The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-9 June 2022, Oslo, Norway

CHALLENGES OF INTEGRATING ADJOINT SIMULATIONS IN INDUSTRIAL TURBOMACHINERY MDO

Jan Backhaus[1] Christian Voß[2] and Christian Frey[3]

Institute of Propulsion Technology, German Aerospace Center (DLR),
Linder Höhe, 51147 Cologne, Germany,
[1] jan.backhaus@dlr.de
[2] christian.voss@dlr.de
[3] christian.frey@dlr.de

Keywords: *MDO, adjoint CFD, gradient enhanced surrogate models, turbomachinery*

Abstract

The design of turbomachinery creates a strong demand for the simultaneous optimization of multiple blade rows with regard to different disciplines including aerodynamics, aeroelasticity, and solid mechanics. Established gradient-free methods, typically surrogate-based methods, have been successfully applied to the optimization of single blade rows and pairs of adjacent rows, typically featuring in the order of 50 design variables per blade row. Gradient-free methods become inhibitively expensive through the increased number of design variables from simultaneous optimizations of many rows. Gradients obtained from adjoint simulations can help in transitioning to larger design spaces as they provide derivatives with respect to each design variable at a computational cost that only depends on the number of objectives. For the transition from gradient-free to gradient-based optimizations, a variety of challenges had to be solved, which will be outlined in this paper.

Practical gradient-based optimization

Throughout the last few years, the adjoint method for computing cost-efficient gradients of computer simulations has become widely adopted, first in academia and increasingly in industry. The advantage of gradient-based over gradient-free optimization methods is obvious for high dimensional optimization problems on sufficiently smooth objectives and constraints.

However, the practical application for the design of turbomachinery with simulation-based objectives is still a challenging task. The challenges fall into the following categories

- Creating an adjoint to the simulation and post-processing
- Choice of parameterization
- Differentiation of the process around the simulation

- Implementing a failure tolerating multi-objective optimization strategy

Generating an efficient adjoint to an existing simulation code is a large topic in itself. What is important to emphasize here, is that industrial CFD typically employs a large variety of modeling extensions. The adjoint to such a primal CFD solver has to differentiate those modeling extensions or otherwise, gradients become inexact. Even though simplifications of the adjoint have been closely examined by Dwight et al. [1], acceptance of adjoint methods can be diminished by restrictions to the range of supported models. We, therefore, followed an approach to first differentiate the complete code by algorithmic differentiation through operator overloading and thereafter apply optimizations, like e.g. fixpoint iteration schemes to the result, to lower CPU-time and memory consumption [2, 3]. This approach has also been described in the context of OpenFOAM [4] and SU2 [5]. This way, all features of the code are differentiated and the resulting adjoint code can always be verified against a black-box reverse mode or black-box tangent mode version.

While many choices for parameterization of turbomachinery blades are described in the literature, industrial users of optimizations often demand that the final result of the optimization be represented in the specific CAD parameterization that has been used for manual design. Those problem-specific parameterizations contain engineering know-how and encode restrictions which cannot be easily enforced with domain agnostic parameterizations. Since optima in another parameterization can't be easily converted to the CAD parameterization without loss of optimality, optimizations are best carried out in the established CAD-based parameterization. This means, that derivatives with respect to such parameterization must be computed. Since in general, the respective CAD tools are third-party software, intrusive methods for computing derivatives, such as the complex-step method or algorithmic differentiation cannot be applied to them. Though it has been demonstrated that CAD kernels can be algorithmically differentiated when the source is available [6].

Everything before the actual simulation, including all modifications to the CAD model and grid generation, can be differentiated by finite-differences if intrusive differentiation is not possible. When differentiating such tools by finite-differences, care must be taken to find step-widths small enough to be in the range of linear behavior of the response, but at the same time large enough to be represented exact enough through the (iterative) numerical algorithms inside the toolchain. Determining a good step-width for finite-differences of numerical procedures has been a research topic for quite some time [7, 8], still it remains challenging and a possible source of error for gradient-based optimizations. The best solution would be if producers of the geometry and meshing tools would provide gradients of their tools.

Having validated partial derivatives at hand, the next question is how to efficiently use these inside an optimization. What distinguishes our focused class of optimizations is that the objectives are expensive to evaluate. Additionally, there are typically multiple objectives and constraints that stem from different disciplines, fluid dynamics and computational mechanics for instance, for which a set of Pareto-optimal solution is sought. These characteristics already occur in the gradient-free case and surrogate models play an important part in their solution. To carry over the benefits of surrogate models, surrogate models were extended to not only reproduce data points but also gradients. A popular approach to this is a gradient extension of Gaussian Process Regression [9], often referred to as Gradient Enhanced Kriging (GEK). GEK has gained popularity in the field

of simulation-based optimization for a variety of beneficial properties. Primarily, GEK can be performed for datasets with missing points or gradients, which makes the optimization process resilient to failing steps of the evaluation process. Additionally, GEK is capable of handling numerical noise in the objective function or gradients. And last but not least, multiple objective functions with and without adjoint gradients can be treated in the same framework to find sets of pareto optimal points. The feasibility of GEK to be used in design optimizations of turbomachinery using an optimization akin to industrial design optimization has been demonstrated before [3]. Here we will shortly recapitulate the main features of this optimization and discuss the implications for the industrial adaptation of adjoint methods.

Detailed descriptions of GEK can be found in textbooks, e.g. Forrester, Sóbester, Keane [10]. The key points will be briefly repeated here to introduce the notation. The Kriging implementation used here is described in [11]. GEK constructs a model $\hat{y}$ for an unknown function $y : \mathbb{R}^N \mapsto \mathbb{R}^M$ based on P observations of the function and Q observations of the gradient of the function $Y = [y^{(1)}, \ldots, y^{(P)}, \frac{\partial y(1)}{\partial x^{(1)}}, \ldots, \frac{\partial y(P)}{\partial x^{(N)}}]$. Note, that not all partial derivatives have to be provided, indeed complete gradients can be missing without causing problems.

The prediction is based on the assumption that the function can be represented by its mean plus corrections which are linear combinations of the available observations

$$\widehat{y}(x) = \mu + c^T(x)\mathbf{\Psi}^{-1}(Y - F\mu). \tag{1}$$

The weighting of the observations is represented by the correlation matrix $\mathbf{\Psi}$ and assumes that the correlation of two observations is only a function of their distance

$$\mathrm{Corr}(Y(x_i), Y(x_j)) := e^{-r(x_i, x_j)}. \tag{2}$$

As a correlation structure we chose here the squared exponential, which is equivalent to the Gaussian Correlation function

$$r(x^{(i)}, x^{(j)}) := \sum_{k=1}^{N} \theta_k |x_k^{(i)} - x_k^{(j)}|^2. \tag{3}$$

This encodes the prior belief, that the function to interpolate is infinitely often differentiable, which seems quite restrictive, but has turned out to be successful in practice. To include gradient observations in the prediction the correlation of derivatives with points are included, which are the derivatives of the correlation function

$$\mathbf{\Psi} = \begin{pmatrix} \mathrm{Corr}(x_i, x_j) & \frac{\partial \mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_l)}{\partial \mathbf{x}_l^n} \\ \frac{\partial \mathrm{Corr}(\mathbf{x}_l, \mathbf{x}_i)}{\partial \mathbf{x}_l^n} & \frac{\partial^2 \mathrm{Corr}(\mathbf{x}_j, \mathbf{x}_l)}{\partial \mathbf{x}_l^m \partial \mathbf{x}_l^n} \end{pmatrix} \tag{4}$$

The correlation function has parameters θ_k that determine how fast the correlation decays with distance in each dimension. This enables anisotropy of the space of design parameters. Determining these correlation lengths is called training of the model. Training is done by maximizing the likelihood of the model under the already observed data using a

concentrated logarithmic variant of the likelihood function

$$\ell(\theta) = -\frac{1}{2}\ln(\det(\mathbf{\Psi})) - \frac{D}{2}\ln(\hat{\sigma}^2). \tag{5}$$

In each training iteration, one needs the determinant of the correlation matrix which is obtained by computing the Cholesky-Factorization. It is within this step which is most expensive in GEK and where the model size and ill-conditioning become problematic.

Challenges with GEK

In comparison to the gradient-free variant, referred to as ordinary Kriging, GEK is challenging for the following reasons: The computational effort for ordinary Kriging grows on the number of samples N. For GEK it is proportional to a product of samples and degrees of freedom $(M+1)N$. On the one hand, the addition of M pieces of information for the cost of one additional simulation is the central idea of adding adjoint gradients to the observed training data. On the other hand, this leads to a fast growing correlation matrix, and therefore approximately cubically increases the time to train the kriging model. Consequently, without modifications, GEK does not completely overcome the curse of dimensionality.

Ill-conditioning of Kriging is an often-discussed topic in the literature [12, 13, 14, 15, 16], which becomes problematic during the factorization of the correlation matrix. There are six principal sources of ill-conditioning for ordinary kriging [17] which also apply to GEK: Too narrowly spaced samples (1) cause columns of the correlation matrix to become linear combinations of each other, or the correlation lengths are selected too large (2) during the training. A mismatch between the selected correlation model and the real smoothness of the observed function (3) causes ill-conditioning which rises with the number of inconsistent observations. Numerical noise in the responses (4) accumulates as inconsistencies in the training data which increasingly impedes the model condition. And last but not least there is the problem of scaling, for the parameters (5) and the model responses (6).

Scaling (5, 6) can be solved by non-dimensionalization parameters using a given range and responses by a suitable upper bound. Problems from a mismatch in the correlation structure (3) can be countered through a choice of a correlation function which assume less smoothness of the observations, for example the Matern or thin-plate spline class of correlation kernels. However, this comes at the cost of more computational cost during the training, since these functions and their derivatives are more expensive to evaluate than the squared exponential. A measure to account for noise in the response (4) is the Tikhonov regularization, sometimes called nugget in the context of Kriging (c.f. [18]). It changes the interpolating model into a regression model that only approximates the training data. Adequately constraining the correlation lengths during the training can usually be achieved by tuning the optimization process for the training procedure. Problem (1) requires a careful choice of the sampling positions in order not to produce too highly correlated samples or to eliminate to highly correlated points from the training dataset or regularization.

Solution Approaches

Different solutions to the problem of building GEK surrogates in larger design spaces are described in the literature.

In our previous studies [19, 3] we used a restarting technique for consecutive model trainings and Tikhonov regularization that treats function samples different from gradient observations:

$$\tilde{\Psi} = \Psi + (\epsilon_{i,j}) \tag{6}$$

$$\epsilon_{i,j} = \begin{cases} \epsilon_f & \text{if} \quad i = j, i \leq P \\ \epsilon_g & \text{if} \quad i = j, i > P \\ 0 & \text{if} \quad i \neq j \end{cases} \tag{7}$$

De Baar et al. [14] account for different levels of numerical noise in the function values and gradients as observations in the likelihood optimization of GEK. Dalbey [20] presents an overview of methods to overcome ill-conditioning and proposes a pivoted Cholesky factorization in which the samples which provide the least new information are removed from the correlation matrix along with their gradients. One can compute and constrain the condition number during the training of the hyperparameters and therefore avoid too high correlation lengths as e.g. described by Ollar et al. [15]. Bouhlel and Martins [16] propose a partial least square on the linearization around sampling points to only add artificial samples corresponding to a small number of principal components of the derivative information. Han et al. [21] use a weighted combination of sub-models containing only a selection of the samples and gradients. Özkaya et al. [22] employ a convex-combination of gradient-free and gradient-enhanced Kriging models to improve generalization of GEK models under numerical noise in the gradients.

All the above methods help reducing GEKs ill-conditioning by improving the model for given observations. The idea of providing adjoint gradients is to ameliorate the curse of dimensionality by providing more information per simulation and thus spend less time on simulations. Instead of discarding redundant function values and gradients for which computing time was already spent, it would be more beneficial to only produce information where it is necessary. For optimizations, the infill is usually governed by variations of the "Efficient global optimization" strategy by Jones et al. [23] which aim at balancing Exploitation and Exploration of the meta-model. However, the exploitation part of these methods tends to produce tightly clustered samples. If we add gradient information at every sampling point, these highly correlated gradients tend to produce a highly ill-conditioned correlation matrix. Therefore, future research should focus on gradient aware infill strategies to ensure that gradients are only calculated if is to be expected that the surrogate model benefits from the information.

REFERENCES

[1] Richard P Dwight and Joël Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA journal*, 44(12):3022–3031, 2006.

[2] Max Sagebaum, Emre Özkaya, Nicolas R. Gauger, Jan Backhaus, Christian Frey, Sebastian Mann, and Marc Nagel. Efficient algorithmic differentiation techniques for turbo-machinery design. In *AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, June 2017.

[3] Jan Backhaus, Andreas Schmitz, Christian Frey, Sebastian Mann, Marc Nagel, Max Sagebaum, and Nicolas R. Gauger. Application of an algorithmically differentiated turbomachinery flow solver to the optimization of a fan stage. In *AIAA AVIATION Forum*, 2017.

[4] Markus Towara. *Discrete adjoint optimization with OpenFOAM*. PhD thesis, RWTH Aachen, 2018.

[5] Thomas D Economon. Simulation and adjoint-based design for variable density incompressible flows with heat transfer. *AIAA Journal*, 58(2):757–769, 2020.

[6] Mladen Banović, Orest Mykhaskiv, Salvatore Auriemma, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. Algorithmic differentiation of the Open CASCADE Technology CAD kernel and its coupling with an adjoint CFD solver. *Optimization Methods and Software*, 33(4-6):813–828, 2018.

[7] Philip E Gill, Walter Murray, Michael A Saunders, and Margaret H Wright. Computing forward-difference intervals for numerical optimization. *SIAM Journal on Scientific and Statistical Computing*, 4(2):310–321, 1983.

[8] Ravishankar Mathur. *An Analytical Approach to Computing Step Sizes for Finite-Difference Derivatives*. PhD thesis, University of Texas, 2012.

[9] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

[10] Alexander Forrester, András Sóbester, and Andy Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008.

[11] Andreas Schmitz. *Multifidelity-Optimierungsverfahren für Turbomaschinen*. PhD thesis, Ruhr-Universität Bochum, 2020.

[12] Richard P Dwight and Zhong-Hua Han. Efficient uncertainty quantification using gradient-enhanced kriging. *AIAA paper*, 2276:2009, 2009.

[13] R Zimmermann. On the maximum likelihood training of gradient-enhanced spatial gaussian processes. *SIAM Journal on Scientific Computing*, 35(6):A2554–A2574, 2013.

[14] Jouke H.S. de Baar, Richard P. Dwight, and Hester Bijl. Improvements to gradient-enhanced kriging using a bayesian interpretation. *International Journal for Uncertainty Quantification*, 4(3):205–223, 2014.

[15] Jonathan Ollar, Charles Mortished, Royston Jones, Johann Sienz, and Vassili Toropov. Gradient based hyper-parameter optimisation for well conditioned kriging metamodels. *Structural and Multidisciplinary Optimization*, 55(6):2029–2044, 2017.

[16] Mohamed A Bouhlel and Joaquim RRA Martins. Gradient-enhanced kriging for high-dimensional problems. *Engineering with Computers*, 35(1):157–173, 2019.

[17] George J Davis and Max D Morris. Six factors which affect the condition number of matrices associated with kriging. *Mathematical geology*, 29(5):669–683, 1997.

[18] Alexander I J. Forrester, Andy J Keane, and Neil W Bressloff. Design and analysis of "noisy" computer experiments. *AIAA journal*, 44(10):2331–2339, 2006.

[19] Jan Backhaus, Marcel Aulich, Christian Frey, Timea Lengyel, and Christian Voß. Gradient enhanced surrogate models based on adjoint CFD methods for the design of a counter rotating turbofan. In *Proceedings of the ASME Turbo Expo 2012*, 2012.

[20] Keith R. Dalbey. Efficient and robust gradient enhanced kriging emulators. SANDIA REPORT SAND2013-7022, Sandia National Laboratories Albuquerque, New Mexico 87185 and Livermore, California 94550, August 2013.

[21] Zhong-Hua Han, Yu Zhang, Chen-Xing Song, and Ke-Shi Zhang. Weighted gradient-enhanced kriging for high-dimensional surrogate modeling and design optimization. *Aiaa Journal*, 55(12):4330–4346, 2017.

[22] Emre Özkaya and Nicolas R Gauger. Global aerodynamic design optimization via primal-dual aggregation method. In *Symposium der Deutsche Gesellschaft für Luft- und Raumfahrt*, pages 48–57. Springer, 2018.

[23] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.