Gradient-Based Dovetail Joint Shape Optimization for Stiffness

Xingyuan Sun
Princeton University
Princeton, NJ, USA
xs5@princeton.edu

Chenyue Cai
Princeton University
Princeton, NJ, USA
cc4880@princeton.edu

Ryan P. Adams
Princeton University
Princeton, NJ, USA
rpa@princeton.edu

Szymon Rusinkiewicz
Princeton University
Princeton, NJ, USA
smr@princeton.edu

ABSTRACT

It is common to manufacture an object by decomposing it into parts that can be assembled. This decomposition is often required by size limits of the machine, the complex structure of the shape, *etc.* To make it possible to easily assemble the final object, it is often desirable to design geometry that enables robust connections between the subcomponents. In this project, we study the task of dovetail-joint shape optimization for stiffness using gradient-based optimization. This optimization requires a differentiable simulator that is capable of modeling the contact between the two parts of a joint, making it possible to reason about the gradient of the stiffness with respect to shape parameters. Our simulation approach uses a penalty method that alternates between optimizing each side of the joint, using the adjoint method to compute gradients. We test our method by optimizing the joint shapes in three different joint shape spaces, and evaluate optimized joint shapes in both simulation and real-world tests. The experiments show that optimized joint shapes achieve higher stiffness, both synthetically and in real-world tests.

CCS CONCEPTS

• **Applied computing → Computer-aided manufacturing**.

KEYWORDS

Dovetail Joint, Gradient-based Optimization, End-to-End Differentiable, 3D Printing

ACM Reference Format:
Xingyuan Sun, Chenyue Cai, Ryan P. Adams, and Szymon Rusinkiewicz. 2023. Gradient-Based Dovetail Joint Shape Optimization for Stiffness. In *Symposium on Computational Fabrication (SCF '23), October 8–10, 2023, New York City, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3623263.3623364

1 INTRODUCTION

The dovetail joint has been used in manufacturing for millennia; Figure 1, for example, is a dovetailed box from an Egyptian tomb estimated to have been made circa 1492–1473 B.C. In modern manufacturing, dovetail joints [Ruiz et al. 1984], as shown in Figure 2, are commonly used to connect components in woodworking, turbine blades, 3D printing, *etc.* In woodworking, there are widely-used heuristics regarding the appropriate angles for hardwood versus softwood [Schwarz and Moskowitz 2009], but how can we determine the optimal dovetail shape under more general conditions and with different materials?

Figure 1: Dovetailed box from Hatnefer's tomb, ca. 1492–1473 B.C. (Public domain image via The Metropolitan Museum of Art.)

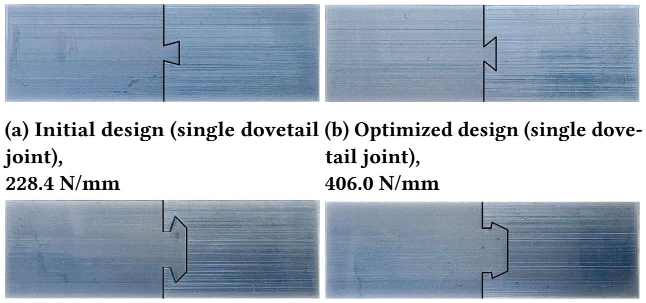
In this work, we study the problem of efficiently finding the optimal dovetail joint design that maximizes the stiffness, given a design space and specified external loads. We choose to study the dovetail joint for its simplicity and ubiquity, but the method we develop should work for any joint shape that is piecewise linear. To find the optimal design, we formalize the task as an optimization problem, with the stiffness as a function of design parameters that characterize the shape of the dovetail joint in the design space. We calculate the gradient of stiffness with respect to the design parameters, and directly optimize them using gradient descent. However, optimizing the shape of a dovetail joint is challenging. First, we need a simulator that can simulate the deformation of the joint given external loads, while considering the contact between two parts of the joint. Second, we need to compute the gradient of stiffness with respect to the design parameters.

To address the first challenge, we perform contact simulations using the penalty approach [Huněk 1993], alternating between the two sides of the joint. In every iteration, we compute the deformation of one side of the joint while considering the other side as rigid, and we penalize the elastic side's penetrating the rigid side. By iteratively applying this alternating approach on the two sides, we have a sufficiently accurate simulator for joint deformation simulation.

As the simulation can be written as partial differential equations (PDEs), the stiffness maximization task can now be understood as a PDE-constrained optimization [De los Reyes 2015]. To perform this optimization, we use the adjoint method [Cao et al. 2003; Errico

SCF '23, October 8–10, 2023, New York City, NY, USA

ACM ISBN 979-8-4007-0319-5/23/10.
https://doi.org/10.1145/3623263.3623364



(a) Initial design (single dovetail joint), 228.4 N/mm

(b) Optimized design (single dovetail joint), 406.0 N/mm

(c) Initial design (complex dovetail joint), 518.1 N/mm

(d) Optimized design (complex dovetail joint), 1036.3 N/mm

Figure 2: Initial and optimized designs of single and complex dovetail joints with average stiffness measured over three batches and external forces applied on the two sides. (a) (c): initial (randomly chosen) single and complex dovetail joints. (b) (d): optimized single and complex dovetail joints, which provide greater stiffness.

1997] to calculate the gradients of stiffness with respect to design parameters.

To demonstrate the effectiveness of our method, we test it both in simulation and in real tests. We first verify the accuracy of the gradients by comparing the adjoint method with the finite difference method. We then run optimization on three different dovetail joint design spaces, each using two different initial designs. Experiments show that optimized designs provide greater stiffness compared to the initial ones, both in simulation and in real tests on 3D-printed structures. We finally study the sensitivity of the optimization result with respect to material parameters by optimizing using different Poisson's ratios. Experiments show that the optimization result is not sensitive to different Poisson's ratios.

2 RELATED WORK

2.1 Dovetail joint shape optimization

Researchers have explored the optimization of dovetail joints using different approaches. The most straightforward approach is simply testing different design parameters. Kogo et al. [2002] and Kogo et al. [2019] conducted tensile and shear tests on carbon-carbon composite dovetail joints with different dovetail angles. Miyauchi et al. [2006] tested wooden dovetail joints with different inclinations and base widths. Jeong et al. [2012] tested different wooden dovetail joints for maximum tension load. Estenlund et al. [2022] studied the dovetail design for mounting coils on rotors by building a simulator and enumerating different dovetail angles. Another approach for dovetail optimization is applying a gradient-free optimizer on a simulator. Hu et al. [2022] tested dovetails with different combinations of tenon length, width, thickness, and angle, and studied the effect of each design parameter using the linear model. Yang et al. [2018] used commercial FEM software for simulation and optimized dovetail shapes for aero-engines using several different gradient-free optimizers. Some researchers build (differentiable) surrogate models for the simulators and optimize the surrogate models. For example, Hahn and Cofer IV [2012] first optimized design parameters using a surrogate model and further used other gradient-free optimizers to optimize design parameters that are sensitive. In this work, instead of enumerating, using gradient-free optimizers or surrogate models, we directly optimize dovetail design parameters in the FEM simulation, using the adjoint method to compute the gradients.

2.2 PDE-constrained optimization

PDE-constrained optimization is a type of constrained optimization problems whose constraints can be written as partial differential equations reflecting the physics that determine the behavior of a system. See, *e.g.*, De los Reyes [2015] for a comprehensive introduction. In a PDE-constrained optimization problem, the objective function depends on both the design variable and the state variable, and the constraints between them can be written as PDEs. There are two major types of approaches to solving PDE constrained optimization problems [Herzog and Kunisch 2010]: *black-box* and *all-at-once*. *all-at-once* methods treat the design variable and the state variable as being independent during optimization, and researchers may use algorithms such as sequential quadratic programming (SQP) [Boggs and Tolle 1995] to solve the problem. *black-box* methods treat only the design as the independent variable during optimization, and researchers may use the adjoint method to calculate the total derivative of the objective with respect to the design variable. See Givoli [2021] for a tutorial on the adjoint method. In this work, we formalize the dovetail joint shape optimization problem as a PDE-constrained optimization task and use the adjoint method with gradient descent to solve it.

3 METHOD

The pipeline of our approach is shown in Figure 3. For an initial set of shape parameters, we first calculate the displacement of the corresponding dovetail joint given a fixed external load on the two sides. We simulate this by employing a contact solver with the penalty approach, alternating between sides while assuming the other side is rigidly fixed in the deformed configuration of the previous iteration. After this alternating solver converges, we use the adjoint method to calculate the derivative of the displacement with respect to design parameters and use line search to find an appropriate step size in the direction of steepest descent.

3.1 Alternating penalty contact simulator

In this subsection, we describe how we simulate the deformation of a specific joint given external loads. Assuming the material is isotropic, we use the linear elastic model from Langtangen and Logg [2017]. Denoting the body as Ω, we have the equations governing the deformation on Ω as

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma} &= f, \\ \boldsymbol{\varepsilon} &= \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\right), \\ \boldsymbol{\sigma} &= \lambda \operatorname{tr}(\boldsymbol{\varepsilon}) I + 2\mu \boldsymbol{\varepsilon}, \end{aligned} \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress tensor, f is the body force (0 in our case), $\boldsymbol{\varepsilon}$ is the strain tensor, $\mathbf{u}$ is the displacement vector, λ and μ are Lamé parameters, and I is the identity matrix. Under the assumption of

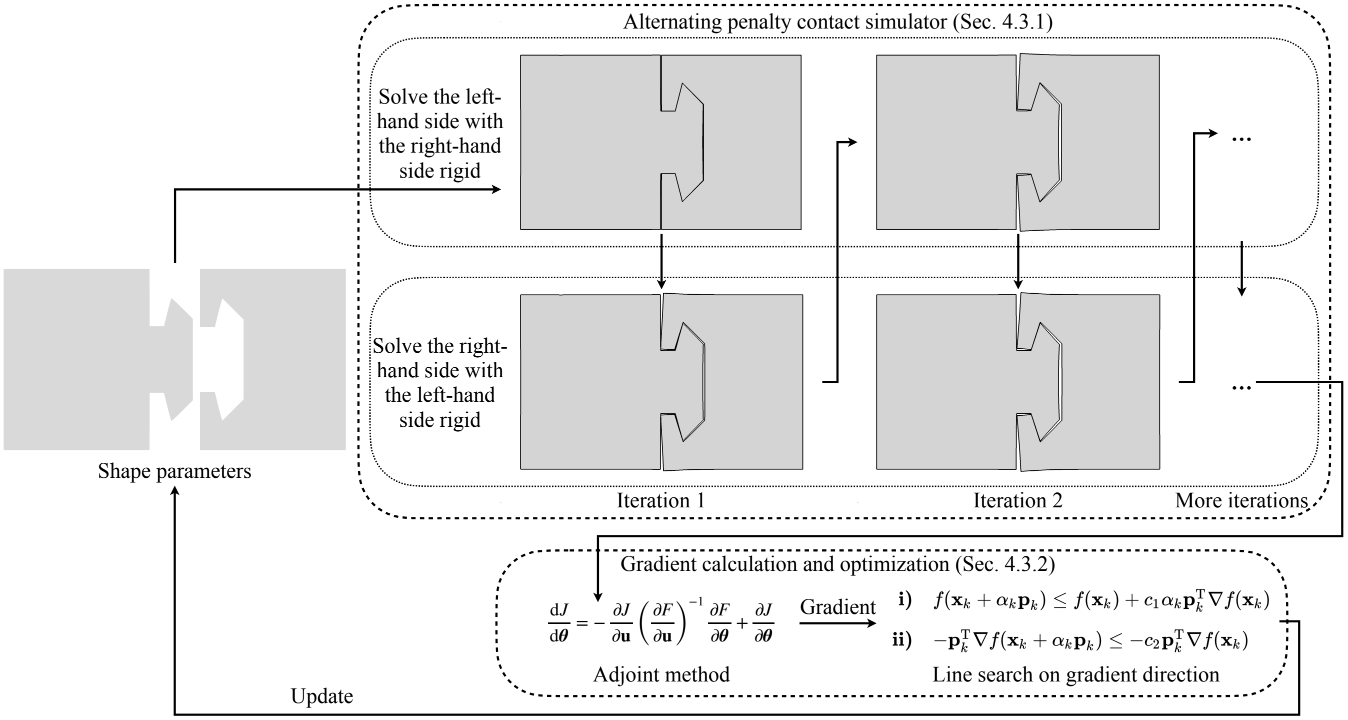

Alternating penalty contact simulator (Sec. 4.3.1)
Solve the left-hand side with the right-hand side rigid
Solve the right-hand side with the left-hand side rigid
...
...
Shape parameters
Iteration 1
Iteration 2
More iterations
Gradient calculation and optimization (Sec. 4.3.2)
$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\theta}} = -\frac{\partial J}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial \mathbf{u}}\right)^{-1}\frac{\partial F}{\partial \boldsymbol{\theta}} + \frac{\partial J}{\partial \boldsymbol{\theta}}$
Gradient
i) $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \mathbf{p}_k^{\mathrm{T}} \nabla f(\mathbf{x}_k)$
ii) $-\mathbf{p}_k^{\mathrm{T}} \nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq -c_2 \mathbf{p}_k^{\mathrm{T}} \nabla f(\mathbf{x}_k)$
Adjoint method
Line search on gradient direction
Update

Figure 3: Given a set of shape parameters, we first use the alternating penalty contact simulator (§ 3.1) to calculate the deformation of the joint, and then use the adjoint method to calculate the gradient and use line search to find the step size for gradient descent (§ 3.2).

planar stress, we have $\lambda = \frac{E\nu}{1-\nu^2}$ and $\mu = \frac{E}{2(1-\nu^2)}$ where E is the Young's modulus of the material (we set it to 1 GPa), and ν is the Poisson's ratio of the material (we set it to 0.4). Equivalently, we are minimizing the total potential energy Π which is defined as [Alnæs et al. 2015]:

$$\Pi \coloneqq \int_{\Omega} \frac{1}{2}\boldsymbol{\varepsilon} : \boldsymbol{\sigma} \mathrm{d}A - \int_{\Omega} f \cdot \mathbf{u} \mathrm{d}A - \int_{\partial\Omega} T \cdot \mathbf{u} \mathrm{d}x, \tag{2}$$

where the colon is the dot product between tensors and T is the traction force. For reasonable deformations, we set T to 0.001 GPa for single dovetail joints and 0.003 GPa for complex and double dovetail joints (see Figure 4). We apply equal traction forces on two sides of the joint as Neumann boundary conditions.

One difficulty is to model the contact between the two sides of the joint. As in Bleyer [2018], we use a penalty approach—solving one side while assuming the other side is rigid, and applying the penalty directly on the displacement field $\mathbf{u}$. Denote the two sides of the joint as Ω_L and Ω_R, and consider the case that we want to solve the deformation on Ω_L while considering Ω_R rigid. For simplicity and due to the piecewise-linear boundary of the joints, we fit lines to all the contacting edges (see § 4.1 for more details) of Ω_R and penalize $\mathbf{u}$ on Ω_L if it collides with the fitted lines. We have our penalized total potential energy Π_L for Ω_L as

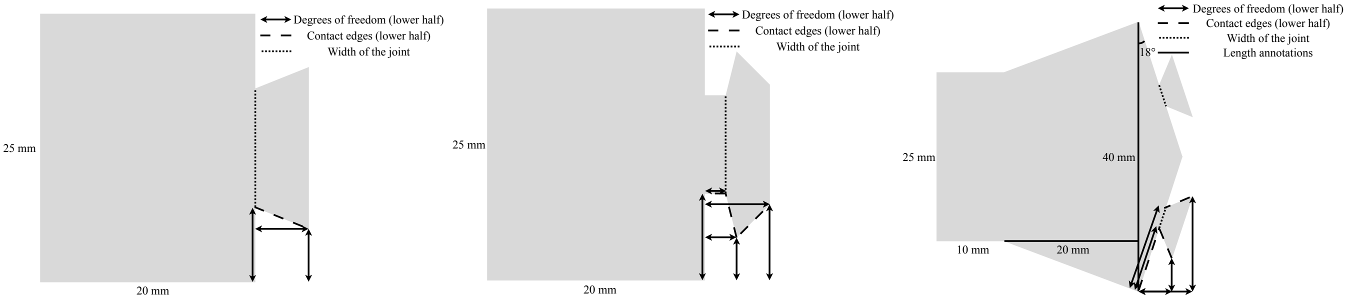
$$\Pi_L \coloneqq \int_{\Omega_L} \frac{1}{2}\boldsymbol{\varepsilon} : \boldsymbol{\sigma} \mathrm{d}A - \int_{\Omega_L} f \cdot \mathbf{u} \mathrm{d}A - \int_{\partial\Omega_L} T \cdot \mathbf{u} \mathrm{d}x \tag{3}$$

$$+ w_{\mathsf{pen}} \cdot \int_{\partial\Omega_L} \texttt{softplus}^2(-\texttt{sdf}(\mathbf{u}; \Omega_L, \Omega_R))\mathrm{d}x, \tag{4}$$

where w_{pen} is the weight of the penalization term, which we set to 1, $\texttt{softplus}(x) = (\ln(1 + \exp(kx))/k)^2$ and k is a scale factor that we set to 50, $\texttt{sdf}$ is the signed distance function and $\texttt{sdf}(\mathbf{u}; \Omega_L, \Omega_R)$ measures the signed distance from the deformed left-hand side to the fitted lines of the deformed right-hand side (positive if outside and negative if inside). We create meshes with a mesh step size of 0.5 mm using pygmsh [Schlömer 2021] and implement the simulator using FEniCS [Alnæs et al. 2015]. We find that four iterations on both sides is usually enough for $\mathbf{u}$ to converge. Note that the simulator works for any piecewise linear joints, not only for dovetail joints.

3.2 Gradient calculation and optimization

In this subsection, we describe how we optimize the shape design parameters θ. With the simulator, we can compute the displacement field $\mathbf{u}(\theta)$. We define the length change of the joint $d(\mathbf{u}(\theta))$ as the difference of average displacement in the horizontal direction between the left and right edges of the joint. Note that maximizing the stiffness is equivalent to minimizing the displacement given fixed traction. We define the optimization objective function $\mathcal{L}(\theta)$


Degrees of freedom (lower half)
Contact edges (lower half)
Width of the joint
25 mm
20 mm
Degrees of freedom (lower half)
Contact edges (lower half)
Width of the joint
25 mm
20 mm
Degrees of freedom (lower half)
Contact edges (lower half)
Width of the joint
Length annotations
18°
25 mm
40 mm
10 mm
20 mm

(a) Single dovetail joint

(b) Complex dovetail joint

(c) Double dovetail joint

Figure 4: Three dovetail joint shape design spaces that are used in the experiments. All designs are top-to-bottom symmetric. (a) *single dovetail joint* is the simplest design; three degrees of freedom. (b) *complex dovetail joint* with a design that is more complex; six degrees of freedom. (c) *double dovetail joint* with two dovetails and a non-vertical boundary in the middle; six degrees of freedom. All contact edges and the width are annotated for every design space.

as

$$\mathcal{L}(\theta) \coloneqq d(\mathbf{u}(\theta)) + w_{\text{min\_l}} \cdot \mathcal{L}_{\text{min\_l}}(\theta) + w_{\text{min\_w}} \cdot \mathcal{L}_{\text{min\_w}}(\theta), \quad (5)$$

where $\mathcal{L}_{\text{min\_l}}(\cdot)$ and $\mathcal{L}_{\text{min\_w}}(\cdot)$ are regularizers, and $w_{\text{min\_l}}$ and $w_{\text{min\_w}}$ are weights for the regularizers, both of which we set to 1. The minimum contact length regularizer penalizes contact between edges that are too short:

$$\mathcal{L}_{\text{min\_l}}(\theta) \coloneqq \sum_{l \in \texttt{contact}(\theta)} \left(\max(\texttt{min\_len} - |l|, 0)\right)^2, \quad (6)$$

where $\texttt{contact}(\theta)$ is the set of all contacting edges (see § 4.1 for more details), $|\cdot|$ measures the length of an edge, and we set `min_len` to 1.5 mm. The minimum width regularization penalty prevents the joint from being too small:

$$\mathcal{L}_{\text{min\_w}}(\theta) \coloneqq \left(\max(\texttt{min\_width} - \texttt{width}(\theta), 0)\right)^2, \quad (7)$$

where $\texttt{width}(\cdot)$ measures the width of a joint (see § 4.1 for more details), and we set `min_width` to 3.5 mm.

To calculate the gradient of the objective function with respect to the design parameters, *i.e.* $\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta}$, we use the adjoint method, which provides the following result:

$$\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta} = -\frac{\partial \mathcal{L}(\theta)}{\partial \mathbf{u}} \left(\frac{\partial F}{\partial \mathbf{u}}\right)^{-1} \frac{\partial F}{\partial \theta} + \frac{\partial \mathcal{L}(\theta)}{\partial \theta}, \quad (8)$$

where $F(\mathbf{u}, \theta) = 0$ is the PDE corresponding to the simulator. We implement the automatic gradient calculation using dolfin-adjoint [Dokken et al. 2020; Mitusch et al. 2019] and PyTorch [Paszke et al. 2019].

To optimize θ, we use gradient descent. For every step, we perform a line search along the gradient direction using SciPy [Virtanen et al. 2020], finding a step size that satisfies strong Wolfe conditions [Wolfe 1969, 1971]. If the line search fails, we randomly sample a step size from $\mathcal{N}(0, 0.5^2)$, where a negative step size indicates moving in the opposite direction. This approach introduces controlled noise to prevent entrapment in local minima. We perform 15 optimization steps and keep the step with minimum $d(\mathbf{u}(\theta))$. Lastly, to prevent landing in design parameters that are sensitive to manufacturing errors, every time we evaluate the objective function or its gradient, we apply independent random noise sampled from $\mathcal{N}(0, 0.01^2)$ to every dimension of θ three times and take the average.

4 EXPERIMENTS

In this section, we study the effectiveness of our method. We present different dovetail joint design spaces for all the experiments in § 4.1. We then provide simulated results (§ 4.2) and check the correctness of gradients on mesh vertex coordinates (§ 4.3). We show the main result—the optimization results in § 4.4, evaluating them both synthetically and in real experiments. Finally, to better understand the effect of material parameters, we show the sensitivity test of optimization results with respect to the Poisson's ratio in § 4.5.

4.1 Shapes for experiments

As shown in Figure 4, we use three different joint design spaces for all the experiments. We only visualize the left part of the joint, as the right part is complementary to it. All joints are top-to-bottom symmetric, so all simulation is only performed on the lower half of the mesh for lower computational cost. The first design space is named *single dovetail joint* (Figure 4a), which contains the simplest form of a dovetail joint and three degrees of freedom. The second design space is called *complex dovetail joint* (Figure 4b), which still contains only one dovetail shape but with a structure that is more complex and has six degrees of freedom. The third design space is *double dovetail joint* (Figure 4c), which contains two dovetail structures, a non-vertical boundary in the middle, and six degrees of freedom. We annotate all the contacting edges for each design space, which are used in the contact penalizer and the minimum contact length regularizer. We also label the width of every design space, which is used in the minimum width regularizer.

4.2 Alternating penalty contact simulator results

In this subsection, we visualize simulation results from the alternating penalty contact simulator to demonstrate it is producing reasonable results, and the results are shown in Figure 5. In the first iteration, we undertake the two-step process: first, solving the

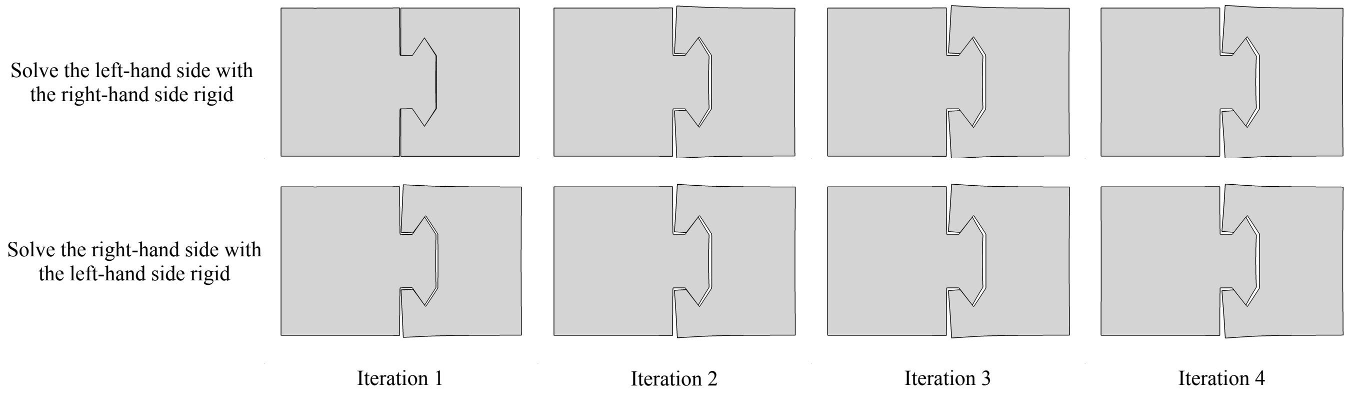
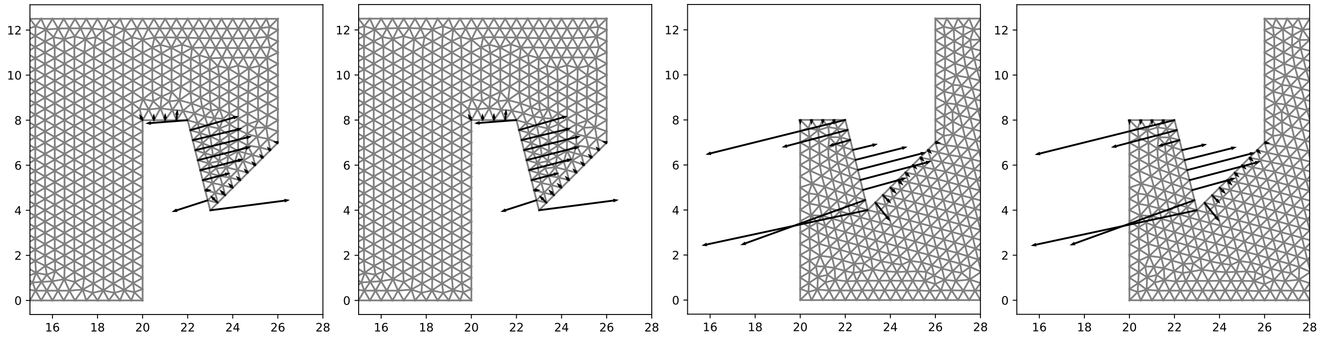

Solve the left-hand side with the right-hand side rigid
Solve the right-hand side with the left-hand side rigid
Iteration 1
Iteration 2
Iteration 3
Iteration 4

Figure 5: Simulation results on a specific dovetail joint design. The alternating simulator produces reasonable results as the number of iterations increases, and the results converge in the end.



(a) Gradients on the left half; adjoint method **(b) Gradients on the left half; finite difference** **(c) Gradients on the right half; adjoint method** **(d) Gradients on the right half; finite difference**

Figure 6: Gradient directions on three contacting edges calculated using the adjoint method and the finite difference method. The two results are indistinguishable, which indicates the gradient calculation is correct.

left-hand side while keeping the right-hand side fixed, leading to a slight leftward movement of the left side. Subsequently, in the same iteration, we address the right-hand side while maintaining the left side in a fixed state, causing the dovetail section on the right side to bend in opposite directions. In the second iteration, we observe a progressive leftward pull of the left-hand side when the right-hand side remains fixed, followed by increased bending of the right-hand side, consequently creating more space for the left-hand side to experience a further leftward pull in the third iteration. With each iteration, the results evolve and eventually converge as the number of iterations increases.

4.3 Gradient correctness check

In this subsection, we check the correctness of our gradient computation. We calculate the derivative of the displacement with respect to the coordinates of vertexes from the mesh in two different approaches: the adjoint method and the finite difference method, as shown in Figure 6. For the finite difference method, we use a step size of 10^{-4}. We only visualize gradients on three edges, as interior gradients should be all zero. The two sets of gradients are indistinguishable, which infers our gradient calculation is consistent with the finite difference method. The average relative difference from the adjoint method results to the finite difference results is 1.82×10^{-4}, which is negligibly small.

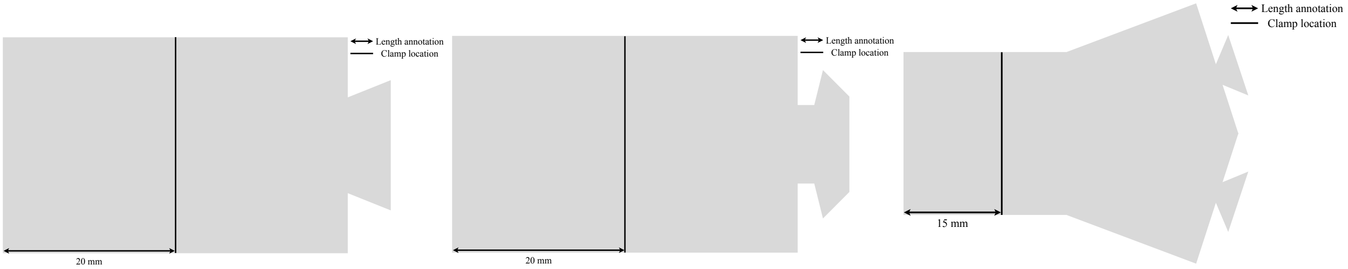
4.4 Shape optimization for stiffness

Fabrication and experimental setup. We use Fusion 360[1] to draw 3D shapes and CHITUBOX[2] and UVtools[3] for slicing. We use the ELEGOO Saturn S resin 3D printer to print laminates of ABS-like resin with a height of 5 mm. For the printed parts to be assemblable, we introduce a gap of 0.1 mm between the joints. To measure the stiffness of a joint, we use an Instron 600DX universal testing machine, which is set to move at a speed of 20 mm/min until the tested object breaks and produces a position-load curve. We also include tabs on two sides of the joint for the machine to clamp, as shown in Figure 7: the parts on the left-hand side of the solid line are the tabs, and the solid lines indicate the position of the clamp.

[1] https://www.autodesk.com/products/fusion-360

[2] https://www.chitubox.com/

[3] https://github.com/sn4k3/UVtools



(a) Single dovetail joint **(b) Complex dovetail joint** **(c) Double dovetail joint**

Figure 7: Tabs are added in real printings such that the universal testing machine can clamp the printed parts.

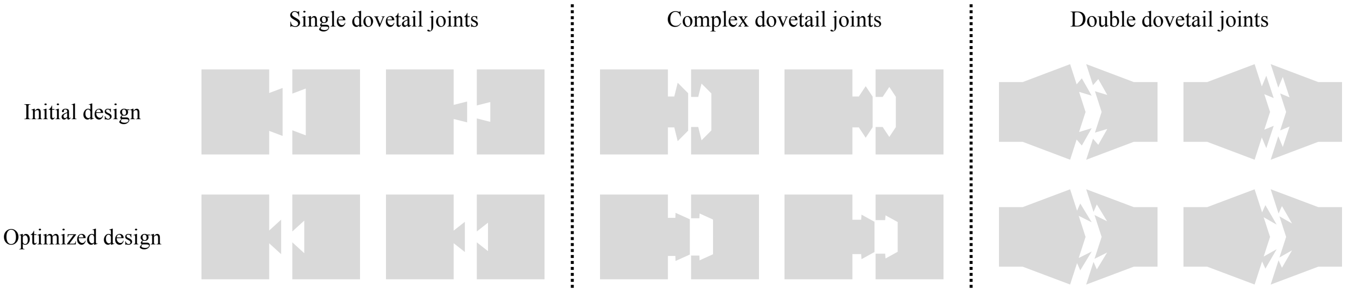


Figure 8: Initial and optimized designs of dovetail joints. The optimized designs are similar for the same design space though the initializations are very different.

Table 1: Real measured stiffness of initial and optimized designs over three batches. The significant increase in stiffness indicates that our algorithm successfully optimizes the joint shape design.

Joint type	Initial design	Optimized design
Single dovetail joint	219.9±22.6	**362.3±26.9**
	228.4±53.2	**406.0±15.1**
Complex dovetail joint	518.1±29.2	**1036.3±61.6**
	553.3±55.7	**1120.0±102.4**
Double dovetail joint	380.6±23.8	**527.4±48.3**
	360.2±12.1	**611.8±64.0**

Optimization and test results. For each design space, we randomly select two different initial sets of shape parameters and optimize them for fifteen gradient descent steps. We select the iteration with the smallest simulated displacement, and the results are shown in Figure 8. The optimized results are similar though the initializations are quite different. The simulated results are shown in Figure 9. We print three copies of each design and test them on the universal testing machine with position-load curves recorded. We measure the stiffness using the position change between a load of 30 N and a load of 60 N, and the results are listed in Table 1. The significant differences between the initial and optimized designs indicate the effectiveness of our algorithm.

4.5 Poisson's ratio sensitivity test

One interesting question raised by our method is the effect of material parameters on the optimization results. As Young's modulus is obviously not affecting our optimization process (if simultaneously increasing the traction), we study whether changing Poisson's ratio would change the results, and the results are shown in Figure 10. As different Poisson's ratios produce almost the same optimization result, we observe that the results are not sensitive to Poisson's ratio.

5 DISCUSSION

In this project, we studied the task of dovetail joint shape optimization to maximize its stiffness, and, as existing works, we formalized the task as an optimization problem, viewing the stiffness of the joint as a function of shape design parameters. Existing works use search algorithms, gradient-free optimizers, or surrogate models for optimization, which have limited efficiency. To use gradient-based optimizers, we first built our own contact simulator by alternatively simulating the deformation of one side of the joint while considering the other side as rigid, using the penalty approach. We then use the adjoint method and gradient descent for optimization. For experiments, we first verified the gradients were correct by comparing them to the gradients calculated from the finite difference method. We then tested the optimized joint shapes on different initializations in different dovetail shape design spaces, both synthetically and in real experiments, showing that the optimized joints are much stiffer than the initial ones. Note that our method is not

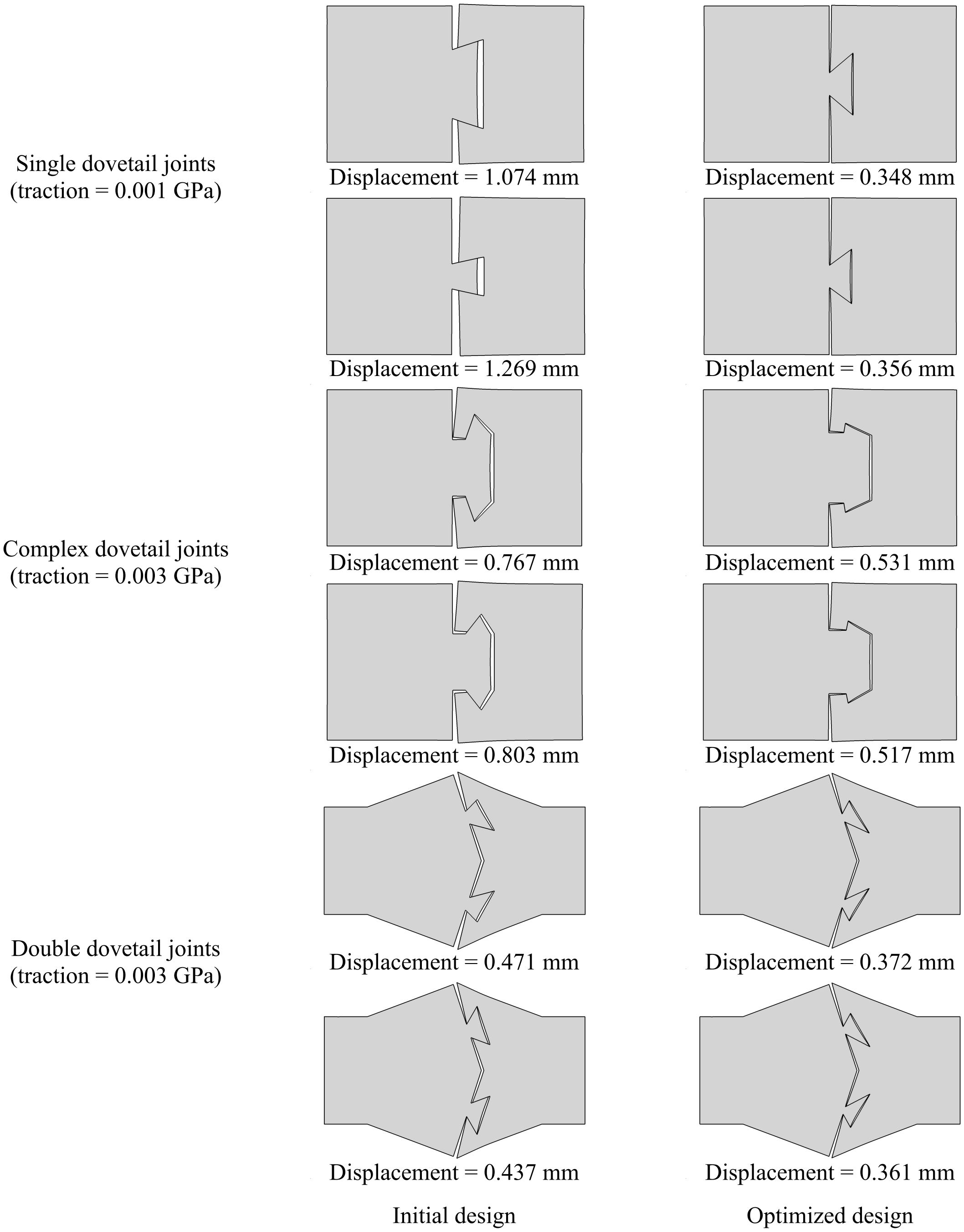


Figure 9: Simulated results and displacements of initial and optimized designs. The displacements of the optimized designs are much smaller than those of the initial designs, which indicates the optimization algorithm is working effectively.

restricted to dovetail joint shape optimization but works for joints with piecewise linear joint boundaries.

We would also like to discuss some limitations and future directions of our approach. First, the simulator has limited accuracy as several assumptions and simplifications are made, *e.g.*, plane stress, fitting boundaries using lines for the penalty term, *etc.* Real experiments showed that the simulator is still reasonably accurate, but future work can be done on more accurate simulators. Besides, the optimization is not in real-time, and most optimization (15 gradient descent steps) in this project takes 10 to 20 minutes to finish on a laptop computer. However, compared to the manufacturing time, this is acceptable, and there are amortized approaches that

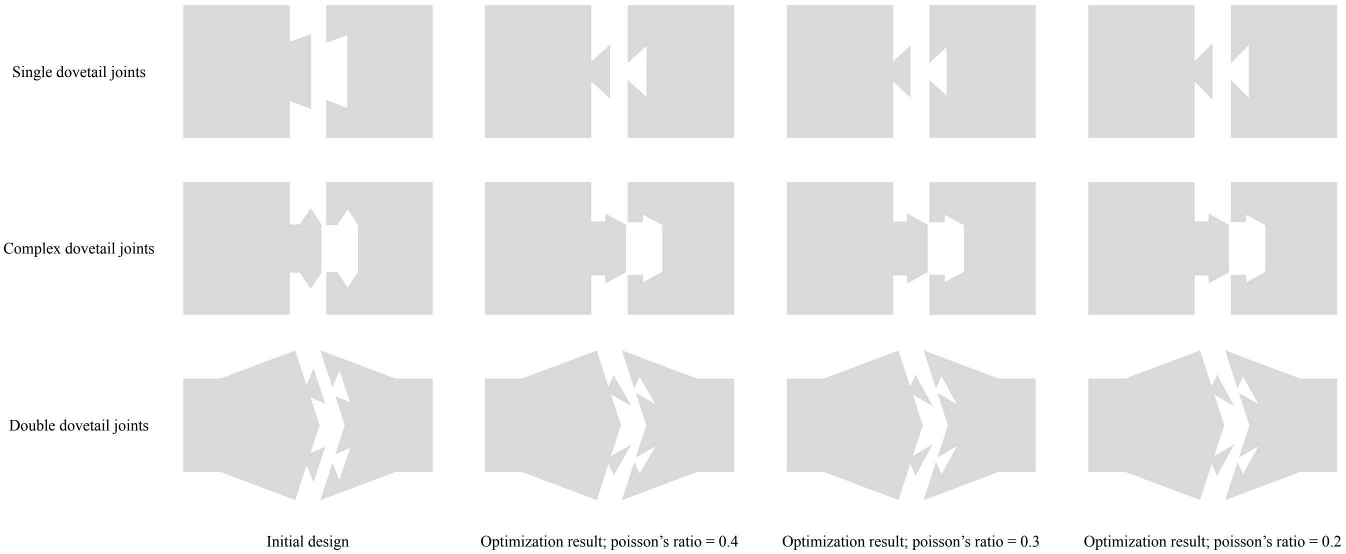

Single dovetail joints
Complex dovetail joints
Double dovetail joints
Initial design
Optimization result; poisson's ratio = 0.4
Optimization result; poisson's ratio = 0.3
Optimization result; poisson's ratio = 0.2

Figure 10: Optimization results from the same initial design but using different Poisson's ratios. As the difference is negligible, we observe that the optimization results are not sensitive to Poisson's ratio.

can significantly reduce the running time. Finally, the optimized results from gradient-based optimizers can be local minima. Possible solutions include using different initializations, introducing randomness during optimization, *etc.*

ACKNOWLEDGMENTS

We would like to thank Eder Medina for setting up the resin 3D printer, members of the Princeton Laboratory for Intelligent Probabilistic Systems for valuable discussions and feedback, Joseph Vocaturo, and the Princeton Department of Civil and Environmental Engineering for providing the universal testing machine. This work is partially supported by the Princeton School of Engineering and Applied Science, as well as the U. S. National Science Foundation under grants #IIS-1815070, #IIS-2007278, and #OAC-2118201.

REFERENCES

Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. 2015. The FEniCS project version 1.5. *Archive of Numerical Software* 3, 100 (2015).

Jeremy Bleyer. 2018. *Numerical Tours of Computational Mechanics with FEniCS.* https://doi.org/10.5281/zenodo.1287832

Paul T Boggs and Jon W Tolle. 1995. Sequential quadratic programming. *Acta numerica* 4 (1995), 1–51.

Yang Cao, Shengtai Li, Linda Petzold, and Radu Serban. 2003. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution. *SIAM journal on scientific computing* 24, 3 (2003), 1076–1089.

Juan Carlos De los Reyes. 2015. *Numerical PDE-constrained optimization.* Springer.

Jørgen S Dokken, Sebastian K Mitusch, and Simon W Funke. 2020. Automatic shape derivatives for transient PDEs in FEniCS and Firedrake. *arXiv preprint arXiv:2001.10058* (2020).

Ronald M Errico. 1997. What is an adjoint model? *Bulletin of the American Meteorological Society* 78, 11 (1997), 2577–2592.

Samuel Estenlund, Alexandra Tokat, Jonas Engqvist, and Mats Alaküla. 2022. Dovetail Design for Direct Cooled Rotor: Design and Manufacturing. In *2022 International Conference on Electrical Machines (ICEM)*. IEEE, 2121–2127.

Dan Givoli. 2021. A tutorial on the adjoint method for inverse problems. *Computer Methods in Applied Mechanics and Engineering* 380 (2021), 113810.

Youngwon Hahn and John I Cofer IV. 2012. Design study of dovetail geometries of turbine blades using abaqus and isight. In *Turbo Expo: Power for Land, Sea, and Air*, Vol. 44731. American Society of Mechanical Engineers, 11–20.

Roland Herzog and Karl Kunisch. 2010. Algorithms for PDE-constrained optimization. *GAMM-Mitteilungen* 33, 2 (2010), 163–176.

Wengang Hu, Runzhong Yu, Mengyao Luo, and Arif Caglar Konukcu. 2022. Study on tensile strength of single dovetail joint: experimental, numerical, and analytical analysis. *Wood Material Science & Engineering* (2022), 1–9.

I Huněk. 1993. On a penalty formulation for contact-impact problems. *Computers & structures* 48, 2 (1993), 193–203.

GY Jeong, MJ Park, JS Park, and KH Hwang. 2012. Predicting load-carrying capacity of dovetail connections using the stochastic finite element method. *Wood and Fiber Science* (2012), 430–439.

Y Kogo, H Hatta, T Sugibayashi, and M Toyoda. 2019. Application of Carbon-Carbon Composites to Dovetail Joint Structures. In *Proceedings of the Eighth Japan-US Conference on Composite Materials.* CRC Press, 693–702.

Yasuo Kogo, Hiroshi Hatta, Masaji Toyoda, and Toshio Sugibayashi. 2002. Application of three-dimensionally reinforced carbon–carbon composites to dovetail joint structures. *Composites science and technology* 62, 16 (2002), 2143–2152.

Hans Petter Langtangen and Anders Logg. 2017. *Solving PDEs in python: the FEniCS tutorial I.* Springer Nature.

Sebastian K Mitusch, Simon W Funke, and Jørgen S Dokken. 2019. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software* 4, 38 (2019), 1292.

Kazunari Miyauchi, Minoru Masuda, and Koji Murata. 2006. Analysis of strain distributions of wooden dovetail joints using digital image correlation method. *JOURNAL-SOCIETY OF MATERIALS SCIENCE JAPAN* 55, 4 (2006), 367.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems.* 8026–8037.

C Ruiz, PHB Boddington, and KC Chen. 1984. An investigation of fatigue and fretting in a dovetail joint. *Experimental mechanics* 24, 3 (1984), 208–217.

Nico Schlömer. 2021. pygmsh: A Python frontend for Gmsh. *GPL-3.0 License* (2021).

Christopher Schwarz and Joel Moskowitz. 2009. *The Joiner and Cabinet Maker: His Work and Its Principles.* Lost Art Press.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.

Philip Wolfe. 1969. Convergence conditions for ascent methods. *SIAM review* 11, 2 (1969), 226–235.

Philip Wolfe. 1971. Convergence conditions for ascent methods. II: Some corrections. *SIAM review* 13, 2 (1971), 185–188.

Tian Yuan Yang, Duo Qi Shi, and Zhen Cheng. 2018. 2D Geometrical Parameters Optimization Design Method of CMC/Metal Dovetail Joint. In *Materials Science Forum*, Vol. 923. Trans Tech Publ, 156–163.