



Article Gradient-Free and Gradient-Based Optimization of a Radial Turbine[†]

Nicolas Lachenmaier ^{1,*}, Daniel Baumgärtner ², Heinz-Peter Schiffer ³, and Johannes Kech ¹

- ¹ MTU Friedrichshafen GmbH, Maybachplatz 1, 88045 Friedrichshafen, Germany; johannes.kech@ps.rolls-royce.com
- ² Technical University of Munich, Chair of Structural Analysis, Arcisstr. 21, 80333 Munich, Germany; daniel.baumgaertner@tum.de
- ³ Institute of Gas Turbines and Aerospace Propulsion, Technical University Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany; schiffer@glr.tu-darmstadt.de
- * Correspondence: nicolas.lachenmaier@ps.rolls-royce.com
- This Paper is an Extended Version of Our Paper Published in the Proceedings of the 13th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, ETC13, Lausanne, Switzerland, 8–12 April 2019; Paper No. 263.

Received: 25 February 2020 ; Accepted: 17 June 2020; Published: 6 July 2020



Abstract: A turbocharger's radial turbine has a strong impact on the fuel consumption and transient response of internal combustion engines. This paper summarizes the efforts to design a new radial turbine aiming at high efficiency and low inertia by applying two different optimization techniques to a parametrized CAD model. The first workflow wraps 3D fluid and solid simulations within a meta-model assisted genetic algorithm to find an efficient turbine subjected to several constraints. In the next step, the chosen turbine is re-parametrized and fed into the second workflow which makes use of a gradient projection algorithm to further fine-tune the design. This requires the computation of gradients with respect to the CAD parametrization, which is done by calculating and combining surface sensitivities and design velocities. Both methods are applied successfully, i.e., the first delivers a well-performing turbine, which, by the second method, is further improved by 0.34% in efficiency.

Keywords: Large Diesel Engine; turbocharger; radial turbine; optimization; meta-model; adjoint sensitivity; efficiency; inertia

1. Introduction

The stationary and transient performance of large combustion engines is coined by the design of its turbocharger system, i.e., its compressor and turbine. In the present paper, a radial turbine is designed with a focus on both engine efficiency and transient response. These goals are directly linked to the turbine efficiency and its inertia. Furthermore, a high life expectancy is requested, which constrains the stress levels.

This problem is tackled with two different, subsequent design steps: The first is a gradient-free multidisciplinary optimization based on a meta-model assisted evolutionary algorithm. It can be considered standard in the sense that there are numerous publications that use comparable methods to improve rotor designs, see [1–3] or [4]. This method can explore the design space. However, its success is limited as the calculation effort scales with the number of design parameters, i.e., increasing the design space or, respectively, the potential for improvement comes with a great computational cost.

To overcome this major drawback, the global optimization in the first step with a gradient-based optimization in a second step. Additionally, the gradients in the second step are computed following an adjoint technique, which has the advantage that the effort per iteration does not scale with the

number of parameters and promises fast design improvements. However, differentiating every available cost function with high accuracy and subsuming all of them in a single optimization is challenging—which is probably why only few authors have presented comparable multidisciplinary design chains. However, there are some, and the potential for design improvements has proven to be considerable, see [5] or [6]. Setting up a multidisciplinary gradient-based design chain, one faces several difficulties. First, the designer needs to have tools available that can calculate mesh sensitivities for the considered cost functions. By now, several open-source and commercial CFD codes are available that offer methods to calculate the flow-related ones using adjoint methods. However, finding codes that can do this for the rest of the considered cost functions (eigenfrequency, mass, inertia) proves troublesome. Hence, in this paper, we apply a combination of commercial codes, open-source academic codes as well as in-house developed programs. Second, one needs to find a way to calculate design velocities, which is compatible with the CAD tool in which the turbine is parametrized. Within this work, a finite difference approach is applied to achieve this. Third, data must be mapped in between meshes, so interpolation methods need to be applied, such as the implemented nearest neighbor algorithm in the presented workflow. Finally, an optimizer is required that handles the different tools and steers the overall process, like the in-house implementation of the gradient projection algorithm according to [7].

In the following, both optimization methodologies will be presented: In the next chapter, the gradient-free framework is detailed, and a turbine is designed. In the subsequent chapter, this turbine is further improved by applying the gradient-based workflow.

2. Gradient-Free Optimization

Figure 1 visualizes the gradient-free workflow, which will be explained in this first part of the paper. Initially, a fluid and a solid geometry are created and meshed based on a parametrized CAD model and a parameter set *s*. The simulation is performed in three steps: First, an adiabatic CFD (Computational Fluid Dynamics) simulation is conducted to calculate the efficiency η . Second, the solid mesh is activated to facilitate a CHT (Conjugate Heat Transfer) calculation that is initialized with the aforementioned CFD flow field. Finally, the CSM (Computational Structural Mechanics) is run, i.e., the temperature field in the solid, the surface pressure distribution and the rotation rate n_{max} are used to calculate both the eigenfrequencies and the stresses. Furthermore, the rotor mass *m*, the center of gravity z_g , and the inertia J_z are calculated. The workflow is wrapped within an optimization software that implements a meta-model assisted genetic algorithm described and detailed at the end of this chapter.



Figure 1. Gradient-free setup.

2.1. Geometry Parametrization

The CAD geometry is built using the 3D modeling software CAESES®. The blade is modeled in three steps:

1. A meridional view of the blade is created using B-Splines, including the leading edge, trailing edge, hub, and shroud contour, see Figure 2. In the same step, the turbine back and outlet are designed.

2. The blade camber surface is created based on the meridional surface by specifying a θ -distribution at hub, Figure 3, which is controlled via inlet angle β_{in} and outlet angle β_{out} plus two weighting points (red). The θ -distribution solely depends on the axial position z, and describes the peripheral angle. Consequently, the position x of every point on the camber surface is fully defined by its radius r and its axial position z (x := x (r, $\theta(z)$)). The major benefit of this camber description is that it leads to radially fibered blades, Figure 4, that prevent bending stresses within the blades due to centrifugal forces.

3. Two splines are used to define a thickness distribution at both hub and shroud, Figure 3. A third spline is used to control the blending of the spanwise thickness.



Figure 3. Thickness and θ -distribution.



Figure 4. Radially fibered blades.

The model includes several further important features, Figure 5: Internally, the largest possible fillet radius is calculated based on the smallest distance of two adjacent blades. The latter mostly depends on the chosen number of blades and its hub thickness. This result is used to design the variable fillet connection of blade and hub geometry and is automatically chosen as large as possible to minimize the stresses in the fillets. The model also includes parameters to control the scallops. They are used to steer the inertia of the rotor and the stresses at the turbine rear side. The fluid geometry includes the gap between the heat shield and the rotor. The shroud tip gap can be defined, and both length and diameter of the diffuser can be set. However, these parameters are fixed during the optimization. In total, $n_{gf} = 42$ parameters, including the number of blades, are varied during optimization, see Table 1.



Figure 5. Solid (l.) and fluid (r.) body.

Eventually, the heat transfer calculations are substantially simplified as the fluid and the solid body share the same periodic surfaces, i.e., the fluid-solid-interface has a perfect overlap, see Figure 6.

Model Part	Parameters	Gradient-Free Setup	Gradient-Based Setup
	Leading edge	3	3
	Trailing edge	2	4
Meridional contour	Shroud contour	4	4
	Hub contour	4	4
	Axial length	1	1
Camber	θ -curve	5	9
	At hub	8	8
	At shroud	4	4
Thickness	Hub-to-shroud distribution	4	4
	Leading edge ellipticity	0	0
	Trailing edge ellipticity	3	3
Scallops		3	3
Number of blades		1	0
Sum		42	47

Table 1. Variable parameters for gradient-free and gradien	it-based	1 setur
---	----------	---------



Figure 6. Mesh for CFD and CHT.

2.2. Meshes

The solid geometry for CSM is meshed with ALTAIR SIMLABTM producing roughly 80.000 tetrahedral second-order elements. Table 2 presents a mesh study with three different meshes. A Richardson extrapolation [8] is used to calculate an estimate for the asymptotic values ϕ_{ext} of the eigenfrequency f_e and von Mises stresses σ . The relative extrapolated error

$$e_{ext} = \left| \left(\phi_{fine} - \phi_{ext} \right) / \phi_{ext} \right| \tag{1}$$

shows that a high uncertainty of σ_{fillet} must be expected. This observation is not surprising as small radii lead to high stress concentrations and therefore require many elements.

The fluid geometry for CFD and the solid geometry for CHT are meshed with unstructured polyhedrals using the STAR-CCM+ meshing capabilities. Prism layers with a stretching of 1.2 are used to resolve the boundary layer, which maintains $y^+ \approx 1$ and leads to approximately 1 million cells

in total. The flow outlet is elongated with a mesh extrusion. Table 3 summarizes a mesh study that once more uses a Richardson extrapolation to get an estimate for the asymptotic solution. The relative extrapolated error is considered reasonably small.

Even though block-structured meshes offer a higher ratio of accuracy to computational cost, unstructured meshes are preferred here, since they offer the flexibility to mesh the fillets, scallops, and heatshield gap. Furthermore, a node-conformal mesh at the interface between solid and fluid can be created, Figure 6. This facilitates a robust simulation of the turbine temperature field.

Table 2. Mesh study on CSM mesh.

# Elements	f_e [Hz]	σ_{fillet} [MPa]	σ_{back} [MPa]
19051	6179.0	466.1	458.1
36419	6155.4	478.0	459.2
ϕ_{fine} , 83585	6141.0	489.2	462.0
φext	6130.0	516.7	466.0
e_{ext} in %	0.18	5.31	0.85

# Cells	$p_{t,in}$ [Pa]	$T_{t,out}$ [K]	η_{ts}
230325	320,590.0	620.1	0.8349
572710	320,475.0	617.9	0.8451
ϕ_{fine} , 1089061	320,282.6	617.2	0.8460
ϕ_{ext}	320,194.7	616.3	0.8463
<i>e</i> _{ext} in %	0.03	0.14	0.03

 Table 3. Mesh study on CFD mesh.

2.3. Simulation Setup

After the meshes have been generated, an adiabatic CFD calculation in STAR-CCM+ is conducted. The boundary conditions are set as follows: The inlet mass flow \dot{m} and the back pressure $p_{s,out}$ are fixed. In absence of a volute, an inlet flow angle α_{in} has to be set and is adjusted during simulation to assure that the turbine is delivering the target power $P_T = \dot{m}\Delta h_t$. The angle is increased when the current power is too high and vice versa. This way, $p_{t,in}$, which is connected to the turbine inlet temperature $T_{t,in}$ in piston engines, is not known a priori. A simple, linear way ([9], p. 30) is used to model this dependency to set the inlet temperature dynamically

$$T_{t,in} := T_{t,in}(p_{t,in}) = T_Z \left[1 - \frac{\kappa - 1}{\kappa} \cdot \left(1 - \frac{p_{t,in}}{p_Z} \right) \right], \tag{2}$$

where T_Z and p_Z are in-cylinder temperature and pressure at valve opening. The flow is assumed fully turbulent. The Reynolds number based on the inlet width *b* is $Re \approx 2.2 \cdot 10^5$, which is why a turbulent velocity profile is set at the inlet, i.e.,

$$v(r) = v_{max} \cdot (1 - 2\frac{r}{b})^{1/8},$$
(3)

with *r* as the distance from the mid of the inlet channel. The *k*- ω -*SST*-model is used for closure of the RANS equations.

After convergence of the CFD simulation, the polyhedral mesh of the solid region is activated, Figure 7. A constant temperature T_{shaft} is set at the shaft. Every other wall is assumed adiabatic, i.e., the shaft is the only heat sink and radiation is neglected. As this CHT simulation starts from a converged flow solution, it converges reasonably fast: $\frac{\text{Runtime CHT}}{\text{Runtime adiabatic CFD}} \approx 0.8$. The temperature field is mapped onto the tetrahedral mesh as depicted in Figure 7. Additionally, the surface pressure distribution is mapped onto the surface of the tetrahedral mesh.



Figure 7. Tetrahedrals (CSM, left), polyhedrals (CHT, right).

Subsequently, the von Mises stresses within the rotor are calculated with the open-source finite element code CALCULIX [10] based on the maximum rotation rate n_{max} , including both the previously calculated temperature and pressure field. While the temperature has a significant impact on the stresses, especially in the turbine back, the latter is almost negligible as it just slightly increases the stresses at the rotor-shaft-connection. The first blade eigenfrequency f_e is calculated based on this pre-stressed state.

Lastly, inertia J_z , mass m, and center of gravity z_g are calculated. Gauss's theorem is used, e.g.,

$$J_z = \rho \int_V (x^2 + y^2) \, \mathrm{d}v = \rho \int_S \left(x^3/3, y^3/3, 0 \right)^\top \cdot n \, \mathrm{d}a, \tag{4}$$

to calculate these quantities based on a tesselated STL file of the solid body surface.

The respective simulation runtimes are reported in Table 4.

Simulation Step	Relative Wall Clock Time	#CPUs
Geometry generation	0.04	1
Meshing	0.34	2
CFD	0.29	24
CHT	0.23	24
CSM	0.10	12
Total runtime	1.00	
	(47 min)	

Table 4. Wall clock time within gradient-free setup.

2.4. Objectives and Constraints

Two objective values are maximized during the optimization: The total-to-static efficiency and the total-to-total efficiency, Equation (5):

$$\eta_{ts} = \frac{T_{t,in} - T_{t,out}}{T_{t,in} \cdot \left(1 - \left(\frac{p_{s,out}}{p_{t,in}}\right)^{\frac{\kappa-1}{\kappa}}\right)}, \qquad \qquad \eta_{tt} = \frac{T_{t,in} - T_{t,out}}{T_{t,in} \cdot \left(1 - \left(\frac{p_{t,out}}{p_{t,in}}\right)^{\frac{\kappa-1}{\kappa}}\right)}.$$
(5)

These cost functions have a high correlation, yet, it is not one, see the final database in Figure 8. Maximizing η_{ts} is equivalent to assuming the exit kinetic energy after diffuser being a loss. However, within modern piston engines the radial turbine is not the last geometry in the exhaust piping system, i.e., there usually will be an exhaust after treatment (EAT). Consequently, from a system perspective, the stagnation pressure between turbine diffuser exit and EAT is important. Summarized, assuming having the choice between two designs with the same η_{ts} , then deciding for the one with the higher η_{tt} is reasonable.



Figure 8. Comparison of η_{ts} and η_{tt} .

Apart from these objectives, several constraints are considered: The first eigenfrequency of the blade

$$f_e \ge s_{margin} \cdot \frac{n_{EO} \cdot 60\frac{rpm}{s}}{n_{max}} \tag{6}$$

is constrained to lie above a specific engine order $n_{EO} \in \mathbb{N}$ (+ safety margin), which is an empirically chosen value. This constraint is set to prevent high cycle fatigue damages during operation.

The inertia J_z of the turbine is constrained as it affects the turbocharger's transient performance as a lower inertia leads to faster rotor acceleration. In that regard, the compressor inertia is not equally significant as its density is substantially lower due to different materials: $3 \cdot \rho_{Aluminum} \approx \rho_{Inconel}$.

The rotor mass *m* is constrained as it slightly affects the manufacturing costs, but more importantly, influences the rotor dynamics as lower masses lead to smaller rotation orbits assuming comparable eccentricity and center of gravity. As such, it prevents the rotor from rubbing against the housing during operation even for small tip gaps. For the same reason, also the axial position of the center of gravity z_g is constrained.

The von Mises stresses σ are constrained at two different surfaces where peaks are expected: On the fillet surface between two adjacent blades and at the turbine rear side, see Figure 5. The allowable stress is set well below the materials yield strength σ_y with a large safety margin as the turbine is designed to withstand a high number of load cycles. Furthermore, the stresses are evaluated in an integral sense as in [5], i.e.,

$$\sigma_{S} = \sqrt[p]{\frac{1}{|S|} \int_{S} \sigma^{p} \, \mathrm{d}a}, \quad p = 10, \quad S \in \{\mathrm{fillet}, \mathrm{back}\}.$$
(7)

This formulation has a smoothing effect on the non-differentiable $\max{\{\cdot\}}$ -evaluation. Consequently, it the quality of the applied meta-models and is less prone to mesh induced noise in the results. The optimization problem is summarized as follows:

maximize
$$\eta_{ts}$$
 and η_{tt} such that
$$\begin{cases} f_e > c_1 & z_g < c_4 \\ J_z < c_2 & \sigma_{fillet} < c_5 \\ m < c_3 & \sigma_{back} < c_5 \end{cases}$$
(8)

2.5. Optimization Results

Based on the previously described workflow, the automated design evaluation is implemented within the optimization software MODEFRONTIER. To sample the design space, first, a design of experiments is set up. A uniform Latin hypercube is used that delivers 395 successful design evaluations. Upon this database, the optimization is started exploiting one of the software's implemented meta-model assisted multi-objective genetic algorithm (GA) called FAST MOGA II [11]. During the optimization, the software sets up four meta-models (polynomial SVDs, neural networks, radial basis functions, kriging) per output function and automatically chooses the most successful one for every function by comparing their prediction to the evaluated designs of the newest generation of designs. 'Success' is determined by using a mean error. Afterwards, the new generation is added to the meta-model training base and the next generation is started. Another 1302 designs are assessed this way. 1697 designs are evaluated in total with 1105 being infeasible and violating at least one of the constraints, see Figure 9. The design with the highest efficiency η_{tt} fulfilling all constraints is picked and prepared for the subsequent design step, described in the next chapter. Figures 4–7 show the chosen design. This design, coincidentally, is the one with the highest total-to-static efficiency η_{ts} , i.e., there is just one rank 1 Pareto design. Evaluating 1697 designs serially on 24 CPUs would have taken approximately 1300 h (see Table 4), so this process is very costly. This is driven by both the runtime per design and the 42 free parameters. Reducing the latter may drastically reduce the optimization runtime. However, deciding which parameters to render down is a difficult task on its own, which may either be guided by the designer's experience or a prior sensitivity study. At last, this motivates the workflow presented in the next chapter as the number of free parameters has a marginal impact on the optimization runtime in the gradient-free setup.



Figure 9. Optimization results.

3. Gradient-Based Optimization

Figure 10 summarizes the process used to drive a gradient-based optimization of the radial turbine. The central idea is to locally approximate each response function f_i linearly as

$$f_j(\boldsymbol{s} + \Delta \boldsymbol{s}) \approx f_j(\boldsymbol{s}) + \Delta \boldsymbol{s} \cdot \frac{\partial f_j}{\partial \boldsymbol{s}}.$$
 (9)

A gradient projection algorithm is applied to drive an optimization aimed at raising the efficiency η_{tt} while satisfying several constraints. Calculating $\frac{\partial f_j}{\partial s}$ for every response function in a reasonable timeframe is, however, not straightforward:

Using finite differences is too costly as it would require n_{gb} design evaluations to approximate $\frac{\partial f_i}{\partial s}$. Hence, the derivative is split via the chain rule

$$\frac{\partial f_j}{\partial s} = \frac{\partial f_j}{\partial X} \cdot \frac{\partial X}{\partial s},\tag{10}$$

where *X* refers to the surface mesh node positions and $\frac{\partial f_j}{\partial X}$ is computed in a single simulation per response function. Still, n_{gb} evaluations are necessary to approximate $\frac{\partial X}{\partial s}$; however, these operations are cheap. Both steps are detailed below.

Unfortunately, not all response functions handled in the gradient-free framework are available in this gradient-based setting. In particular, the calculation of the stress sensitivity $\frac{\partial \sigma}{\partial X}$ has not been included yet. One way to implement this has been presented by [12].



Figure 10. Gradient-based optimization workflow.

3.1. Altered Geometry Parametrization and Simulation Setup

To investigate whether the method is capable of further improving the design, the geometry is re-parametrized, i.e., two splines are exchanged to add further degrees of freedom. The two chosen splines describe the turbine trailing edge, and the θ -distribution, Figures 2 and 3, sketched in red (old) and green (new). In total, it contains $n_{gb} = 47$ parameters (reminder: $n_{gf} = 42$). As the number of blades is a discontinuous parameter, it needs to be set constant.

The CFD simulation setup is adapted for this optimization: $T_{t,in}$ and α_{in} remain fixed and a stagnation pressure at inlet is used, i.e., $p_{t,in}$ is set at the inlet. As *in* might change as the design changes with these boundary conditions, it is now handled as a constraint.

The runtime of this setup changes compared to the gradient-free setup: The primal CFD run now takes approximately twice as long as going down to machine precision is advisable in the context of the subsequent adjoint calculation. The adjoint run takes approximately three times as long as the primal

run accounting for two cost functions and the chosen GMRES solver. However, neither a temperature field nor stresses are calculated in this gradient-based setup.

3.2. Design Velocity

Tesselated surface descriptions, i.e., STL files, Figure 11, are used to calculate the design velocities $\frac{\partial X}{\partial s_i}$ with a first order finite differencing scheme, where X are the centers of the STL triangles and s_i , $i \in \{1, ..., n_{gb}\}$ is a CAD parameter. To calculate the sensitivity, two geometries are necessary: The baseline geometry X based on the parameter set $s = \{s_1, ..., s_{n_{gb}}\}$ and X_i , which is varied in one parameter $\{s_1, ..., s_{i-1}, s_i + h_i, s_{i+1}, ..., s_{n_{gb}}\}$ by a small distance h_i . The derivative is calculated pointwise

$$\frac{\partial x}{\partial s_i} \approx \frac{x_i - x}{h_i},\tag{11}$$

where $x \in X$, $x_i \in X_i$. Care must be taken as to which x_i the point x is compared to for the difference calculation in Equation (11). The nearest neighbor algorithm from PYTHONS SCIPY library is used to find the point x_i which is located closest to x. The design velocity is calculated as in Equation (11) (depicted in blue in Figure 12).



Figure 11. Tesselated surface mesh.



Figure 12. Design velocity calculation.

3.3. Surface Sensitivity

The surface mesh sensitivity $\frac{\partial f_j}{\partial X}$ must be determined for every response function f_j . The applied methods differ for the various responses:

The mesh sensitivities of both η_{tt} and m are determined with the adjoint method, implemented in STAR-CCM+, which allows for an efficient calculation of the gradients. Applying this method requires the solution of one additional system of equations per response, which are solved with a Krylov subspace scheme. The code implements the discrete adjoint approach, i.e., the derivation of the adjoint equations is based on the discretized Navier–Stokes Equations. The frozen turbulence assumption is used, i.e., the turbulent viscosity is assumed constant during the adjoint simulation. After calculation of the volume mesh sensitivities, only the sensitivities on the surface are used in the optimization. The values of the inner nodes are assumed negligible. Consequently, the sensitivity of the volume mesh regarding to the surface mesh neglected. For a general outline on this and related methods, see [13]. The total-to-static efficiency η_{ts} is no longer regarded: Gradient-based schemes in a multidisciplinary setting require compromise functions that weight individual objectives against each other, e.g., a weighted linear combination $a_1 \cdot \eta_{ts} + a_2 \cdot \eta_{tt}$. In the following, however, we neglect η_{ts} s.t. $(a_1, a_2) = (0, 1)$.

The first eigenfrequency f_e and its mesh sensitivity are calculated with the open-source code KRATOS [14]. After the eigenvalue $\lambda_e = (2\pi f_e)^2$ and its eigenvector Φ_e have been determined, the calculation of the sensitivity with respect to a mesh node position $x \in X$ can be found by differentiating the undamped eigenvalue equation to get

$$\frac{\partial \lambda_e}{\partial \mathbf{x}} = \Phi_e^\top \left(-\lambda_e \frac{\partial M}{\partial \mathbf{x}} + \frac{\partial K}{\partial \mathbf{x}} \right) \Phi_e, \tag{12}$$

where *M*, *K* are the finite element mass and stiffness matrix, i.e., no further linear system must be solved. In KRATOS, first order finite differences are used to determine both $\frac{\partial M}{\partial x}$ and $\frac{\partial K}{\partial x}$. The mesh sensitivity of nodes not lying on the surface is neglected and assumed zero. For a derivation of Equation (12), see e.g., [15].

The surface sensitivity of J_z , m, and z_g can be determined analytically based on the tesselated surface. Take a triangle of the triangulated surface and let Δa be its area. Assume this triangle being translated by Δs in surface normal direction n. Then, the surface sensitivity of e.g., the inertia J is

$$\frac{1}{\rho} \frac{\partial J_z}{\partial n} = \lim_{\Delta s \to 0} \frac{\int_{V = \Delta s \cdot \Delta a} (x^2 + y^2) \, \mathrm{d}v}{\Delta s} \approx \Delta a (\bar{x}^2 + \bar{y}^2) \tag{13}$$

In total, the optimization problem may be summarized as follows:

maximize
$$\eta_{tt}$$
 such that
$$\begin{cases} f_e > f_{e,1} \\ J_z < J_{z,1} \\ m < m_1 \\ z_g < z_{g,1} \end{cases}$$
 (14)

Please note that the subscript 1 refers to the values of design iteration 1, i.e., the chosen design at the end of the gradient-free optimization.

3.4. Gradient Calculation

Figure 13 exemplarily summarizes the steps described above. On the left, the surface sensitivity of the eigenfrequency f_e is shown. In the middle, the design velocity of one parameter s_i describing the blade thickness at shroud is presented. On the right, the pointwise product of these two is visualized, which is calculated using another nearest neighbor search. Integration along the surface finally delivers $\frac{\partial f_j}{\partial s_i}$.



Figure 13. Eigenfrequency sensitivity $\left\|\frac{\partial f_e}{\partial x}\right\|$ (left), design velocity $\left\|\frac{\partial x}{\partial s_i}\right\|$ (mid) and pointwise product $\left\|\frac{\partial f_e}{\partial x} \cdot \frac{\partial x}{\partial s_i}\right\|$ (right).

3.5. Optimization Results

The gradient-based optimization is driven with a gradient projection algorithm, which facilitates the handling of both inequality and equality constraints, see [7], ch. 5: First, the gradient of the efficiency η_{tt} is orthogonally mapped into the subspace of all active, linearized constraints to calculate the projected descent direction. Equality constraints, such as the mass flow constraint, are always considered active, while inequality constraints including parameter boundaries are inactive as long as they are not violated. The second step is a damped correction step that copes with the difference between the target value and the actual value of the respective constraint. Finally, the chosen descent step is a linear combination of these two, i.e., projection and correction step.

The success of the optimization is tightly linked to the quality of the gradient approximation. Hence, a study based on forward differences is conducted to judge whether the presented chain of methods exploiting surface sensitivities and design velocities (SS+DV) delivers reasonable derivatives. However, determining a gradient via finite differences (FD) is expensive as the baseline design plus $n_{gb} = 47$ variations thereof must be evaluated to facilitate a finite difference gradient approximation. A comparison for a few cost functions is presented in Figure 14, where the dotted lines show the FD gradients, and the solid lines represent the derivatives based on the SS+DV approach for all $n_{gb} = 47$ parameters. The gradients $\frac{\partial J_z}{\partial s}$, $\frac{\partial f_e}{\partial s}$ and $\frac{\partial m}{\partial s}$ match reasonably with only a few deviating parameters.



Figure 14. $\frac{\partial f_e}{\partial s}$, $\frac{\partial J_z}{\partial s}$, $\frac{\partial m}{\partial s}$ each calculated both via finite differences (FD) and via surface sensitivities + design velocities (SS+DV).

The optimization runs successfully, see Figure 15. Within five iterations, the optimization stagnates. At this point, an increase in η_{tt} by 0.34% can be observed. Both the initial design and the design from iteration 8 are shown in Figure 16 for comparison. Figure 16 shows that the major design change lies in the θ -distribution. In iteration 4, the eigenfrequency f_e drops below its constraint value, and the projected descent algorithm starts its correction steps. However, m drifts away from its target value. This behavior indicates either imprecise gradients or a too strongly damped correction. The cost functions inertia J_z , center of gravity z_g , and rotor mass m hardly change throughout the optimization run, see Figure 15. Even though the stresses σ_{rear} and σ_{fillet} have no longer been evaluated for the gradient-based setup, the minimal changes in geometry suggest that no major increase in stress has to be expected. Furthermore, it is worth mentioning that the total-to-static η_{ts} efficiency is hardly raised (~0.1%). This result indicates that the change in η_{tt} is partly linked to an increased kinetic energy at the turbine outlet, i.e., the swirl is higher.



Figure 16. gradient-free design (iteration 1, red) and gradient-based design (iteration 8, green).

4. Conclusions

Two workflows to optimize a radial turbine with different advantages and disadvantages have been presented. They have shown to complement each other well: The gradient-free setup can be used to sample the design space and find a reasonable turbine design. Furthermore, it offers the possibility to handle discrete variables such as the number of blades. Yet, the designer is obliged to limit the number of free variables as computational cost becomes an issue, known as the curse of dimensionality. The gradient-based process allows for an increase in degrees of freedom without majorly affecting the computational costs. In fact, it is negligible compared to the costs that come along with the gradient-free setup. Within this study, the runtime of the gradient-based setup turned out to be two orders of magnitude lower: The runtime is driven by the primal and adjoint CFD simulations, which are more costly than in the gradient-free setup as mentioned earlier. Neglecting the CHT calculations, we estimate a ratio of

$$\frac{\text{Primal and adjoint runs in gradient-based workflow}}{\text{CFD runs in gradient-free workflow}} = \frac{(2+3) \cdot 10}{1697} \approx 0.029$$

for this specific case. However, one must keep in mind that the gradient-free workflow started from scratch while the gradient-based workflow was initiated from an optimized design. Hence, no general conclusions on the runtimes of the workflows may be drawn.

Future work will focus on three aspects: First, the gradient-free setup will be supplemented with life-expectancy calculations that necessitate prior stress and temperature field simulations. Second, the gradient-based setup will be complimented with stress sensitivities to allow for a more holistic and comparable design chain. Third, the gradient quality of the flow-related sensitivities will be improved by dropping the frozen turbulence assumption and introducing a more sophisticated derivative of volume mesh regarding the surface mesh.

Author Contributions: Conceptualization, N.L.; funding acquisition, H.-P.S. and J.K.; investigation, N.L.; methodology, N.L. and D.B.; project administration, N.L.; supervision, H.-P.S. and J.K.; visualization, N.L.; writing, original draft, N.L.; writing, review and editing, N.L., D.B., H.-P.S. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the financial support by the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) in the project GAMMA (project number 03ET1469a). The APC of this paper was funded by Euroturbo.

Acknowledgments: We gratefully thank MTU Friedrichshafen GmbH for the permission to publish this paper and Friendship Systems for their support in building the CAD geometry.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Latin

e _{ext}	Extrapolated relative error $[-]$
f _e	Eigenfrequency $[1/s]$
f_i	Cost function $j[-]$
h	Specific enthalpy $[J/kg]$
J	Inertia $[kg \cdot m^2]$
Κ	Stiffness matrix [-]
m	Mass flow $[kg/s]$
т	Rotor mass [kg]
Μ	Mass matrix [–]
п	Rotation rate [<i>rpm</i>]
р	Pressure [Pa]
s	Parameter set $[-]$
Т	Temperature $[K]$
υ	Velocity $[m/s]$
z_g	Center of gravity $[m]$
Greek	
κ	Isentropic exponent [-]
η	Efficiency [-]
σ	Stress tensor [MPa]
Subscripts	
gb	gradient-based
gf	gradient-free
S	Static quantity
t	Stagnation quantity
Abbreviations	
CFD	Computational Fluid Dynamics
CHT	Conjugate Heat Transfer
CSM	Computational Structural Mechanics
GA	Genetic algorithm
RANS	Reynolds-averaged Navier-Stokes
STL	Standard Tesselation Language

References

- 1. Verstraete, T. Multidisciplinary Turbomachinery Component Optimization Considering Performance, Stress, and Internal Heat Transfer. Ph.D. Thesis, von Karman Institute for Fluid Dynamics—University of Gent, Gent, Belgium, 2008.
- 2. Müller, L.; Alsalihi, Z.; Verstraete, T. Multidisciplinary Optimization of a Turbocharger Radial Turbine. *J. Turbomach.* **2013**, *135*. [CrossRef]
- 3. Peter, J. Numerische Untersuchung und Optimierung des Laufrades einer Pkw-Abgasturboladerturbine. Ph.D. Thesis, Leibniz Universität Hannover, Hannover, Germany, 2016.
- 4. Khairuddin, U.B.; Costall, A.W. Aerodynamic Optimization of the High Pressure Turbine and Interstage Duct in a Two-Stage Air System for a Heavy-Duty Diesel Engine. *J. Eng. Gas Turbines Power* **2018**, 140. [CrossRef]
- Verstraete, T.; Müller, L.; Müller, J.D. Multidisciplinary Adjoint Optimization of Turbomachinery Components Including Aerodynamic and Stress Performance. In Proceedings of the 35th AIAA Applied Aerodynamics Conference, Denver, CO, USA, 5–9 June 2017.
- 6. Schwalbach, M.; Verstraete, T.; Müller, L.; Gauger, N. CAD-Based adjoint multidisciplinary optimization of a radial turbine under structural constraints. In Proceedings of the Montreal 2018 Global Power and Propulsion Forum, Zug, Switzerland, 7–9 May 2018.
- 7. Haftka, R.T.; Gürdal, Z.; Kamat, M. *Elements of Structural Optimization (Solid Mechanics and Its Applications)*; Springer: Berlin/Heidelberg, Germany, 2013.
- 8. Celik, I.B.; Ghia, U.; Roache, P.J.; Freitas, C.J.; Coleman, H.; Raad, P.E. Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications. *J. Fluids Eng.* **2008**, *130*. [CrossRef]
- 9. Pucher, H.; Zinner, K. Aufladung von Verbrennungsmotoren: Grundlagen, Berechnungen, Ausführungen (German Edition); Springer: Berlin/Heidelberg, Germany, 2012.
- 10. Dhondt, G.; Wittig, K. Calculix: A Free Software Three-Dimensional Structural Finite Element Program; MTU AeroEngines: Munich, Germany, 1998.
- 11. Montrone, T.; Turco, A.; Rigoni, E. *FAST Optimizers: General Description*; Esteco Technical report; Esteco: Trieste, Italy, 2014.
- 12. Verstraete, T.; Müller, L.; Müller, J.D. CAD-Based Adjoint Optimization of the Stresses in a Radial Turbine. In Proceedings of the ASME Turbo Expo 2017, Charlotte, NC, USA, 26–30 June 2017.
- 13. Peter, J.E.; Dwight, R.P. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Comput. Fluids* **2010**, *39*, 373–391. [CrossRef]
- 14. KRATOS-Multiphysics. Available online: https://github.com/KratosMultiphysics/Kratos (accessed on 30 September 2018).
- 15. Firl, M. Optimal Shape Design of Shell Structures. Ph.D. Thesis, Chair of Structural Mechanics—Technical University of Munich, Munich, Germany, 2010.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).