International Forum on Aeroelasticity and Structural Dynamics
IFASD 2024
17-21 June 2024, The Hague, The Netherlands

ELASTIC WIND TUNNEL MODEL DESIGN VIA EIGENVECTOR-BASED LEVEL-SET TOPOLOGY OPTIMIZATION

Eisuke Nakagawa[1], **Tomohiro Yokozeki**[1], **Natsuki Tsushima**[2]

[1]University of Tokyo
Hongo, Tokyo, 113-8656, Japan
nakagawa-eisuke185@g.ecc.u-tokyo.ac.jp
yokozeki@aastr.t.u-tokyo.ac.jp

[2]Japan Aerospace Exploration Agency
Mitaka, Tokyo 181-0015, Japan
tsushima.natsuki@jaxa.jp

Keywords: Wind tunnel model, topology optimization

Abstract: In the field of aeroelasticity, acquiring both static and dynamic aeroelastic characteristics via wind tunnel test is useful. However, designing elastic wind tunnel models has been conventionally laborious. To address this issue, this study aims to develop a numerical method for designing elastic wind tunnel models by taking advantage of additive manufacturing technique. The wind tunnel model is designed to have equivalent static responses, eigenvalues, and eigenvectors to full-scale aircraft. The internal structure of the wind tunnel model is reconstructed using topology optimization. The optimization is performed using a level-set-based method with conforming meshes. The proposed method successfully optimized the static characteristic to meet a target static characteristic. As for the optimization of dynamic characteristics, however, some numerical instabilities have been observed, preventing logical optimization of eigenvalues and eigenmode shapes.

1 INTRODUCTION

It is attracting interest to design and control dynamic characteristics of structures as well as static characteristics. In the field of aeroelasticity, the acquisition of aeroelastic characteristics through wind tunnel testing remains important today, despite the development of numerical computation methods and high-performance computers. The significance of aeroelastic analysis has increased due to the growing demand for designing higher-performance aircraft. In transonic and supersonic airflow, where new developments are actively taking place, the cost of numerical computation is impractically high. Also, the existence of geometrically nonlinear effects adds to the complexity of the analysis. Therefore, experimental testing using wind tunnel models becomes indispensable. To analyze dynamic characteristics like flutter through wind tunnel testing, it is essential to replicate not only the outer geometry but also structural properties such as mass and stiffness distribution. However, the scaled model's specific stiffness and overall density vary significantly based on the scale ratio and wind tunnel testing configurations, making their replication challenging [1].

In recent years, the methodology for designing wind tunnel models has been developed alongside technological innovations. Traditional wind tunnel models were constructed by designing

non-load-bearing skin fairings, simplified internal structures, and incorporating weights for adjustments [2]. Such processes were time-consuming and costly. However, this situation is changing with the emergence of Additive Manufacturing (AM) [3, 4]. Recent advancements in AM allow for the fabrication of complex structures using various materials such as plastics, metals and combinations of which [5]. Flutter tests of wind tunnel models manufactured by metal 3D printing have been conducted [6], yet sufficient research has not been conducted on methods for reverse-engineering the internal structures of wind tunnel models to meet given structural characteristics. In designing structures using AM, topology optimization stands as a prominent method. The primary challenge in topology optimization lies in determining the sensitivity of input variables. Combined with automatic differentiation [7], this gradient computation can be performed efficiently and accurately for a wide range of objective functions [8]. However, the application of topology optimization to the design of elastic wind tunnel models has not been sufficiently explored.

This study aims to establish a method for designing elastic wind tunnel models with equivalent static responses, eigenvalues, and eigenmodes to full-scale aircraft. This is because wind tunnel models with the same static responses, eigenvalues, and eigenmodes are expected to exhibit the same aerodynamic elastic behavior, even if their internal structures differ. Elastic wind tunnel models are assumed to be formed using AM, and this is achieved by reconstructing the internal structure through topology optimization. For topology optimization, a level-set-based method using conforming meshes, which is expected to provide the highest accuracy, is adopted [9].

2 METHODOLOGY

2.1 Level-set topology optimization

A level-set-based approach is used in this work to optimize the topology of elastic wind tunnel models. The overview of level-set topology optimization is summarized in [9]. In this work, the value of the level-set function ϕ is

$$\begin{cases} \phi < 0 & \text{insie the material domain} \\ \phi = 0 & \text{on the boundary} \\ \phi > 0 & \text{outside the material domain} \end{cases} \tag{1}$$

The optimization procedure in this work is divided into four stages as shown in Fig. 1.

1. A scaled surface geometry is input and a structured grid is set inside the geometry to parameterize the level-set function (LSF). The LSF parameter is denoted as ψ and is initialized in this stage. The LSF parameter ψ is defined in the structured grid, and the value of LSF is calculated as

$$\phi = \boldsymbol{f}_{\text{RBF}}(\psi) \tag{2}$$

   where ϕ is the LSF value on the structured grid, and $\boldsymbol{f}_{\text{RBF}}$ is a radial basis function (RBF) interpolation function.

2. The boundary of the structure is extracted in STL format. The boundary is implicitly defined by the LSF as the zero level set of the LSF and is approximated with triangle polygons. For convenience, ϕ and the coordinates of nodes on the boundary $\boldsymbol{v}_{\text{surf}}$ are linked by two mapping matrices $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, which are calculated in every iteration, in the

following form:

$$\boldsymbol{v}_{\text{surf}} = \frac{\boldsymbol{M}_1\boldsymbol{\phi}}{\boldsymbol{M}_2\boldsymbol{\phi}} \tag{3}$$

3. The material domain is discretized by tetrahedral elements, and the global stiffness matrix and global mass matrix are calculated by the finite element method (FEM). The conforming meshing is done by a third-party software TetGen [10]. The coordinates of surface mesh nodes and $\boldsymbol{v}_{\text{surf}}$ are linked by another mapping matrix $\boldsymbol{M}_3$ in the following form:

$$\boldsymbol{v}_{\text{meshSurf}} = \boldsymbol{M}_3\boldsymbol{v}_{\text{surf}} \tag{4}$$

so that $\boldsymbol{\phi}$ and the resulting stiffness matrix and mass matrix are indirectly linked by simple linear algebra calculation.

4. Structural analysis is done by FEM analysis. The objective function is calculated using the results of the FEM analysis, and the gradient of the objective function F_{obj} w.r.t. $\boldsymbol{\psi}$ is calculated. $\boldsymbol{\psi}$ is updated using the gradient information as

$$\boldsymbol{\psi} = \boldsymbol{\psi} - \alpha\frac{\partial F_{\text{obj}}}{\partial \boldsymbol{\psi}} \tag{5}$$

where α is a step size. The updated $\boldsymbol{\psi}$ is converted to $\boldsymbol{\phi}$, and the process is repeated from stage (2) to stage (4) until the convergence of the objective function.

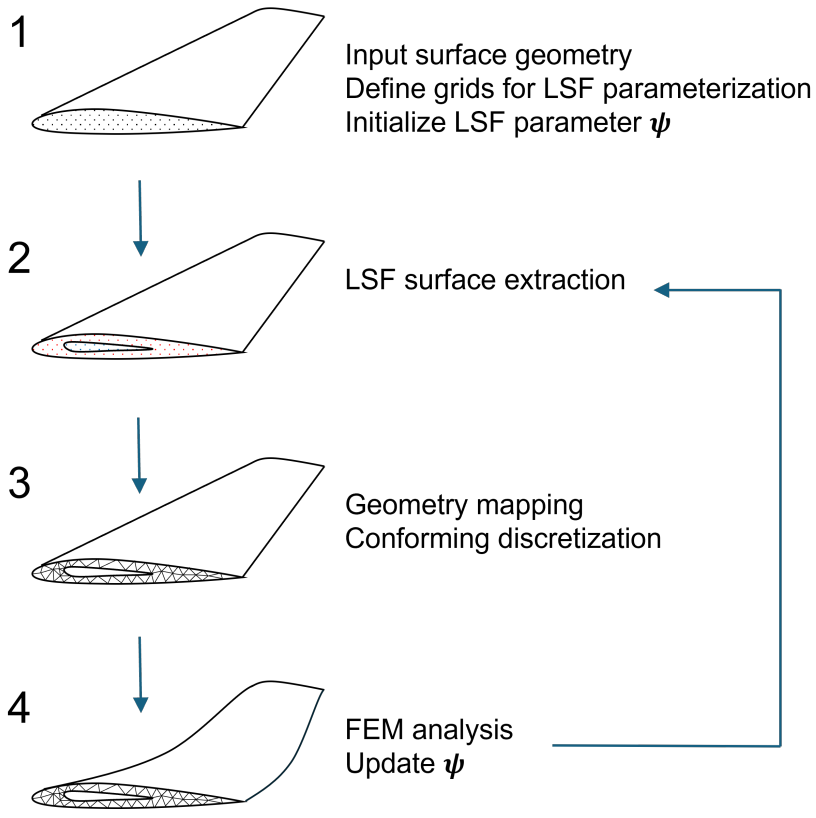

Figure 1: Optimization procedure

The core part of the program is written with a Python library JAX [11]. With JAX's Autograd tool, in which JAX automatically differentiates Python codes, the optimization uses gradient

information to update the design variables. The objective function is defined using total mass, static deflection, eigenfrequencies, and eigenvectors. The FEM calculation is done by third-party software MSC Marc and the results are imported into the JAX program, which is due to the lack of fast solvers in JAX. The undergoing updates aim to replace this outsourcing with a self-contained JAX program. The calculation of the gradient of static deflection w.r.t. input global stiffness matrix is implemented via an implicit differentiation method [12]. The gradient of eigenvectors w.r.t. input global stiffness matrix and input global mass matrix is calculated via an approximation method described in 2.3.

2.2 Consideration of static deformation

The static characteristic of the wind tunnel model is represented by a compliance matrix of its reduced order FE model. The reduction of the model is achieved by the Guyan reduction method [13], which is exact for static analysis. The active nodes are randomly chosen from the surface nodes of the model.

2.3 Approximation of eigenvector gradient

The calculation of eigenvector gradients for large sparse matrices is computationally expensive as it requires all eigenvectors to calculate a single eigenvector gradient [14]. Several methods have been proposed to approximate the eigenvector gradients. However, to the authors' knowledge, no method is suitable for calculating eigenvector gradients w.r.t. every element of the matrix. In this study, an approximation of the eigenvector gradients is introduced with a simple error estimation method. The characters used in this section do not hold any specific meaning in other sections.

2.3.1 Gradient approximation

Consider a standard eigenvalue problem

$$\boldsymbol{A}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \quad \boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij} \tag{6}$$

where $\boldsymbol{A}$ is a n×n real symmetric matrix, $\boldsymbol{u}_i$ is the i-th eigenvector, λ_i is the i-th eigenvalue, and δ_{ij} is the Kronecker delta. While the exact eigenvalue gradient is given as

$$\frac{\partial \lambda_i}{\partial \boldsymbol{A}} = \boldsymbol{u}_i^\top \otimes \boldsymbol{u}_i \tag{7}$$

where $\otimes$ is the Kronecker product, the analytical eigenvector gradient is given as

$$\frac{\partial \boldsymbol{u}_i}{\partial A_{kl}} = \sum_{i \neq j}^{n} \left(\frac{1}{\lambda_i - \lambda_j} \left(\frac{\partial \boldsymbol{A}}{\partial A_{kl}} \boldsymbol{u}_i \right) \cdot \boldsymbol{u}_j \right) \boldsymbol{u}_j = \sum_{i \neq j}^{n} \left(\frac{[\boldsymbol{u}_i]_l \cdot [\boldsymbol{u}_j]_k}{\lambda_i - \lambda_j} \right) \boldsymbol{u}_j \tag{8}$$

where $[\boldsymbol{u}_i]_l$ is the l-th element of $\boldsymbol{u}_i$. The contribution of the j-th eigenvector to the gradient of the i-th eigenvector decreases as the difference between the eigenvalues λ_i and λ_j increases. Since our interest is in the lowest eigenvalues and eigenvectors, the approximation neglects the contribution of the higher vectors.

2.3.2 Error estimation

Define $\boldsymbol{e}_{sj}$ as

$$\boldsymbol{e}_{sj} = \left(\frac{[\boldsymbol{u}_i]_{l_s} \cdot [\boldsymbol{u}_j]_{k_s}}{\lambda_i - \lambda_j} \right) \boldsymbol{u}_j \tag{9}$$

$$\|\boldsymbol{e}_{sj}\| = e_{sj} \tag{10}$$

The exact gradient vector $\boldsymbol{v}_{s\,\text{exact}}$ and the approximated gradient vector $\boldsymbol{v}_{s\,\text{approx}}$ are given as

$$\boldsymbol{v}_{s\,\text{exact}} = \sum_{j=1}^{n} \boldsymbol{e}_{sj}, \quad \boldsymbol{v}_{s\,\text{approx}} = \sum_{j=1}^{m} \boldsymbol{e}_{sj} \tag{11}$$

where m is the number of eigenvectors considered in the approximation and $m < n$.

Define new 1D vectors $\boldsymbol{v}_{\text{exact}}$ and $\boldsymbol{v}_{\text{approx}}$ as

$$\boldsymbol{v}_{\text{exact}} = [\boldsymbol{v}_{1\,\text{exact}}^\top, \ldots, \boldsymbol{v}_{S\,\text{exact}}^\top]^\top, \quad \boldsymbol{v} = [\boldsymbol{v}_{1\,\text{approx}}^\top, \ldots, \boldsymbol{v}_{S\,\text{approx}}^\top]^\top \tag{12}$$

The cosine similarity between the exact and approximated gradient vectors is given as

$$\frac{\boldsymbol{v}_{\text{exact}}^\top \boldsymbol{v}_{\text{approx}}}{\|\boldsymbol{v}_{\text{exact}}\|\|\boldsymbol{v}_{\text{approx}}\|} = \frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sqrt{\sum_{s=1}^{S}\sum_{j=1}^{n} e_{sj}^2}\sqrt{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}} = \sqrt{\frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sum_{s=1}^{S}\sum_{j=1}^{n} e_{sj}^2}} \tag{13}$$

Here we introduce a heuristic assumption that e_{sj} monotonically decreases at a rate of at least first order w.r.t. j and becomes infinitesimal at $j = n$. Therefore

$$\begin{aligned}
\frac{\boldsymbol{v}_{\text{exact}}^\top \boldsymbol{v}_{\text{approx}}}{\|\boldsymbol{v}_{\text{exact}}\|\|\boldsymbol{v}_{\text{approx}}\|} &= \sqrt{\frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sum_{s=1}^{S}\sum_{j=1}^{n} e_{sj}^2}} = \sqrt{\frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2 + \sum_{s=1}^{S}\sum_{j=m}^{n} e_{sj}^2}} \\
&\geq \sqrt{\frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2 + \sum_{s=1}^{S}\sum_{j=m}^{n} \left(\frac{n-j}{n-m} e_{sm}\right)^2}} \\
&= \sqrt{\frac{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2}{\sum_{s=1}^{S}\sum_{j=1}^{m} e_{sj}^2 + \frac{1}{(n-m)^2}\sum_{s=1}^{S} e_{sm}^2 \sum_{j=m}^{n}(n-j)^2}}
\end{aligned} \tag{14}$$

In this way, the lower bound of the cosine similarity is approximated using m eigenvectors.

2.4 Aeroelastic scaling

In this study, the Mach number and dynamic pressure are control variables. In a wind tunnel where the airflow is quasi 1D isentropic, the relation between the total temperature T_{0t} and the static temperature T_t is given by

$$\frac{T_{0t}}{T_t} = 1 + \frac{\gamma - 1}{2} M^2 \tag{15}$$

where γ is the heat capacity ratio and M is the Mach number. Using the temperature at flight altitude T_f, the scaling factor for velocity κ_v, identical to the scaling factor for sonic speed, is given as

$$\kappa_v = \frac{a_t}{a_w} = \sqrt{\frac{\gamma R T_t}{\gamma R T_f}} = \sqrt{\frac{T_{0t}}{\left(1 + \frac{\gamma-1}{2} M^2\right) T_f}} \tag{16}$$

where a_t and a_w are the sonic speeds at flight altitude and wind tunnel, respectively, and R is the gas constant.

The scale factor for length $\kappa_l = L_t/L_f$, the scale factor for the dynamic pressure $\kappa_p = P_t/P_f$ and κ_v give the other scaling factors as

$$\kappa_f = \kappa_p \kappa_l^2 \tag{17}$$

$$\kappa_t = \kappa_l / \kappa_v \tag{18}$$

$$\kappa_m = \kappa_l^3 \kappa_p / \kappa_v^2 \tag{19}$$

where κ_f, κ_t and κ_m are the scaling factors for force, time and mass, respectively.

2.5 Loss function

The definition of the loss functions used in this work is given in this section. In optimization with automatic differentiation, the choice of the loss function is up to the users. The loss functions for mass, static deformation, eigenvalues, and eigenvectors are defined as

$$F_m = c_m \left| \frac{m^{(i)}}{m_{\text{target}}} - 1 \right| \tag{20}$$

$$F_s = c_s \| \boldsymbol{C}^{(i)} - \boldsymbol{C}_{\text{target}} \|_{\text{F}}^{\ 2} \tag{21}$$

$$F_\lambda = c_\lambda \sum_k^{n_\lambda} \left| \frac{\lambda_k^{(i)}}{\lambda_{\text{target},k}} - 1 \right| \tag{22}$$

$$F_v = c_v \sum_k^{n_\lambda} \left(\left| \mathcal{C}(\boldsymbol{v}_k^{(i)}, \boldsymbol{v}_{\text{target},k}) \right| - 1 \right) \tag{23}$$

where m is the mass, $\boldsymbol{C}$ is the compliance matrix of the reduced order model, λ_k is the k-th eigenvalue, $\boldsymbol{v}_k$ is the k-th eigenvector, n_λ is the number of eigenvalues considered, and $\mathcal{C}(\boldsymbol{a}, \boldsymbol{b})$ is the cosine similarity between $\boldsymbol{a}$ of $\boldsymbol{b}$. $|\square|$ denotes absolute value and $\|\square\|_{\text{F}}$ denotes Frobenius norm. The superscript $\square^{(i)}$ denotes values of the i-th iteration, and the subscription $\square_{\text{target}}$ denotes the target values. The coefficients $c_\square$ are the normalizers for each loss function. The objective function is given as a weighted sum of these loss functions.

3 NUMERICAL RESULTS

In this section, optimization results for the single objective cases are presented. The JAX program is run on an NVIDIA GeForce RTX 4090, and MSC Marc is run on an i9-9980XE CPU.

3.1 Model description

The reference wind tunnel model is described in this section. The wind tunnel model is a scaled-down model of a test wing with NASA SC(2)-07 14 airfoil. The scaling factors are calculated as explained in 2.4. The configuration of the full sale wing and scaling factors are shown in Tables 1-2.

Table 1: Configuration of the full-scale wing

Semispan length	8 m
Root chord	2 m
Sweep back angle	20°
Flight Mach num.	0.85

Table 2: Scaling factor for the wind tunnel model

Length κ_l	0.03
Dynamic pressure κ_p	2.4
Time κ_t	0.028
Mass κ_m	5.6×10^{-5}

The wing structure is defined as a shell model with a uniform thickness of 3 mm. The material properties are provided in Table 3.

Table 3: Material properties

Young's modulus	70 GPa
Poisson's ratio	0.3
Density	2700 kg/m^3

The same material properties are used for the wind tunnel model. The overview of the full-scale wing is shown in Fig. 2.

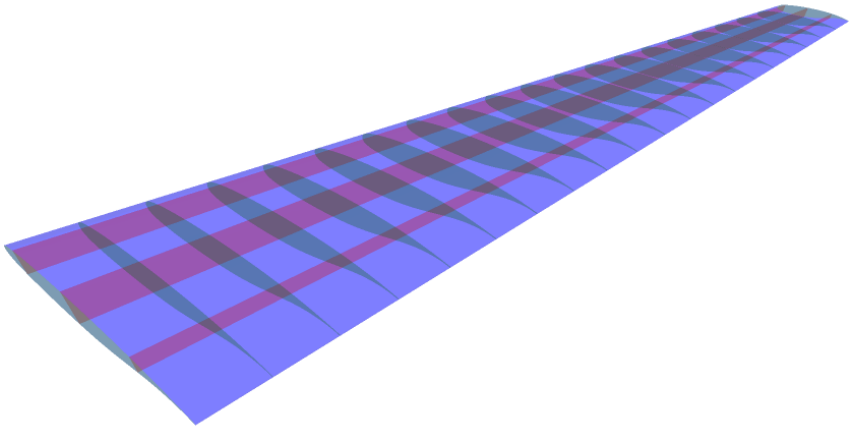

Figure 2: Overview of the full-scale wing

3.2 Calculation results

The LSF parameter ψ is defined in a structured grid system. The value of the level-set function ϕ at each grid point is initialized so that the initial wind tunnel model design has uniformly distributed holes inside the bulk material domain as shown in Fig. 3.

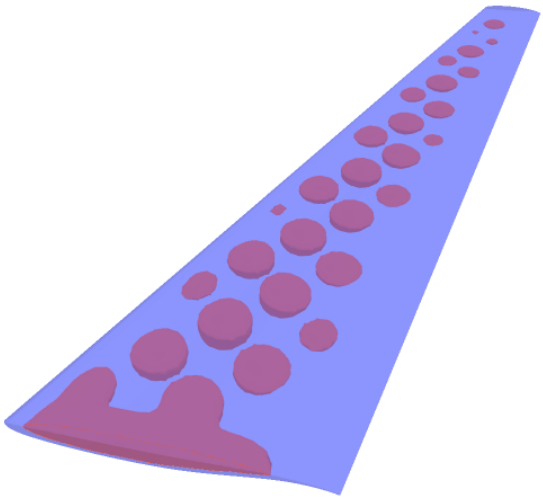

Figure 3: Initialization of the level-set function

In each iteration process, the derivative of the loss function F_s w.r.t. the LSF parameter ψ is calculated by Automatic Differentiation. The LSF parameter ψ is updated by the gradient

descent method as

$$\boldsymbol{\psi}^{(k+1)} = \boldsymbol{\psi}^{(k)} - \alpha \frac{\partial F_s}{\partial \boldsymbol{\psi}}, \tag{24}$$

where α is a step size.

3.2.1 Static deformation optimization result

The optimization result for the static deformation is presented in this section. The value of the loss function at each iteration step is shown in Fig. 4. The value of the loss function was reduced to less than half after optimization. The distribution of the holes inside the wind tunnel model after the final iteration is shown in Fig. 5. Notably, the size of the holes at the wing root increased significantly, indicating substantial changes during the optimization process. This implies that the stiffness in the region near the root greatly influences the overall deformation behavior, which is a reasonable outcome. Furthermore, the difference in wingtip displacement under vertical load decreased from 29% to 2.7% before and after optimization.

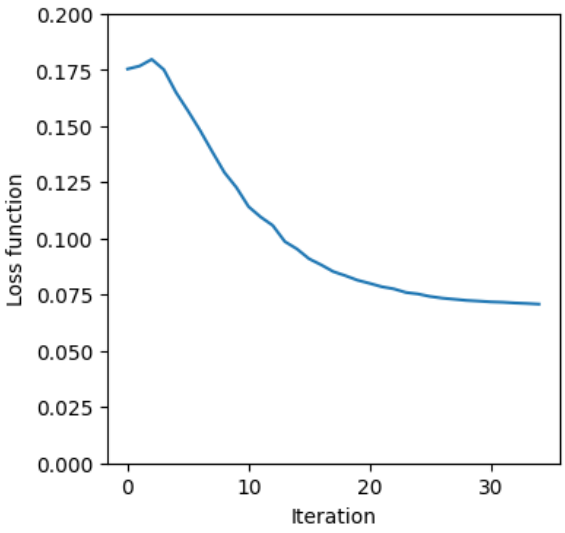

Figure 4: Loss function value during optimization

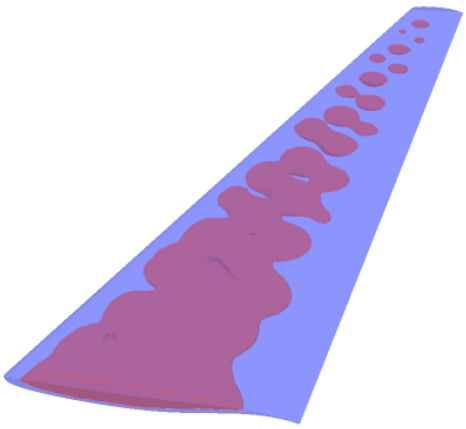

Figure 5: Wind tunnel model in 35th iteration

3.2.2 Eigenvalue optimization result

Two optimization results for the eigenvalue are presented in this section.

In the first case, only the first eigenvalue is optimized. The values of the loss function F_λ at each iteration step are shown in Fig. 6 and the wind tunnel model after the final iteration is shown in Fig. 7. The eigenfrequency of the first mode in the initial and final iteration is shown in Table 4. During the optimization process, a sharp increase in the loss function value was observed in the 7th iteration, yet the value recovered small in the final iteration. After 9 iterations, the inner structure of the wind tunnel model has changed notably. The holes in the wing tip region became small, while in the middle section, the holes became large and some merged with each other. The holes in the wing root section have not changed. The eigenfrequency of the first mode was reduced to 152.2 Hz from 154.0 Hz after optimization, which is close to the target value of 152.0 Hz.

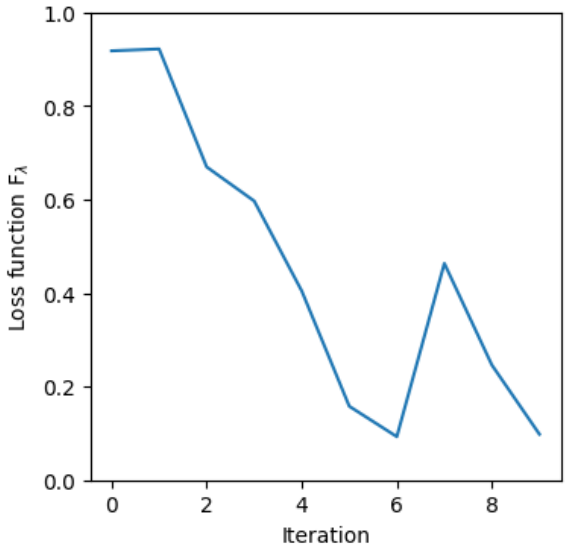

Figure 6: Loss function value during optimization

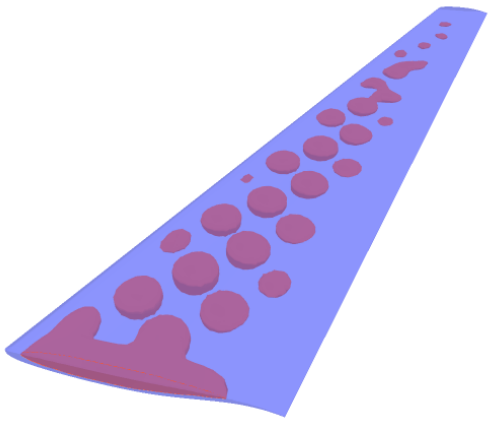

Figure 7: Wind tunnel model in 9th iteration

Table 4: Eigenfrequency of the first mode in each iteration

	Eigenfrequency of 1st mode
Iteration 0	154.0 Hz
Iteration 9	152.2 Hz
Target value	152.0 Hz

In the second case, the first two eigenvalues are optimized simultaneously. The Eigenfrequencies of the first and second modes in the initial and 7th (best case) iterations are shown in Table 5. The values of the loss function F_λ at each iteration step are shown in Fig. 8. The values of F_λ slightly improved until the 7th iteration, yet failed to converge in the following iteration. The eigenfrequency of the first mode was reduced to 153.7 Hz from 154.0 Hz after optimization and the eigenfrequency of the second mode was increased to 648.4 Hz from 647.3 Hz after optimization. Although the improvement is relatively small, the optimization managed to reduce the eigenfrequency of the first mode while increasing the eigenfrequency of the second mode simultaneously.

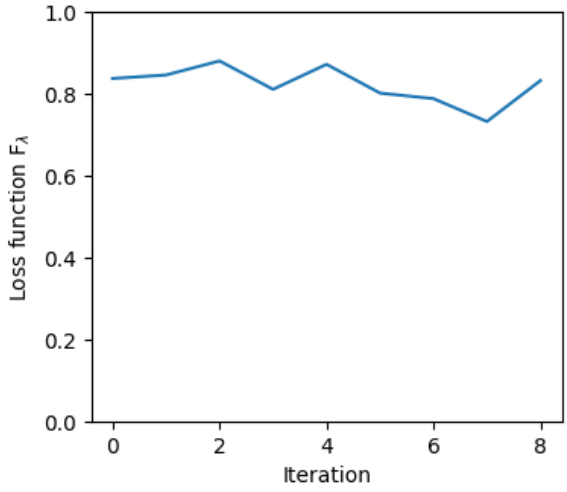

Figure 8: Loss function value during optimization

Table 5: Eigenfrequencies of the first and second modes in each iteration

	Eigenfrequency of 1st mode	Eigenfrequency of 2nd mode
Iteration 0	154.0 Hz	647.3 Hz
Iteration 7	153.7 Hz	648.4 Hz
Target value	152.0 Hz	655.2 Hz

3.2.3 Eigenvector optimization result

The optimization result for the eigenvector is presented in this section. The first eigenmode shape in the initial state was already close to the target shape and the cosine similarity was more than 0.999. Therefore, the first two eigenmodes were optimized simultaneously. The values of the loss function F_{v} at each iteration step are shown in Fig. 9. The mean cosine similarity of the first and second mode shapes from the target mode shapes is shown in Table 6. The value of F_{v} became minimum at the 5th iteration and started diverging in the following iteration. The value of the mean cosine similarity is slightly improved after optimization.

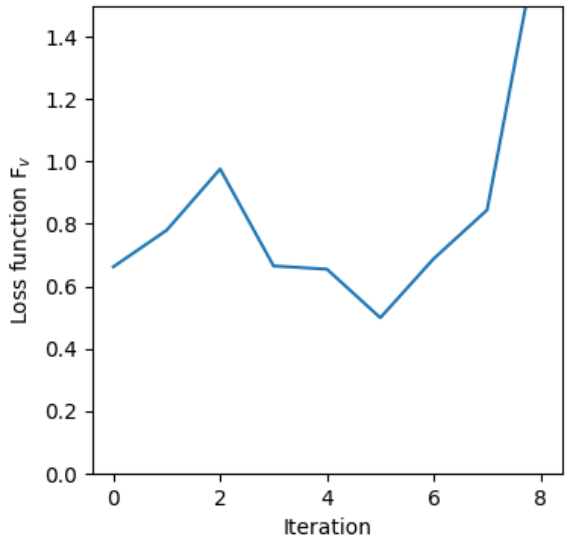

Figure 9: Loss function value during optimization

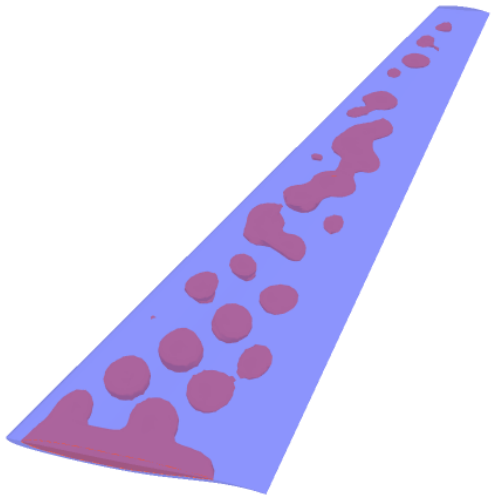

Figure 10: Wind tunnel model in 5th iteration

Table 6: Mean cosine similarity of 1st and 2nd modeshapes

	Mean cosine similarity of 1st and 2nd modeshapes
Iteration 0	0.991
Iteration 5	0.993

4 DISCUSSION

In the case of static deformation optimization, the loss function decreased monotonously, indicating that the optimization process was successful. However, in the case of eigenvalue optimization, the loss function fluctuated around the initial value. Also, irrationally large sensitivities have been observed during the calculation. In a simple test case where the number of total degrees of freedom is up to 5,000, the gradients calculated via automatic differentiation coincided with the gradients calculated via the finite difference method. However, in the case of

a more complex model with 100,000 degrees of freedom, large discrepancies were observed. In some cases, the maximum eigenvalue gradients, normalized to 1, with respect to the coordinate values of a single node exceeded 1,000. This gradient value should be at least less than 0.001. These large values can be significant noise and conceal the actual gradient information, resulting in divergence of the optimization process. A possible cause of this problem is the existence of low-quality elements in the mesh. The results of eigenvalue analysis are more significantly affected by the quality of the mesh compared to the results of static deformation analysis. This problem is more prominent in the case of eigenvector optimization.

5 CONCLUSION

A topology optimization method for optimizing the dynamic and static characteristics of a structure is proposed. As a demonstration, the method is applied to a wind tunnel model design problem and the optimization results for static and dynamic optimization cases are shown. In the dynamic optimization case, extraordinarily large sensitivities were observed during the calculation, resulting in the potential loss of actual gradient information. The ongoing work is to investigate the cause of this problem and to develop a solution. In the current implementation, some elements in the derivative of the loss functions F_λ and F_v are overly estimated, resulting in an exploding gradient problem. This is possibly due to the existence of low-quality elements in the mesh. Because of the computational cost in calculating the mapping matrix $\boldsymbol{M}_3$ in Eq. 4, the number of nodes is virtually limited to 100,000. The proceeding modification includes adding a preprocessing step to accelerate the calculation of the mapping matrix $\boldsymbol{M}_3$ to allow for a larger number of nodes.

6 REFERENCES

[1] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L. (2013). *Aeroelasticity*. Courier Corporation.

[2] Tang, D. and Dowell, E. H. (2001). Experimental and theoretical study on aeroelastic response of high-aspect-ratio wings. *AIAA journal*, 39(8), 1430–1441.

[3] Zhu, W. (2019). Models for wind tunnel tests based on additive manufacturing technology. *Progress in Aerospace Sciences*, 110, 100541.

[4] Oliveira, É., Sohouli, A., Afonso, F., et al. (2022). Dynamic scaling of a wing structure model using topology optimization. *Machines*, 10(5), 374.

[5] Tsushima, N. and Higuchi, R. (2022). Stiffness and strength evaluation of lattice-based mechanical metamaterials by decoupled two-scale analysis. *Materials Today Communications*, 31, 103598.

[6] Tsushima, N., Saitoh, K., and Nakakita, K. (2023). Structural and aeroelastic characteristics of wing model for transonic flutter wind tunnel test fabricated by additive manufacturing with alsi10mg alloys. *Aerospace Science and Technology*, 140, 108476.

[7] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., et al. (2018). Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153), 1–43.

[8] Chandrasekhar, A., Sridhara, S., and Suresh, K. (2021). Auto: a framework for automatic differentiation in topology optimization. *Structural and Multidisciplinary Optimization*, 64(6), 4355–4365. ISSN 1615-1488. doi:10.1007/s00158-021-03025-8.

[9] Van Dijk, N. P., Maute, K., Langelaar, M., et al. (2013). Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48, 437–472.

[10] Si, H. et al. (2007). Tetgen. *Tetrahedral mesh generator and threedimensional delaunay triangulator*.

[11] Bradbury, J., Frostig, R., Hawkins, P., et al. (2018). Jax: composable transformations of python+ numpy programs.

[12] Blondel, M., Berthet, Q., Cuturi, M., et al. (2022). Efficient and modular implicit differentiation. *Advances in neural information processing systems*, 35, 5230–5242.

[13] Guyan, R. J. (1965). Reduction of stiffness and mass matrices. *AIAA journal*, 3(2), 380–380.

[14] Magnus, J. R. (1985). On differentiating eigenvalues and eigenvectors. *Econometric theory*, 1(2), 179–191.

ACKNOWLEDGEMENT

The authors acknowledge the Japan Society for the Promotion of Science for the financial support of this project (Grant-in-Aid for Scientific Research, 21H01527).

COPYRIGHT STATEMENT