

Discrete Adjoint Gradient Evaluations for Linear Stress and Vibration Analysis

Marc Schwalbach*, Tom Verstraete*, Nicolas R. Gauger†

February 19, 2019

Abstract

This paper presents methods used to perform discrete adjoint gradient evaluations for linear stress and vibration analysis. The methods are implemented within the framework of a discrete adjoint structural solver being developed for multidisciplinary adjoint optimizations of turbomachinery components. The code is differentiated using the algorithmic differentiation (AD) tool CoDiPack in tandem with manual treatment of the iterative solvers.

Stress analysis leads to a linear system of equations that is typically solved by an iterative solver (e.g. GMRES). To ensure accuracy, the adjoint problem is formulated as a new linear system of equations to be solved.

Vibration analysis results in a generalized eigenvalue problem that is also typically solved by an iterative solver. The adjoint problem takes out the generalized eigenvalue solve and replaces it by one outer product per eigenfrequency, leading to significantly cheap eigenfrequency gradients for vibration analysis.

1 Introduction

State of the art adjoint optimization methods in the field of turbomachinery focus mainly on aerodynamic cost functions and constraints [26, 13, 25]. While such optimization methods can produce an aerodynamically optimized shape, the structural requirements may not be met. These are usually evaluated a posteriori rather than within the optimization process, leading to several re-iterations in practice. In order to take structural constraints into account and reduce design iterations, a multidisciplinary adjoint optimization framework for turbomachinery is introduced in section 2. To realize this, a discrete adjoint structural solver that is equipped to meet the needs of turbomachinery design

*von Karman Institute for Fluid Dynamics

†TU Kaiserslautern

is required.

Stress and vibration analyses play a significant role in evaluating the structural requirements of turbomachinery components. These components undergo high rotational speeds in operation, which can lead to large centrifugal forces and structural deformations. As a result of these deformations, the structural stresses increase and need to be kept under the specified safety limit. Additionally, unsteady pressure forces from the surrounding fluid contribute to undesired vibrations that can lead to structural failure. Evaluating the maximum von Mises stress in a static stress analysis and the eigenfrequencies using a vibration analysis, provides engineers with vital information to design structurally feasible components. For gradient-based optimization methods, the sensitivities of stress and vibrational quantities of interest with respect to design parameters are required. To compute the required sensitivities efficiently, an adjoint structural solver that offers stress and vibration analysis has been developed. While recent works [14, 9] have differentiated structural solvers for multidisciplinary optimizations (MDOs) using a more analytical approach, this work focuses on the efficient use of algorithmic differentiation (AD). The discrete adjoint implementations of the linear stress and vibration analysis are discussed in sections 3 and 4, respectively.

While the gradients of the stiffness and mass matrices can be derived analytically, one of the advantages of using AD for code differentiation is that additional code contributions are automatically differentiated. By using an object-oriented design approach, this provides a maintainable code base for future contributions. In section 5, an example is shown where additional contributions for composite material applications to the primal code are automatically differentiated and gradients for material design parameters can be easily obtained.

2 Multidisciplinary Adjoint Optimization

Performing an optimization in a single discipline, such as aerodynamics, neglects crucial constraints imposed by other disciplines, such as structural mechanics. Ideally, the structural constraints are considered during the optimization procedure, such that the resulting shape is aerodynamically optimized and satisfies the structural constraints. This section will briefly motivate the use of gradient-based optimization methods using adjoints and discuss what is required for a multidisciplinary adjoint framework.

Gradient-free optimization methods have been applied successfully in turbomachinery [17, 24, 16, 22, 10]. The methods are non-intrusive and one only needs to evaluate the cost function. Unfortunately, these methods suffer from the curse of dimensionality. This means that the cost of performing a gradient-free optimization is dependent on the size of the design space m , which can be significantly large. Convergence often requires a large amount of function

evaluations. Gradient-based methods, on the other hand, use the gradient of the cost function $J(\mathbf{x}) \in \mathbb{R}$ with respect to the design $\mathbf{x} \in \mathbb{R}^m$,

$$\frac{\partial J}{\partial \mathbf{x}} \in \mathbb{R}^m, \quad (1)$$

to converge towards an optimum with less iterations. However, these methods only converge towards the global optimum if the problem is convex and computing the sensitivities can be cumbersome.

The least intrusive way of approximating the gradient (1) is to use the finite differences (FD) method. However, the cost is dependent on the number of input variables m , costing $2m \cdot \text{cost}(J)$ when using a second order approximation. Here $\text{cost}(J)$ refers to the computational cost of evaluating the *primal model* $J(\mathbf{x})$. As the size of the design space increases, computing the gradients may become infeasible. The adjoint approach, which was first applied in aerodynamic optimization by Pironneau [19], followed by Jameson [7, 6], offers an efficient alternative for computing the gradients. Evaluating the *adjoint model*

$$\bar{\mathbf{x}} = \frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}^T \bar{J} \quad (2)$$

with $\bar{J} = 1$ computes the gradient (1) at a cost dependent on the number of outputs, rather than the number of inputs. The bar notation \bar{J} denotes an adjoint variable. Typically the number of outputs in an optimization is much lower than the number of inputs. Thus, the adjoint method is superior in terms of computational cost. Moreover, being independent of the number of inputs gives engineers a broader design space to work with.

Two approaches can be taken to implement the adjoint model. The *continuous adjoint* approach [8] involves deriving a continuous adjoint model, then discretizing it to solve numerically. This results in computing the gradients of the continuous problem. The *discrete adjoint* approach [15], on the other hand, derives an adjoint model from the discretized system of equations, and computes the gradients of the discretized problem. The source code transformation method *algorithmic differentiation* (AD) [4, 18], sometimes referred to as *automatic differentiation*, can be used to generate the adjoint model of an existing code. Several AD tools are available and are listed on the AD community's web page www.autodiff.org. In this work, the discrete adjoint approach is taken using the AD tool CoDiPack [20].

In the context of AD, evaluating the adjoint model (2) is also referred to as *reverse mode* and may be used interchangeably in this text. AD can also be used to differentiate a primal model $J(\mathbf{x})$ in *forward mode* using the chain rule of differentiation. This paper will concentrate on using AD to implement the adjoint, or reverse mode. For a more detailed description of algorithmic differentiation, the reader is referred to e.g. [4, 18].

In addition to the gradients (1), one requires gradients of constraints with respect to design parameters to formulate a gradient-based multidisciplinary optimization problem. For turbomachinery applications, structural constraints are of particular interest. Previous development efforts of implementing the structural constraints into this multidisciplinary optimization framework include an adjoint CAD transformation for cold-to-hot deformations [21]. To complement this, the gradients of structural quantities of interest, such as the maximum von Mises stress and eigenfrequencies, with respect to design parameters are required. The discrete adjoint models that compute these gradients are the focus of this paper. They are introduced in the next sections 3 and 4.

3 Discrete Adjoint Linear Stress Analysis

The structural linear stress analysis is used to assess the structural integrity based on given forces. In the case of turbomachinery, relevant forces are composed primarily of the centrifugal forces due to high rotational speeds. Additionally, pressure forces from the surrounding fluid may be considered as well. The structural solver discussed in this paper is discretized using the finite element method (FEM). For a detailed discussion on the FEM approach for structural applications, the interested reader is referred to [2].

3.1 Linear Stress Equations

Using the weak formulation of the balance of momentum, the finite element formulation for linear stress analysis can be derived [2] as

$$K\mathbf{u} + M\ddot{\mathbf{u}} = \mathbf{b}, \quad (3)$$

with the stiffness matrix $K \in \mathbb{R}^{m \times m}$, mass matrix $M \in \mathbb{R}^{m \times m}$, load vector $\mathbf{b} \in \mathbb{R}^m$, and displacements $\mathbf{u} \in \mathbb{R}^m$. For a static linear stress analysis, $M\ddot{\mathbf{u}} = 0$. One is left with a linear system of the form

$$K(\mathbf{x})\mathbf{u} = \mathbf{b}(\mathbf{x}), \quad (4)$$

where the stiffness matrix and the load vector depend on the design variables $\mathbf{x} \in \mathbb{R}^m$. The linear system is solved for the displacements which are then used to compute a structural cost function, such as the maximum von Mises stress $\sigma_{max}(\mathbf{u}) \in \mathbb{R}$. Note that the maximum is computed using a continuous function, e.g. the Kreisselmeier-Steinhauser function [11]

$$\sigma_{max} = KS(\sigma) = -\frac{1}{\rho} \ln \left(\sum_i^n e^{-\rho\sigma_i} \right) \quad (5)$$

or the p -norm

$$\sigma_{max} = \|\sigma\|_p = (|\sigma_1|^p + |\sigma_2|^p + \dots + |\sigma_n|^p)^{\frac{1}{p}}. \quad (6)$$

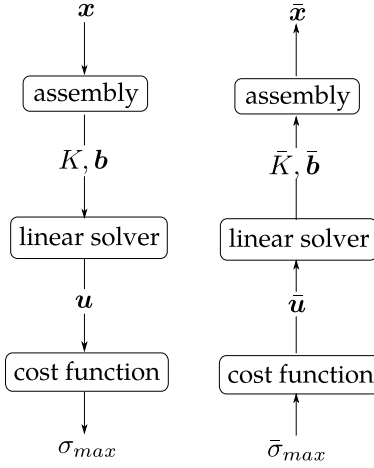


Figure 1: Linear stress analysis algorithm breakdown. Primal(l) and adjoint(r)

For the inclusion of structural constraints in a gradient-based multidisciplinary optimization, the sensitivities

$$\frac{\partial \sigma_{max}}{\partial \mathbf{x}} \in \mathbb{R}^m \quad (7)$$

have to be computed.

3.2 Adjoint Implementation

The implementation of the discrete adjoint linear stress analysis can be broken down into a three-step algorithm (figure 1). First, the linear system is assembled by computing the stiffness matrix K and the load vector \mathbf{b} . Then, the linear system is solved for the displacement field \mathbf{u} . Finally, the displacement results are used to evaluate the cost function σ_{max} . Note that here the mentioned cost function is σ_{max} , yet this can actually be any cost function that is dependent on the displacements \mathbf{u} .

The adjoint model has a similar three step form, only in reverse (figure 1). Seeding the adjoint model

$$\bar{\mathbf{x}} = \frac{\partial \sigma_{max}}{\partial \mathbf{x}}^T \bar{\sigma}_{max} \quad (8)$$

with $\bar{\sigma}_{max} = 1$ and performing a reverse run will compute the sensitivities

$$\frac{\partial \sigma_{max}}{\partial \mathbf{x}} \in \mathbb{R}^m.$$

However, a black-box differentiation using AD would involve differentiating through the iterative linear solver of step 2 in figure 1. This could result in a significantly high memory consumption and computational cost proportional to the number of iterations. Additionally, the same number of iterations and the same basis for the Krylov subspace used by the primal linear solver would be used to compute the derivatives. At that point, the adjoint variables may not actually be converged and inaccurate derivatives would be propagated through the calculation.

Alternatively, the adjoint of the linear system can be treated explicitly as described in [23]. This method would result in the equations

$$\bar{\mathbf{b}} = K^{-T} \bar{\mathbf{u}}, \quad \bar{K}_{i,j} = -\mathbf{u}_j \bar{\mathbf{b}}_i \quad (9)$$

for the adjoints of K and \mathbf{b} . Due to the symmetric positive definite structure of the stiffness matrix for linear stress analyses, an additional simplification can be made such that

$$K \bar{\mathbf{b}} = \bar{\mathbf{u}}. \quad (10)$$

As a result, computing the adjoints of step 2 involves an additional linear system solve with the same matrix K . Recording the operations of the iterative linear solver with AD is no longer necessary. The vector $\bar{\mathbf{u}}$ comes from the adjoint of the cost function evaluation (step 1 of figure 1), which is differentiated using AD. After \bar{K} and $\bar{\mathbf{b}}$ have been computed using (9), they are plugged into the adjoint linear system assembly (step 3 of figure 1) to compute $\bar{\mathbf{x}}$.

3.3 Accuracy and Performance Results

To assess the accuracy of the discussed adjoint formulation, the gradient (7) is computed using a rotating cantilever beam test case with 270 degrees of freedom (figure 2). The beam has fixed boundaries on the left hand edge (figure 2) and concentrated loads on the opposite edge, where the displacements are most prominent. The sensitivities are visualized in figure 3 and plotted in figure 4. The gradient computed using the adjoint method is in good agreement with the gradient computed using finite differences and a forward AD differentiation.

An axial fan geometry, which has been used as a baseline for an optimization in [1], is used to perform run time and memory consumption tests. Run time and memory consumption results using the axial fan test case are shown in table 1. The results reflect the computational effort and memory requirements for computing the gradient (7). A small test case with approximately 30,000 degrees of freedom and a larger test case with approximately 200,000 degrees of freedom (figure 5) are compared, using the **RealReverse** AD type of CoDiPack. The resulting sensitivities are shown in figure 6. The larger test case shows better run time and memory ratios, reaching a computational run time of around 2.1 times the primal run time. This reflects the fact that two linear systems are solved using the same matrix K and solving the linear system accounts for

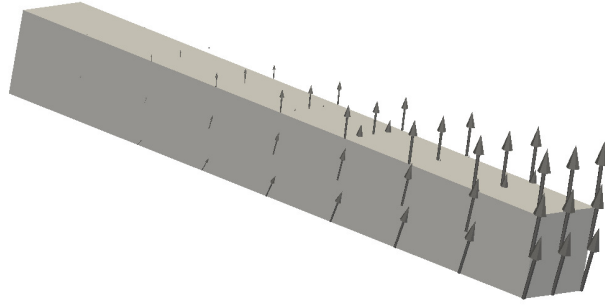


Figure 2: Rotating cantilever beam with displacement vectors

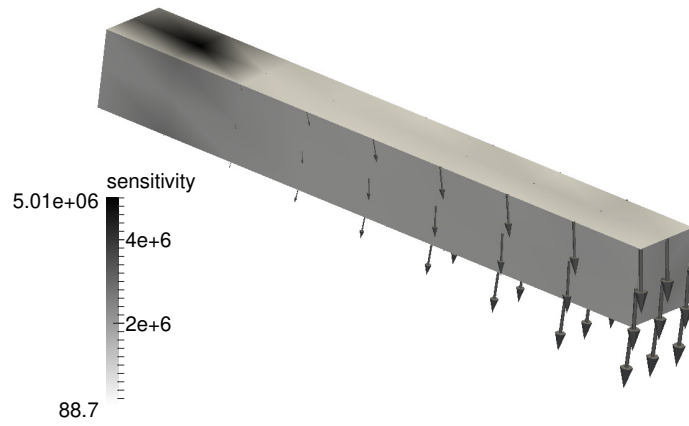


Figure 3: Rotating cantilever beam with sensitivities $|\frac{\partial \sigma_{max}}{\partial \mathbf{x}}| \in \mathbb{R}^{90}$

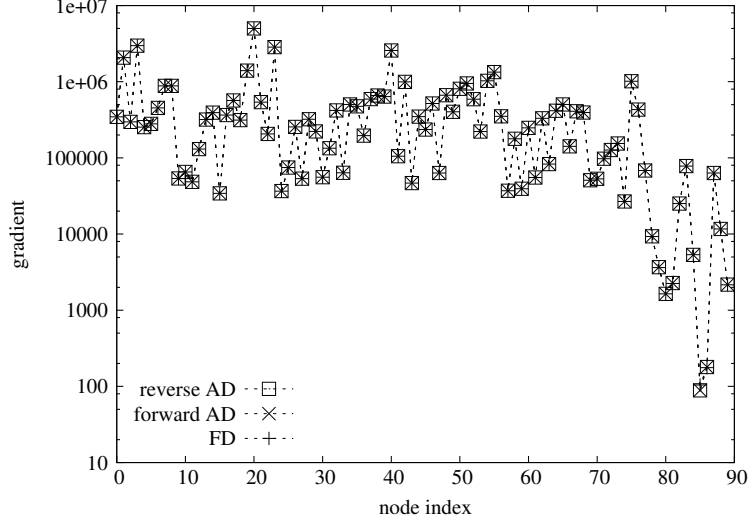


Figure 4: Comparison of $|\frac{\partial \sigma_{max}}{\partial \mathbf{x}}| \in \mathbb{R}^{90}$ using finite differences (FD), reverse AD, and forward AD

Degrees of Freedom, Tape	$\approx 30k$, RR	$\approx 200k$, RR	$\approx 200k$, RRI
relative run time	$2.5 \times T$	$2.1 \times T$	$2.8 \times T$
relative peak memory	$7.7 \times M$	$6.4 \times M$	$5.8 \times M$

Table 1: Axial fan run time and memory consumption results for gradient evaluation $\frac{\partial \sigma_{max}}{\partial \mathbf{x}}$ using the adjoint linear stress analysis. RR: **RealReverse**, RRI: **RealReverseIndex**. T : run time of primal. M : peak memory of primal

the majority of the run time, as is show in figure 7. Note that here the same numerical set up is used to solve both linear systems. The results also show that the **RealReverseIndex** AD type of CoDiPack offers a lower peak memory consumption at the expense of a higher computational effort.

4 Discrete Adjoint Vibration Analysis

A free vibration analysis starts with equation (3) in unloaded form:

$$K\mathbf{u} + M\ddot{\mathbf{u}} = 0 \quad (11)$$

With a proposed solution of

$$\mathbf{u} = \mathbf{u}_k e^{i\omega_k t}, \quad (12)$$

a generalized eigenvalue problem of the form

$$K\mathbf{u}_k = \omega_k^2 M\mathbf{u}_k \quad (13)$$

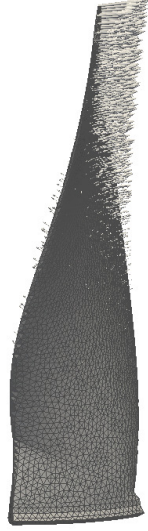


Figure 5: Axial fan mesh with displacement vectors

is obtained. Defining $\lambda_k := \omega_k^2$, the problem can be formulated as

$$(K(\mathbf{x}) - \lambda_k M(\mathbf{x})) \mathbf{u}_k = 0. \quad (14)$$

λ_k is the k -th eigenfrequency and \mathbf{u}_k the k -th eigenvector. The vibration analysis recycles the same stiffness matrix K from the linear stress analysis, but requires an additional matrix, the mass matrix $M \in \mathbb{R}^{m \times m}$. Note that K and M are symmetric, i.e.

$$K = K^T, \quad M = M^T, \quad (15)$$

and the eigenvectors can be normalized such that

$$\mathbf{u}_k^T M \mathbf{u}_k = 1. \quad (16)$$

The system (14) can be solved using an iterative eigenvalue solver and the process can be broken down into a two-step algorithm (figure 8). The gradient of interest in this case is

$$\frac{\partial \lambda_k}{\partial \mathbf{x}} \in \mathbb{R}^n, \quad (17)$$

where up to roughly twenty eigenfrequencies are usually of interest.

4.1 Tangent Model

The tangent model of the vibration analysis is required to derive the adjoint model and will be introduced in this section. One starts by differentiating the

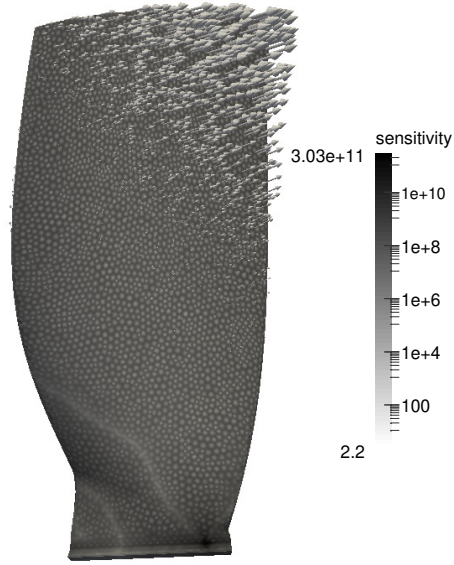


Figure 6: Axial fan with sensitivities $\left| \frac{\partial \sigma_{max}}{\partial \mathbf{x}} \right|$

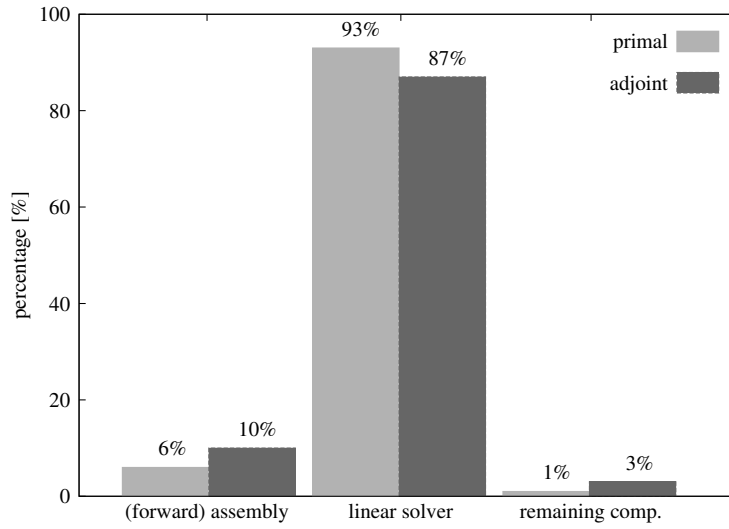


Figure 7: Run time breakdown of primal and adjoint linear stress analysis using axial fan test case

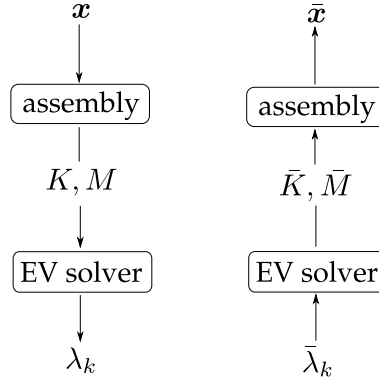


Figure 8: Vibration analysis algorithm breakdown. Primal(l) and adjoint(r)

generalized eigenvalue system (14) by x_i :

$$\frac{\partial(K - \lambda_k M)}{\partial x_i} \mathbf{u}_k + (K - \lambda_k M) \frac{\partial \mathbf{u}_k}{\partial x_i} = 0 \quad (18)$$

$$\frac{\partial K}{\partial x_i} \mathbf{u}_k - \frac{\partial \lambda_k}{\partial x_i} M \mathbf{u}_k - \lambda_k \frac{\partial M}{\partial x_i} \mathbf{u}_k + (K - \lambda_k M) \frac{\partial \mathbf{u}_k}{\partial x_i} = 0 \quad (19)$$

Left multiplying the result by \mathbf{u}_k^T gives

$$\mathbf{u}_k^T \frac{\partial K}{\partial x_i} \mathbf{u}_k - \mathbf{u}_k^T \frac{\partial \lambda_k}{\partial x_i} M \mathbf{u}_k - \mathbf{u}_k^T \lambda_k \frac{\partial M}{\partial x_i} \mathbf{u}_k + \mathbf{u}_k^T (K - \lambda_k M) \frac{\partial \mathbf{u}_k}{\partial x_i} = 0. \quad (20)$$

The definition of the eigenvalue problem (14), symmetry of K and M (15), and mass normalization (16) are used to simplify further to

$$\mathbf{u}_k^T \frac{\partial K}{\partial x_i} \mathbf{u}_k - \frac{\partial \lambda_k}{\partial x_i} \underbrace{\mathbf{u}_k^T M \mathbf{u}_k}_{=1} - \mathbf{u}_k^T \lambda_k \frac{\partial M}{\partial x_i} \mathbf{u}_k + \underbrace{\mathbf{u}_k^T (K - \lambda_k M)}_{=0} \frac{\partial \mathbf{u}_k}{\partial x_i} = 0, \quad (21)$$

which results in

$$\mathbf{u}_k^T \frac{\partial K}{\partial x_i} \mathbf{u}_k - \frac{\partial \lambda_k}{\partial x_i} - \lambda_k \mathbf{u}_k^T \frac{\partial M}{\partial x_i} \mathbf{u}_k = 0. \quad (22)$$

Finally, this is rearranged to

$$\frac{\partial \lambda_k}{\partial x_i} = \mathbf{u}_k^T \left(\frac{\partial K}{\partial x_i} - \lambda_k \frac{\partial M}{\partial x_i} \right) \mathbf{u}_k \quad (23)$$

$$\dot{\lambda}_k = \mathbf{u}_k^T \left(\dot{K} - \lambda_k \dot{M} \right) \mathbf{u}_k. \quad (24)$$

Given that the eigenvector \mathbf{u}_k , eigenfrequency λ_k , and the gradients $\frac{\partial K}{\partial x_i}$, $\frac{\partial M}{\partial x_i}$ have been computed, the eigenfrequency sensitivity (24) can be directly evaluated.

4.2 Adjoint Model

Previous work done by Lee [12] shows an adjoint formulation of the generalized eigenvalue problem, where an augmented response function was used to construct the adjoint equations. The method boils down to solving a linear system of equations for the adjoint variables and plugging them into the differentiated response function to compute the sensitivities (17). However, the gradients of the mass and stiffness matrices with respect to the design variables are still required to complete the equation. Dhondt et al. [3] computed the sensitivities via a symbolic approach as well by direct differentiation of the generalized eigenvalue problem. However, they used a perturbed stiffness and mass matrix to complete the equation, which led to numerically sensitive results.

The method presented in this paper uses a different approach and focuses more on the adjoint method by algorithmic differentiation, which will yield algorithmically exact gradients of the stiffness and mass matrices. While the majority of the discrete problem can be handled directly with AD, the iterative solver of the generalized eigenvalue problem has to be handled separately. It is well understood how this needs to be done for linear systems ([23] and section 3.2 of this paper), but not yet for generalized eigenvalue problems. This is presented in this section.

The adjoint model of the vibration analysis

$$\bar{\mathbf{x}} = \frac{\partial \lambda_k}{\partial \mathbf{x}}^T \bar{\lambda}_k, \quad (25)$$

is seeded with $\bar{\lambda}_k = 1$ to compute the sensitivities of each eigenfrequency of interest. The adjoint implementation (figure 8) involves a black-box AD differentiation of the assembly (step 2). As was the case for the linear stress analysis, one would want to avoid differentiating through an iterative solver for the generalized eigenvalue problem of (14). A similar approach can also be used to compute the adjoints \bar{K} and \bar{M} . Consider the generalized eigenvalue system solver as a function

$$\lambda_k = \lambda_k(K, M) \quad (26)$$

that takes matrices K and M as input arguments and outputs the eigenfrequency λ_k . The following dot product relationship [18, 4] between the tangent and adjoint models holds:

$$\langle \dot{K}, \bar{K} \rangle + \langle \dot{M}, \bar{M} \rangle = \langle \dot{\lambda}_k, \bar{\lambda}_k \rangle. \quad (27)$$

Plugging in the tangent model (24) yields

$$\langle \dot{K}, \bar{K} \rangle + \langle \dot{M}, \bar{M} \rangle = \langle \mathbf{u}_k^T (\dot{K} - \lambda_k \dot{M}) \mathbf{u}_k, \bar{\lambda}_k \rangle \quad (28)$$

$$\langle \dot{K}, \bar{K} \rangle + \langle \dot{M}, \bar{M} \rangle = \langle \mathbf{u}_k^T \dot{K} \mathbf{u}_k, \bar{\lambda}_k \rangle + \langle -\lambda_k \mathbf{u}_k^T \dot{M} \mathbf{u}_k, \bar{\lambda}_k \rangle. \quad (29)$$

In index notation, this can be expressed as

$$\underbrace{\sum_{i,j} \dot{K}_{ij} \bar{K}_{ij}}_{R1} + \underbrace{\sum_{i,j} \dot{M}_{ij} \bar{M}_{ij}}_{R2} = \underbrace{\bar{\lambda}_k \sum_{i,j} \dot{K}_{ij} u_{k,i} u_{k,j}}_{R1} + \underbrace{-\bar{\lambda}_k \lambda_k \sum_{i,j} \dot{M}_{ij} u_{k,i} u_{k,j}}_{R2}. \quad (30)$$

From relationship *R1*, an expression for the adjoint stiffness matrix \bar{K}_{ij} can be deduced:

$$\sum_{i,j} \dot{K}_{ij} \bar{K}_{ij} = \bar{\lambda}_k \sum_{i,j} \dot{K}_{ij} u_{k,i} u_{k,j} \quad (31)$$

$$\bar{K}_{ij} = \bar{\lambda}_k u_{k,i} u_{k,j} \quad (32)$$

Similarly, an expression for the adjoint mass matrix is found using relationship *R2*:

$$\sum_{i,j} \dot{M}_{ij} \bar{M}_{ij} = -\bar{\lambda}_k \lambda_k \sum_{i,j} \dot{M}_{ij} u_{k,i} u_{k,j} \quad (33)$$

$$\bar{M}_{ij} = -\bar{\lambda}_k \lambda_k u_{k,i} u_{k,j} = -\lambda_k \bar{K}_{ij} \quad (34)$$

After having solved for the eigenfrequency λ_k and eigenvector \mathbf{u}_k , adjoint equations (32) and (34) can be seeded with $\bar{\lambda}_k = 1$. The adjoints \bar{K} and \bar{M} are then plugged into step 2 of figure 8, the adjoint system assembly, to finally compute the gradient (25). Note that unlike the adjoint linear solver, no additional eigenvalue systems have to be solved and the adjoints can even be evaluated explicitly.

4.3 Accuracy and Performance Results

The accuracy of the adjoint vibration analysis is assessed by computing the gradient (17) with the same cantilever beam test case as in section 3.3. The accuracy results are shown in figure 9. The gradients computed using the adjoint implementation show an excellent accordance with the finite differences and forward AD computed gradients.

A run time comparison of the primal versus adjoint implementations was made using the larger axial fan test case described in section 3.3. The results of the run time breakdown are shown in figure 11. Note that in this case only the assembly of the mass matrix M is measured, since the stiffness matrix K from the linear stress analysis can be reused. Nearly the entire computational effort (99.5%) of the primal run is due to the iterative eigenvalue solver. Not needing to differentiate through this expensive function allows more efficient adjoint runs. Ten eigenfrequencies and their respective gradients are computed for the run time breakdown. This means that the timing results are for ten adjoint evaluations, each amounting to approximately 1% of the considered run time.

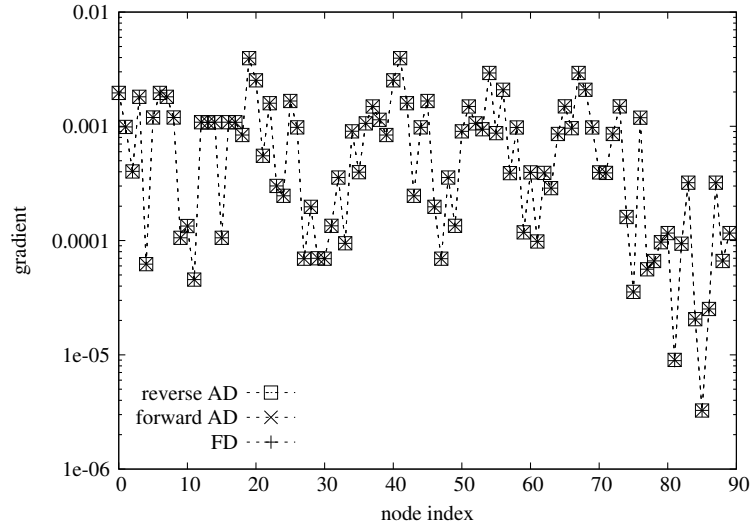


Figure 9: Comparison of $\|\frac{\partial \lambda_k}{\partial \mathbf{x}}\| \in \mathbb{R}^{90}$ using finite differences (FD), reverse AD, and forward AD

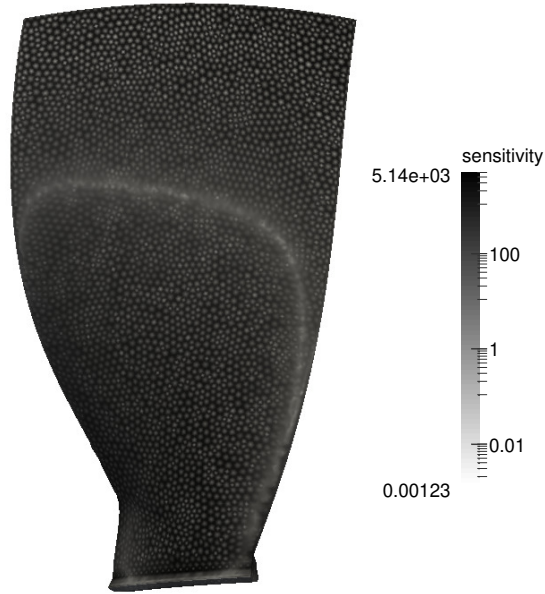


Figure 10: Axial fan with sensitivities $|\frac{\partial \lambda_0}{\partial \mathbf{x}}|$

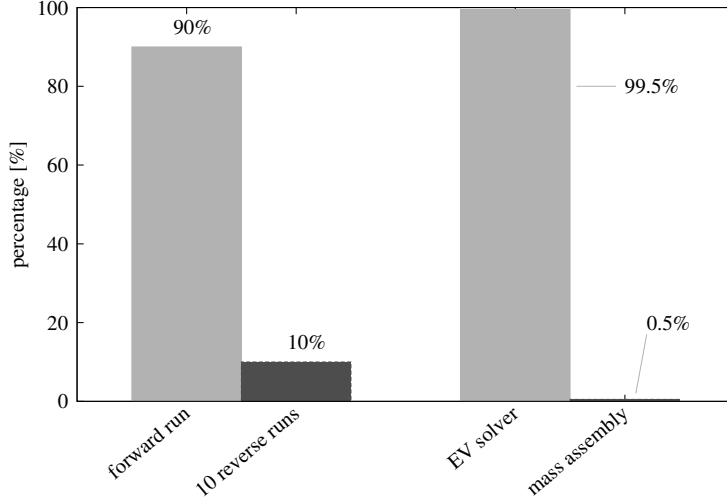


Figure 11: Vibration analysis run time breakdowns of adjoint(l) and primal(r)

5 Differentiation of Further Contributions

In addition to the node coordinates \mathbf{x} , which mainly serve the purpose of shape design parameters, alternative design parameters may be of interest as well. For instance, in the case of composite material applications based on first-order shear deformation theory (FSDT), the lamination parameters $V \in \mathbb{R}^l$ can be used to manipulate the material properties [5]. Gradients with respect to the lamination parameters could e.g. be used for material optimization problems. The design space could thus be extended as

$$\mathbf{x}^* = (\mathbf{x}, V) \in \mathbb{R}^{m+l}. \quad (35)$$

As a result, the stiffness and mass matrix computations are now defined to be dependent on the extended design space

$$K := K(\mathbf{x}^*), \quad M := M(\mathbf{x}^*), \quad (36)$$

such that the adjoint vibration analysis as described in section 4 produces gradients with respect to node coordinates, as well as lamination parameters

$$\frac{\partial \lambda_k}{\partial \mathbf{x}} \in \mathbb{R}^m, \quad \frac{\partial \lambda_k}{\partial V} \in \mathbb{R}^l. \quad (37)$$

This is mainly achieved by using an object-oriented design approach in the implementation of composite material capabilities into the code of the vibration solver. The subclass `CompositeElement`, a child of class `Element` (fig 12), defines the construction of the stiffness and mass matrices using lamination

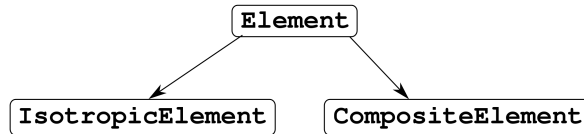


Figure 12: `IsotropicElement` and `CompositeElement` defined as subclasses of `Element`

parameters V . The algorithmically differentiated portion of the code was implemented for the parent class `Element`, enabling the introduction of additional element classes without additional AD differentiation. As a result, the composite element subclass and its matrix construction functions are automatically differentiated, which ensures maintainability of the differentiated code for future contributions.

Using a flat plate composite material test case (fig 13), the eigenfrequency gradients with respect to lamination parameters $\frac{\partial \lambda_k}{\partial V}$ were computed. A comparison of adjoint-computed gradients with FD and forward AD in figure 14 show a positive agreement.

6 Conclusion

This paper presented the methods used for a discrete adjoint gradient evaluation for linear stress and vibration analysis within the context of a discrete adjoint structural solver for turbomachinery applications. The adjoint linear stress analysis involves solving an additional linear system to avoid the black-box algorithmic differentiation of an iterative linear solver. Performance tests show that computing the sensitivities using the adjoint implementation costs as little as approximately 2.1 times the primal run time with 6.4 times the memory consumption, or 2.8 times the primal run time with 5.8 times the memory consumption.

The adjoint vibration analysis uses a similar approach to compute the adjoints of the generalized eigenvalue problem explicitly, without the need of solving any additional eigenvalue systems. Performance tests have shown that one reverse evaluation of this adjoint implementation costs about 1% of the forward run time. The accuracy test results for linear stress and vibration analysis show a good agreement of sensitivities computed using finite differences, forward AD, and adjoint, i.e. reverse, AD.

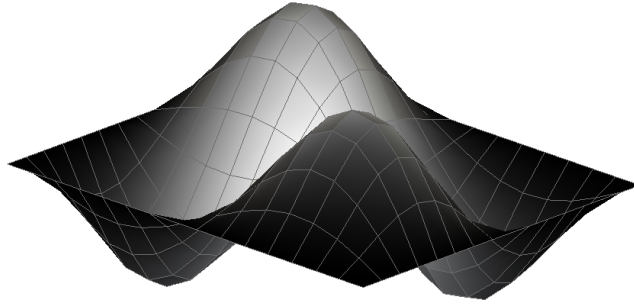


Figure 13: Flat plate free vibration test case showing mode 4 displacements

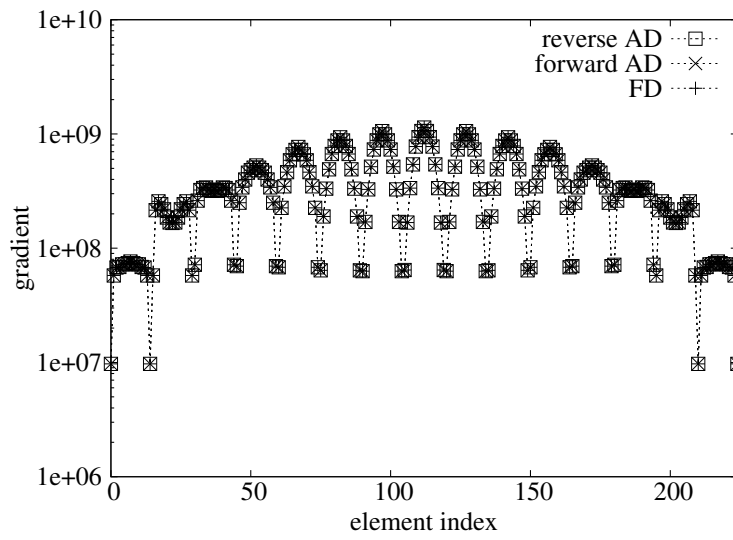


Figure 14: Comparison of gradient $\frac{\partial \lambda_0}{\partial V_{10}}$ computed using reverse AD, forward AD, and finite differences (FD)

With an AD differentiated code base, additional code contributions are automatically differentiated. As a result, eigenfrequency gradients with respect to composite material design parameters could be directly computed and the sensitivities correlate with the accuracy previously shown.

Further work would include extending the linear stress analysis to a nonlinear stress analysis, and a pre-loaded vibration analysis.

Funding

This project has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 642959.

References

- [1] Christopher Chahine, Tom Verstraete, and Li He. Multidisciplinary Design Optimization of an Aero-Engine Fan Blade with Consideration of Bypass and Core Performance. *Proceedings of the 11th World Congress on Structural and Multidisciplinary Optimization*, (June):5–10, 2015.
- [2] Guido Dhondt. *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. John Wiley & Sons, 2004.
- [3] Guido Dhondt, Marc Nagel, and Nicolas Gauger. Calculation of the eigenfrequency sensitivity of aircraft engine structures. *Proceedings of EUROGEN 2005*, 2005.
- [4] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, 2008.
- [5] Zafer Gürdal, Raphael T Haftka, and Prabhat Hajela. *Design and optimization of laminated composite materials*. John Wiley & Sons, 1999.
- [6] A. Jameson and J. Reuther. Control Theory Based Airfoil Design Using the Euler Equations. *AIAA paper 94-4272, 5th AIAA/US AF/NAS A/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, FL*, 1994.
- [7] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.
- [8] Antony Jameson, Sriram Shankaran, and Luigi Martinelli. Continuous adjoint method for unstructured grids. *AIAA journal*, 46(5):1226–1239, 2008.

- [9] Gaetan K W Kenway, Graeme J Kennedy, and Joaquim R R A Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA Journal*, 52(5):935–951, 2014.
- [10] JH Kim, JH Choi, and KY Kim. Design Optimization of a Centrifugal Compressor Impeller Using Radial Basis Neural Network Method. *ASME Turbo Expo*, (October):1–9, 2009.
- [11] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. In *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, 1979.
- [12] Tae Hee Lee. Adjoint Method for Design Sensitivity Analysis of Multiple Eigenvalues and Associated Eigenvectors. *AIAA Journal*, 45(8):1998–2004, 2007.
- [13] Jiaqi Luo, Chao Zhou, and Feng Liu. Multipoint design optimization of a transonic compressor blade by using an adjoint method. *Journal of Turbomachinery*, 136(5):051005, 2014.
- [14] Dimitri Mavriplis, Evan Anderson, Ray S. Fertig, and Mark Garnich. Development of a High-Fidelity Time-Dependent Aero-Structural Capability for Analysis and Design. *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (January):1–14, 2016.
- [15] Dimitri J. Mavriplis. Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes. *AIAA Journal*, 45(4):741–750, 2007.
- [16] Temesgen Mengistu and Wahid Ghaly. Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models. *Optimization and Engineering*, 9(3):239–255, 2008.
- [17] Lasse Mueller, Zuheyr Alsalihi, and Tom Verstraete. Multidisciplinary optimization of a turbocharger radial turbine. *Journal of Turbomachinery*, 135(2):021022, 2013.
- [18] Uwe Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*, volume 24. Siam, 2012.
- [19] Olivier Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(01):97–110, 1974.
- [20] M. Sagebaum, T. Albring, and N. R. Gauger. High-performance derivative computations using codipack. *arXiv preprint arXiv:1709.07229*, 2017.
- [21] Marc Schwalbach and Tom Verstraete. Towards multidisciplinary adjoint optimization of turbomachinery components. In V. Plevris M. Papadrakakis, V. Papadopoulos, G. Stefanou, editor, *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering*, pages 3999 – 4010, Crete, 2016.

- [22] U Siller, C Voß, and E Nicke. Automated Multidisciplinary Optimization of a Transonic Axial Compressor. *Proceedings of the 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, (January):1–12, 2009.
- [23] Mohamed Tadjouddine, Shaun A Forth, and Andy J Keane. Adjoint differentiation of a structural dynamics solver. In *Automatic Differentiation: Applications, Theory, and Implementations*, pages 309–319. Springer, 2006.
- [24] Tom Verstraete. *Multidisciplinary turbomachinery component optimization considering performance, stress, and internal heat transfer*. PhD thesis, von Karman Institute, 2008.
- [25] Benjamin Walther and Siva Nadarajah. Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. *Journal of turbomachinery*, 135(2):021017, 2013.
- [26] Sebastian Willeke and Tom Verstraete. Adjoint optimization of an internal cooling channel u-bend. In *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*. American Society of Mechanical Engineers, 2015.