogurtsov 2014b): (**a**) 3D view; (**b**) E-field amplitude distribution; (**c**) initial (*dashed line*) and optimized (*solid line*) frequency characteristics. The problem has only a few design variables but the high-fidelity simulation model is very expensive (~20 h with GPU acceleration)

metaheuristic algorithms), is virtually impossible in a reasonable timeframe. Figure 2.2c shows the optimized pattern, obtained using a surrogate-based methodology with an underlying analytical array factor model and a suitably defined response correction (Koziel et al. 2015).

Figure 2.3 shows an example of an integrated photonic component, a so-called add-drop microring resonator (Bogaerts et al. 2012). Although single components like this one are described by a rather small number of parameters (here, just three), the high-fidelity EM simulation is very expensive (~20 h of CPU time with GPU acceleration when using fine discretization of the structure). Simulation times of more involved structures (such as coupled microrings) are significantly longer. Figure 2.3c shows initial and optimized characteristics of the microring for an example optimization problem, here, obtained at a coarse-mesh simulation setup and exploiting the so-called feature-based optimization (less than 20 EM simulations in total; Koziel and Ogurtsov 2014b).

Another example of an expensive engineering problem is the design of a transport aircraft (Fig. 2.4a), which is a highly coupled multidisciplinary system—involving disciplines such as aerodynamics, structures, controls, and weights—with thousands of design variables. One of the major concerns in aircraft design is the aerodynamic performance where the wing shape plays a major role (Holst and Pulliam 2003; Leoviriakit and Jameson 2005; Epstein and Peigin 2006; Mavriplis 2007; Leung 2010). The evaluation time of a single high-fidelity model of the external aerodynamics at transonic speeds can be more than 24 h (depending on the available computational power) (Leifsson and Koziel 2015a).

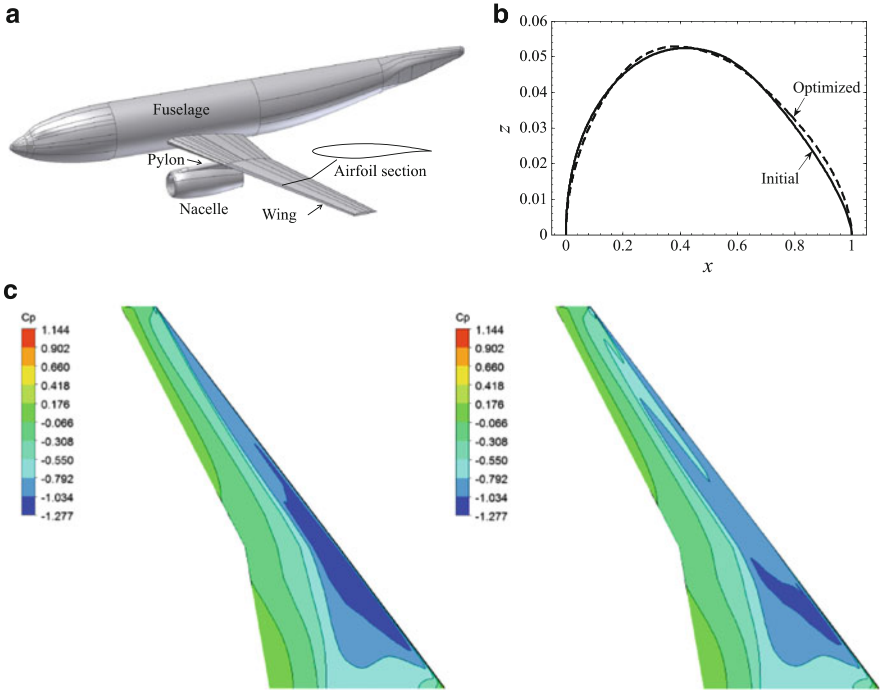

Fig. 2.4 Wing of a typical transport aircraft (Leifsson and Koziel 2015a): (**a**) wing-body-nacelle configuration, (**b**) the upper surface shapes of the initial (*solid*) and optimized (*dashed*) airfoil, (**c**) contours of pressure coefficient of the initial (*left*) and optimized (*right*) wings

Direct aerodynamic shape optimization of the wing (only one of the disciplines) is impractical without adjoint sensitivity information (Jameson 1988). Even with this information, the aerodynamic shape design task is challenging, due to numerical noise in the computational fluid dynamics (CFD) models, as well as the multimodality of the design space. Figure 2.4b shows the shape of the upper surface of one of the wing airfoil sections, before and after optimization with a surrogate-based optimization technique and a shape parameterization with five variables (Jonsson et al. 2013a). The total number of equivalent high-fidelity model evaluations to obtain the optimized design is 15 (the evaluation time of the high-fidelity model is 27.5 h, the surrogate model is 100 times faster), or 17 days. Figure 2.4c shows the pressure distribution of the initial and optimized designs, indicating a significant reduction of upper surface pressure shock strength, leading to a reduction in wing drag by 4.6 %.

The design of trawl-doors—which are an important part of a typical fishing gear (Fig. 2.5a)—is another example involving expensive aerodynamic (or hydrodynamic) CFD simulations, but in a different medium and at much lower speeds (Haraldsson 1996; Jonsson et al. 2013a). Trawl-doors function much like aircraft wings (an example configuration is shown in Fig. 2.5b); that is, they maintain certain lift for a given operating condition, but are always operated at

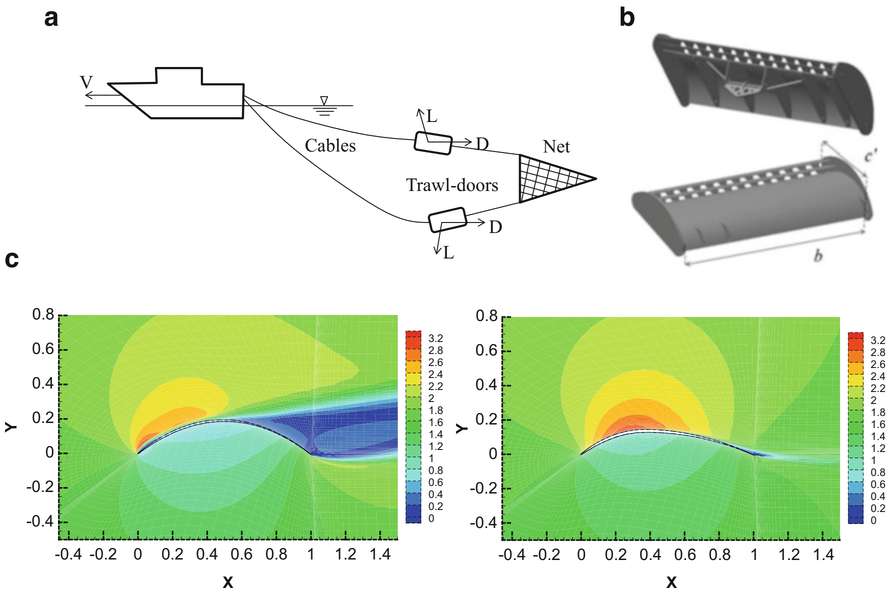

Fig. 2.5 Trawl-doors on a fishing vessel (Jonsson et al. 2013a): (**a**) main parts of the fishing gear (not drawn to scale), and (**b**) a typical trawl-door with two slots at the leading edge, and (**c**) velocity contours (in m/s) of initial (*left*) and optimized designs (*right*)

high lift. The flow physics of these devices are highly complex and become increasingly difficult to simulate with added number of airfoil elements. A two-dimensional CFD simulation of a trawl-door takes around 4 h on a typical desktop computer. Designable parameters are on the order of 10–15 in the two-dimensional single-element case, but can go up to 50 for a full three-dimensional multi-element case. Figure 2.5c shows the velocity contours around an initial and optimized single-element trawl-door (Leifsson et al. 2014a). The initial and optimized designs yield the same lift forces. However, the initial design operates at a high angle of attack (around 25°) with massive flow separation on the upper surface, whereas the optimized design operates at the same lift but at a much lower angle of attack (around 7°) and with almost no flow separation due to an improved shape.

The simulation-driven design examples discussed here illustrate problems that are common for many engineering disciplines. Among these, high computational cost of evaluating simulation models is usually a major bottleneck when it comes to numerical optimization. As also indicated, surrogate-based methods—particularly if properly tailored to a given class of problems—may allow for solving such tasks in a reasonable timeframe.

Chapter 3
Fundamentals of Numerical Optimization

Although the main focus of the book is on surrogate-assisted optimization using physics-based low-fidelity models and response correction techniques, we provide—for the sake of making the material self-contained—some basic information about conventional optimization algorithms. In this book, we refer to conventional (or direct) methods as those that handle the expensive simulation model directly in the optimization scheme (as opposed to surrogate-based approaches where most of the operations are carried out using a fast surrogate). In particular, each candidate solution suggested by the optimizer is evaluated using a computationally expensive high-fidelity simulation. Figure 3.1 shows a generic flow of the direct simulation-driven optimization process.

In this chapter, we provide an outline and a brief overview of conventional optimization techniques, including gradient-based and derivative-free methods, as well as metaheuristics. The readers interested in further information on these techniques are referred to the literature (e.g., Nocedal and Wright 2000; Yang 2010; Conn et al. 2009).

3.1 Optimization Problem

In this chapter, we consider the following formulation of the optimization problem:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}) \tag{3.1}$$

where $f(\boldsymbol{x})$ is a scalar objective function, whereas $\boldsymbol{x} \in X \subseteq R^n$ is the search space (the objective function domain). We recall that in practice, the objective function is a composition $U(\boldsymbol{f}(\boldsymbol{x}))$ of the merit function U and (usually vector-valued) system response $\boldsymbol{f}(\boldsymbol{x})$, cf. Chap. 2. The objective function represents the performance of the system of interest. An alternative formulation of the problem with explicit

S. Koziel, L. Leifsson, *Simulation-Driven Design by Knowledge-Based Response Correction Techniques*, DOI 10.1007/978-3-319-30115-0_3

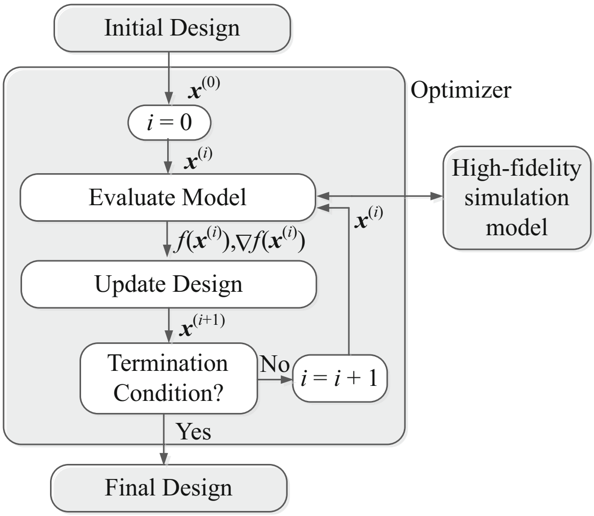

Fig. 3.1 Direct simulation-driven optimization flow. The candidate designs generated by the algorithm are evaluated through a high-fidelity simulation for verification purposes and to provide the optimizer with information to search for better designs. The search process may be guided by the model response only, or by (if available) the response as well as its derivatives (gradient)

constraints (lower/upper bounds, linear and nonlinear inequality or equality constraints) has been presented in Chap. 2.

In the remaining section of this chapter we outline the conventional optimization methods, both gradient based and derivative free (including population-based metaheuristics).

3.2 Gradient-Based Optimization Methods

Gradient-based algorithms belong to the most popular and widely used optimization techniques (Nocedal and Wright 2000). The fundamental piece of information directing the search for a better solution is the gradient $\nabla f(\boldsymbol{x})$ of the objective function $f(\boldsymbol{x})$ defined as

$$\nabla f(\boldsymbol{x}) = \left[\frac{\partial f}{\partial x_1}(\boldsymbol{x}) \quad \frac{\partial f}{\partial x_2}(\boldsymbol{x}) \quad \ldots \quad \frac{\partial f}{\partial x_n}(\boldsymbol{x})\right]^T \tag{3.2}$$

Assuming that f is sufficiently smooth (i.e., at least continuously differentiable), the gradient provides information about the local behavior of f in the neighborhood of $\boldsymbol{x}$. In particular, we have

$$f(\boldsymbol{x} + \boldsymbol{h}) \cong f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \cdot \boldsymbol{h} < f(\boldsymbol{x}) \tag{3.3}$$

for a sufficiently small vector $\boldsymbol{h}$, provided that $\nabla f(\boldsymbol{x})^T \cdot \boldsymbol{h} < 0$. Furthermore, $\boldsymbol{h} = -\nabla f(\boldsymbol{x})$ determines the direction of the (locally) steepest descent. There are two basic ways to utilize the gradient in the search process: (1) by moving along a descent