

SIMULTANEOUS PACKING AND ROUTING OPTIMIZATION OF THERMAL-FLUID
SYSTEMS

BY
ALEXANDER JESSEE

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:

Assistant Professor Kai James

ABSTRACT

This study investigates the computational design of thermo-mechanical systems. The optimization simultaneously finds the best placement of devices as well as placement and size of routing segments. The layout must satisfy geometric constraints to avoid interference between components. Constraints based on physics models are also included. Physics constraints are important to designing a system feasible for real world use, but are often overlooked in current methods for routing optimization. A thermal conduction finite element model based on geometric projection is presented. Pressure drop of the fluid in the routing is modeled with a lumped parameter pipe flow model. A parameterization is developed that allows the use of gradient based optimization methods. Analytical sensitivities of the physics models and geometric constraints with respect to the parameterization are derived. A simple system is optimized using three different objective functions resulting in three significantly different layouts, demonstrating the importance of including physics in the optimization problem.

ACKNOWLEDGMENTS

I would like to thank my adviser Professor Kai James for his support and guidance of my research. I would also like to acknowledge Professor James Allison and his students Danny Lohan and Satya Peddada for their vision on the applications of this project and their work to make it successful. To my labmates, Anurag, Kenny, Prateek, and Troy thank you for all the discussion and help overcoming challenges large and small over the last two years. Finally, I would like to thank my parents for always encouraging me in pursuit of whatever goals I have.

TABLE OF CONTENTS

Chapter 1	Introduction	1
Chapter 2	Physics models	3
2.1	Steady state thermal conduction	3
2.2	Lumped parameter pipe flow model	8
Chapter 3	Optimization problem and sensitivity analysis	12
3.1	Objective and physics constraints	15
3.2	Geometric constraints	21
Chapter 4	Results	24
Chapter 5	Conclusion	31
References		32
Appendix A	Distance between segments MATLAB code	36

Chapter 1

Introduction

A common type of system consists of several devices that each perform a task, and some routing which connects the devices to transfer energy between them. Examples of these systems are power electronics circuits, printed circuit boards, and vehicle cooling systems. Electrified vehicles may even have a power electronics system which requires an active cooling system thus utilizing two systems of this type. These systems have many requirements, from performance to cost to geometry and volume restrictions. Methods exist for automating the design of routing using algorithmic [16, 26] or optimization [18, 17, 27] methods. These have two shortcomings though. First, they only consider the placement of routing, not the placement of devices, and second, they only consider the most basic requirement of the routing not intersecting between components. The evaluation of other system requirements including ones that require physics simulations is left to human designers. The amount of time required for a designer to generate a feasible design and analyze its performance limits the ability to explore the design space within a project timeline. This can lead to a system design which, although it meets requirements, may not be optimal when considering all aspects of the system. In this research computational methods will be used to generate designs faster and with better performance than systems designed with the need for significant human input.

The first attempts at automating the design of component - routing systems focused on finding optimal pipe routing when the location of the components that are being connected is known. A ground structure approach has been used to minimize pressure loss in a pipe network [8]. In the ground structure method a set of nodes is connected by pipe segments and the diameter of each segment is optimized. This method has also been used in structural problems to optimize trusses [1]. More recently genetic algorithm [18] and ant colony optimization [17, 27] methods have been used to place routing between components in aircraft engines and oil industry equipment. The ant colony method has also been applied to the problem of routing electrical wires in buildings [10].

Topology optimization, where the placement of material in a 2D or 3D domain is optimized, has

been steadily improving. One of the most popular methods of parameterizing the design space is the SIMP method [20, 12]. This method has been used to design structures for maximum stiffness [21] or conducting heat [6, 2]. Topology optimization has also been used to optimize the placement of components and their supporting structure [32, 31]. This allows sections of specific geometry, such as a pattern of bolt holes, to be optimally distributed within a structure. The designs produced by these methods often have the appearance of organic structures. With the rise of additive manufacturing these structures have become feasible to produce [28]. There is still interest in optimizing designs made from stock materials, typically using ground structure methods [25, 30], although projection methods have also been suggested for solving these problems [14, 29].

Topology optimization of problems that include multiple different physics domains has also been studied. The aerodynamic shape and internal structure of a wing has been simultaneously optimized [7, 3, 15]. Additionally, de Kruijf *et al* and Takezawa *et al* performed optimization which included both structural and thermal conduction requirements [9, 23]. These studies solved structural and thermal finite element analysis problems and utilized the SIMP method. A method to optimize pipe routing in an aircraft engine for an objective involving pressure loss and vibration response was proposed by Liu and Jiao [11]. Equations were suggested to model the smoothness and vibration response of a piping section. The vibration and smoothness models did not involve solving a continuum problem in the optimization process.

The optimal layout of routing has been undertaken in some of the research cited above. The research presented here develops a novel technique to simultaneously optimize both devices and routing. The optimization also goes beyond current methods by including continuum and lumped parameter physics models as an objective function or constraint, in addition to purely geometric equations. Sensitivity analysis is performed on these equations to enable the use of gradient based optimization methods. Examples presented later in this paper consider thermal conduction on the continuum level using the finite element method, and a lumped parameter pipe flow model. However, the methods presented here could be extended to model other combinations of physics, for example thermal-electric or structural-fluid systems. Chapter 2 will present the models used to simulate the physics response of the system. Chapter 3 will state the optimization problem and sensitivity analysis. Finally, results of optimizing systems with this method are shown in Chapter 4.

Chapter 2

Physics models

2.1 Steady state thermal conduction

Temperature distribution at any point \mathbf{x} in the domain will be modeled using the finite element method. The strong form of the boundary value problem for heat conduction is given by

$$\nabla \cdot (\boldsymbol{\kappa} \nabla T(\mathbf{x})) + Q = 0, \quad \mathbf{x} \text{ in } \Omega \quad (2.1)$$

$$T(\mathbf{x}) = T^*, \quad \mathbf{x} \text{ on } \Gamma_T \quad (2.2)$$

$$\mathbf{n} \cdot (\boldsymbol{\kappa} \nabla T(\mathbf{x})) = q^*, \quad \mathbf{x} \text{ on } \Gamma_q \quad (2.3)$$

Here $\boldsymbol{\kappa}$ is the matrix of thermal conduction coefficients, $T(\mathbf{x})$ is the temperature solution field, Q is heat flux per unit volume in the domain, and \mathbf{n} is the unit normal to the domain boundary. Temperature, T^* , and heat flux, q^* , boundary conditions are applied on Γ_T and Γ_q portions of the domain boundary respectively. The Galerkin weighted residual method will be used to construct a weak form of the problem. A weight function, w , is applied to satisfy (2.1) in a weighted average sense.

$$\int_{\Omega} w(\nabla \cdot \boldsymbol{\kappa} \nabla T + Q) d\Omega = 0 \quad (2.4)$$

The divergence theorem is applied to the first term of (2.4) to give the final weak form.

$$-\int_{\Omega} \nabla w \cdot \boldsymbol{\kappa} \nabla T d\Omega + \int_{\partial\Omega} w \underbrace{\mathbf{n} \cdot \nabla T}_{q^*} d\partial\Omega + \int_{\Omega} wQ d\Omega = 0 \quad (2.5)$$

If the heat flux on the boundary is due to convection q^* can be replaced with

$$q^* = h(T - T_{env}), \quad (2.6)$$

where h is the convection coefficient and T_{env} is the temperature of the convecting fluid.

The problem is then discretized using isoparametric finite elements. The vector \mathbf{T} contains the local temperatures at the nodes of the finite element mesh. These are the values that the finite element problem will solve. Element shape functions, \mathbf{N} , and shape function gradients, \mathbf{B} , are selected to interpolate the solution from nodal solutions and serve as weighting functions. Making these substitutions $w = \mathbf{N}$, $T = \mathbf{N}\mathbf{T}$, and $\nabla T = \mathbf{B}\mathbf{T}$ the finite element equation is formed.

$$\underbrace{\left(\int_{\Omega} \mathbf{B}^T \boldsymbol{\kappa} \mathbf{B} d\Omega - \int_{\Gamma_q} h \mathbf{N} \mathbf{N}^T d\Gamma_q \right)}_{\mathbf{K}} \mathbf{T} = \underbrace{\int_{\Omega} Q \mathbf{N} d\Omega + \int_{\Gamma_q} h T_{env} \mathbf{N} d\Gamma_q}_{\mathbf{P}} \quad (2.7)$$

Here \mathbf{K} is called the global stiffness matrix and \mathbf{P} is called the global load vector. Equation (2.8) results in a linear algebra problem which can be solved for nodal temperatures.

$$\mathbf{K}\mathbf{T} = \mathbf{P} \quad (2.8)$$

Shape functions are defined such that they are zero everywhere outside of the element, so the global stiffness matrix and load vector are assembled element by element. The element stiffness matrix is shown in equation (2.9) and the element load vector is shown in (2.10).

$$\mathbf{k}_{el} = \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\kappa} \mathbf{B} d\Omega_e - \int_{\partial\Omega_h} h \mathbf{N} \mathbf{N}^T d\partial\Omega_h \quad (2.9)$$

$$\mathbf{p}_{el} = \int_{\Omega_e} Q \mathbf{N} d\Omega_e + \int_{\partial\Omega_h} h T_{env} \mathbf{N} d\partial\Omega_h \quad (2.10)$$

The element equations (2.9) and (2.10) are integrated over the element domain, Ω_e , and the element boundary with convection boundary condition, $\partial\Omega_h$. Details on implementing the finite element solution, such as element shape functions and integrating element equations, can be found in [5].

In this research the geometric projection method of Norato et al [14] will be used. This method is an alternative to the SIMP method for parameterizing an optimization problem. In the original work, the method was used to optimize structures while ensuring that the resulting design could be made from stock materials such as structural beams. This method is advantageous for the routing and packaging optimization problem as well. The geometric projection method will create designs that can be manufactured out of standard circular pipes, while using a set of design variables that

fit well with the lumped parameter model in section 2.2. The remainder of this section will review the geometric projection method, and detail changes made for use with the routing and packaging problem.

In the geometric projection method, each element in the mesh is given a density parameter ρ_i corresponding to the ratio of solid material or void. Density of 1 corresponds to fully solid material in the element and 0 corresponds to fully void. The material properties for each element stiffness matrix are then scaled by the element density parameter. Leaving out the convection boundary condition term, the element stiffness matrix is

$$\mathbf{k}_i = (\rho_{min} + (1 - \rho_{min})\rho_i^p) \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\kappa} \mathbf{B} d\Omega \quad (2.11)$$

$$= (\rho_{min} + (1 - \rho_{min})\rho_i^p) \mathbf{k}_0 \quad (2.12)$$

where p is a penalization parameter used to penalize intermediate densities between 1 and 0 leading to a projection with less gray area between solid and void. The convection boundary condition term is left out because it will not be scaled by the density parameter. If a regular mesh with all elements being the same shape and size is used, then equation (2.12) can be applied since the integral term is the same for all elements. A minimum density, ρ_{min} , is enforced to prevent singularity in the global stiffness matrix. For structural problems, the smallest ρ_{min} that prevents ill conditioning is used. In thermal problems though, a minimum density can be chosen to simulate the thermal conductivity of the surrounding medium, for example air. The density of each element is found by projecting a geometric shape onto the mesh. Norato et al proposed bars with rounded ends as a shape which could be projected easily, and the same will be used here. Each bar will have three parameters, segment start and end points \mathbf{x}_0 and \mathbf{x}_f , and bar width w , see figure 2.1. The parameter for out of plane thickness is left out here because we will not allow bars to be removed. The bars will form a flow network and if they were allowed to disappear the flow would be broken. The signed distance between a bar q and an element with center at \mathbf{p} is

$$\phi_q(d_q(\mathbf{x}_{q_0}, \mathbf{x}_{q_f}, \mathbf{p}), w) = d_q(\mathbf{x}_{q_0}, \mathbf{x}_{q_f}, \mathbf{p}) - \frac{w}{2} \quad (2.13)$$

where d_q is the distance between the segment q and point \mathbf{p} . This distance depends on whether \mathbf{p} is closest to an endpoint or the connecting line. The distance equation (2.14) is replicated here from

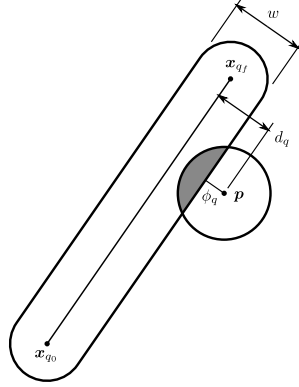


Figure 2.1: Bar projection

Norato et al [14].

$$d_q = \begin{cases} \|b\| & \text{if } \mathbf{a} \cdot \mathbf{b} \leq 0 \\ \|g\| & \text{if } 0 < \mathbf{a} \cdot \mathbf{b} < \mathbf{a} \cdot \mathbf{a} \\ \|e\| & \text{if } \mathbf{a} \cdot \mathbf{b} \geq \mathbf{a} \cdot \mathbf{a} \end{cases} \quad (2.14)$$

Where

$$\mathbf{a} = \mathbf{x}_{q_f} - \mathbf{x}_{q_0} \quad (2.15)$$

$$\mathbf{b} = \mathbf{p} - \mathbf{x}_{q_0} \quad (2.16)$$

$$\mathbf{e} = \mathbf{p} - \mathbf{x}_{q_f} \quad (2.17)$$

$$\mathbf{g} = \left[\mathbf{I} - \frac{1}{\|\mathbf{a}\|^2} \mathbf{a} \otimes \mathbf{a} \right] \mathbf{b} \quad (2.18)$$

A circle of radius r is placed at the element center. The density assigned to each element is the area of the circle covered by the bar divided by total area of the circle, see shaded area of figure 2.1. The density as a function of signed distance is shown below.

$$\rho_q(d_q(\mathbf{x}_{q_0}, \mathbf{x}_{q_f}, \mathbf{p}), r) = \begin{cases} 0 & \phi_q > r \\ \frac{1}{\pi r^2} \left[r^2 \cos^{-1} \left(\frac{\phi_q(d_q)}{r} \right) - \phi_q(d_q) \sqrt{r^2 - \phi_q(d_q)^2} \right] & -r \leq \phi_q \leq r \\ 1 & \phi_q < -r \end{cases} \quad (2.19)$$

In the routing and packaging optimization problem there are also devices which need to be modeled in the finite element analysis. Devices will be approximated as polygonal shapes with



Figure 2.2: Device projection

straight edges. Each device will be defined by a reference point, \mathbf{c}_d , and a set of vectors, \mathbf{b}_i , pointing from the reference point to polygon vertices. The device densities are calculated by first projecting each edge of the polygon and then filling in elements inside the polygon with density of 1, see figure 2.2. Densities for each edge $\tilde{\rho}_e$ are calculated using (2.19). End points of edge segments, \mathbf{x}_{e_0} and \mathbf{x}_{e_f} , are found by:

$$\mathbf{x}_{e_0} = \mathbf{x}_i = \mathbf{c}_d + \mathbf{b}_i \quad (2.20)$$

$$\mathbf{x}_{e_f} = \mathbf{x}_{i+1} = \mathbf{c}_d + \mathbf{b}_{i+1} \quad (2.21)$$

The densities of all the edges in the device are then merged by using a p-norm approximation of the maximum density.

$$\rho_d(\mathbf{c}_d, \mathbf{p}) = \left[\sum_{e=1}^{N_e} (\tilde{\rho}_e(d_e(\mathbf{x}_{e_0}, \mathbf{x}_{e_f}, \mathbf{p})))^p \right]^{\frac{1}{p}} \quad (2.22)$$

After merging edge densities, all elements with centers inside the polygon are assigned $\rho_d = 1$. Elements with centers inside the polygon can be found using the MATLAB function `inpoly()` or using the algorithm described in [4].

Finally the density used in equation (2.12) is calculated by merging densities of all bars and devices.

$$\rho_i = \left[\sum_{q=1}^{N_q} (\rho_q(d_q(\mathbf{x}_{q_0}, \mathbf{x}_{q_f}, \mathbf{p}_i)))^p + \sum_{d=1}^{N_d} (\rho_d(\mathbf{c}_d, \mathbf{p}_i))^p \right]^{\frac{1}{p}} \quad (2.23)$$

Devices may also add or take heat from the domain. The projection in equation (2.22) will be used to accomplish this. Rather than assuming a constant internal heat generation Q , each device will

have its own Q_d value. Element load vectors are then modified using this Q_d and the device density.

$$\mathbf{p}_e = \sum_{d=1}^{N_d} \rho_{de}^p Q_d \int_{\Omega_e} \mathbf{N} d\Omega_e \quad (2.24)$$

$$= \sum_{d=1}^{N_d} \rho_{de}^p Q_d \mathbf{p}_0 \quad (2.25)$$

The convection boundary condition term of equation (2.10) is left out again as it will not be scaled with density.

2.2 Lumped parameter pipe flow model

Consider a power electronics system with liquid cooling. The pump used to move fluid through the cooling loop will consume power that could be used elsewhere, and it will add to the volume and weight of the system. It is of interest to minimize the pressure drop in the cooling loop to reduce the power demand on the pump. Alternatively a pump may already be selected and the power consumption accounted for. In this case we wish to optimize some other aspect of the system while constraining the pressure drop below the rated value for the pump. Both cases will be explored in chapter 4 but first, calculation of pressure losses in a flow loop will be presented here.

The cooling loop will be modeled using a lumped parameter pipe flow model. The lumped parameter model uses empirical relations to model sections of the flow loop with only a few parameters [19]. The lumped parameter model is relatively fast to solve and has few parameters compared to computational fluid dynamics methods, making the lumped parameter model ideal for optimization problems. Several assumptions are made for modeling pipe flow in this research:

1. Flow is incompressible
2. All components are in series with no branches
3. Everything is in the same plane relative to ground (no height change)
4. Flow rate at the inlet is known
5. Flow is turbulent everywhere

Most of these assumptions could be relaxed if more accuracy is desired, with the penalty of increased computational expense and more complex sensitivity analysis.

We begin with equation (2.26) which is derived in detail from an energy balance in [19]. Each term in the equation is put in a form to have units of length. This is the standard form as pressure losses for devices and pipe components are reported in units of pressure head (m). The equation relates head loss H_L to pressure, P , and velocity, V at points 1 and 2 in the flow loop.

$$H_L = \frac{P_1 - P_2}{\rho_w} + \frac{V_1^2 - V_2^2}{2g} \quad (2.26)$$

Where ρ_w is the weight density and g is gravitational acceleration. Rearranging equation (2.26) to find the pressure difference between two points gives:

$$\frac{P_1 - P_2}{\rho_w} = H_L - \frac{V_1^2 - V_2^2}{2g} \quad (2.27)$$

The flow rate in the system is known so velocity at any point can easily be calculated.

$$V_i = \frac{Q}{A_i} \quad (2.28)$$

Here Q is the volumetric flow rate and A_i is the cross sectional area of the pipe at the point of interest. This leaves only the head loss to be determined. Head loss is a proxy that captures energy loss which occurs between points 1 and 2 for reasons other than velocity change. Head loss, in units of length, is the standard for reporting pressure in flow systems, with pump efficiency and system characterization curves reported in terms of *head* [24]. Equations for calculating head loss for many components of a pipe flow system can be found in [19], but here only two will be of interest: first, losses do to friction between the fluid and pipe wall, sometimes called major loss; second, losses due to elbows connecting straight segments of pipe. Each straight segment of pipe and each elbow has a loss coefficient, K , assigned based on the geometry of the segment. Total head loss for n_{pe} pipe elements in series can be calculated by combining the loss coefficients according to the equation:

$$H_L = \frac{\dot{w}^2}{2g\rho_w^2} \sum_{i=1}^{n_{pe}} \frac{K_i}{A_i^2} \quad (2.29)$$

where \dot{w} is the weight flow rate.

The loss coefficient for a straight segment of pipe with length l_i and diameter d_i is

$$K_i^s = f_i \frac{l_i}{d_i}, \quad (2.30)$$

and for an elbow with bend angle α_i and bend radius r_i

$$K_i^e = f_i \alpha_i \frac{r_i}{d_i} + (0.1 + 2.4f_i) \sin\left(\frac{\alpha_i}{2}\right) + \frac{6.6f_i(\sin(\frac{\alpha_i}{2}) + \sqrt{\sin(\frac{\alpha_i}{2}) + \varepsilon_s})}{\left(\frac{r_i}{d_i}\right)^{4\alpha_i/\pi}} - f_i \frac{2l_c}{d_i}. \quad (2.31)$$

See figure 2.3 for a description of the elbow geometry. In equation (2.31) a small perturbation ε_s has been added to make the expression differentiable at $\alpha = 0$. The first term in equation (2.31) accounts for frictional losses in the arc length of the elbow. The final term reduces the loss coefficient to account for the length of straight pipe that is clipped off by the elbow, l_c . Implementing loss coefficient in this way allows each pipe section to be modular. If the length of straight pipe were reduced directly at the straight loss coefficient calculation, information about the connecting pipe and elbow would be needed. The bend angle is found by defining two vectors \mathbf{a} and \mathbf{b} based on the endpoints of two connected segments.

$$\mathbf{a} = \mathbf{x}_{a_f} - \mathbf{x}_{a_0} \quad (2.32)$$

$$\mathbf{b} = \mathbf{x}_{b_f} - \mathbf{x}_{b_0} \quad (2.33)$$

Then from the definition of dot product:

$$\theta = \cos^{-1}(v) \quad (2.34)$$

where

$$v = (1 - \varepsilon_c) \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (2.35)$$

A perturbation ε_c has been incorporated into equation (2.35) to restrict the range so that $v \in [0, (1 - \varepsilon_c)]$. This is done to avoid the undefined derivative of θ when $v = 1$. Angle α in (2.31) is defined as the supplementary angle of θ .

$$\alpha = \pi - \theta \quad (2.36)$$

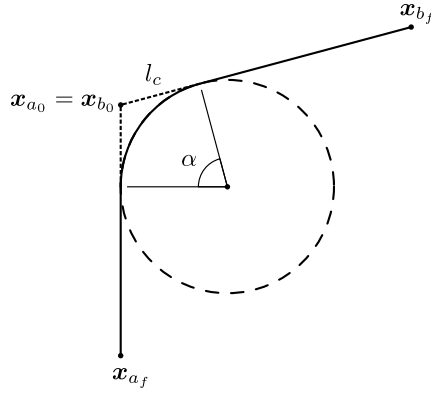


Figure 2.3: Pipe elbow geometry

Clipped length is calculated with the equation

$$l_c = r_i \sqrt{\frac{1+v}{1-v}}. \quad (2.37)$$

The variable f_i is called the *friction factor* and it appears in both loss coefficient equations. Friction factor is a function of Reynold's number, and can also account for roughness of pipe walls. Many equations for friction factor have been developed based on experimental results. In the present work the equation for turbulent flow in smooth pipes proposed by Blasius is used [13].

$$f_i = 0.3164 Re_i^{-0.25}, \quad (2.38)$$

with Reynold's number

$$Re_i = \frac{V_i d_i \rho_m}{\mu}, \quad (2.39)$$

where ρ_m is mass density and μ is viscosity of the fluid. The use of the thermal conductivity and pipe flow models in optimization will be discussed in Chapter 3.

Chapter 3

Optimization problem and sensitivity analysis

This section will formally state the optimization problem, and provide derivations for the sensitivities needed to use gradient based optimization methods. The models presented in Chapter 2 lead to a specific choice of design variables which will be discussed in this section. Furthermore, this set of design variables can be used to define geometric constraints to prevent interference between different components of the system.

In this research the system that is being optimized consists of a number of devices and the routing which connects the devices, for example see figures 4.2 or 4.6. The goal is to find the optimal system layout. System layout will be defined as the placement of the devices and routing in space, as well as some size parameters such as diameter of routing. Each device and element of the routing will be parameterized as generally as possible. This simplifies calculation of sensitivities and also allows programming the problem in an object oriented manner. More complex devices and routing can be modeled without much additional work because of the use of object oriented programming.

In addition to \mathbf{c}_d and \mathbf{b}_i introduced in chapter 2, devices may have ports with location \mathbf{p}_i relative to the reference point. Ports are the locations where routing is connected to the device. As the reference point moves the polygon and ports will move with it. The shape of devices and location of ports will not be optimized, therefore the only design variable for each device is the location of \mathbf{c}_d . It has been useful to allow some devices to be fixed by excluding the reference point for that device from the design variables for the problem. This can be used to specify interaction points with parts of the system that are not part of the optimization. Pairs of ports will be connected by routing as defined by the needs of the system. Each connection will be made up of one or more straight segments of routing. Increasing the number of segments in a connection allows more options in the design space, but will be more computationally expensive. The routing segments have parameters for the start and end points, \mathbf{x}_{i_0} and \mathbf{x}_{i_f} respectively, and width, w_i . These can all change during the optimization so they are all part of the design variables for the problem.

The models presented in chapter 2 are explicitly calculated in terms of the above parameterization, so sensitivity calculation with respect to those parameters is straightforward. Although, a problem arises around the connections between two routing segments, or a routing segment and a device port. Since each element of the system has its own independent parameters there is nothing to prevent the optimizer from breaking the connections. There are two ways to solve this issue. The first method attempts to enforce constraints between connected points. The constraint could be implemented as either a linear equality (3.1) or nonlinear inequality (3.2) constraint.

$$x_i - x_j = 0 \quad (3.1)$$

$$(x_i - x_j)^2 - \varepsilon \leq 0 \quad (3.2)$$

Not all optimization algorithms allow the use of equality constraints, making the nonlinear version more generally useful. If the nonlinear version is used it requires a large number of nonlinear constraints, increasing the solve time. In practice, a small amount of error ε had to be allowed to prevent the optimization from failing due to over constraint, but still had difficulty converging.

The second method to solve the connection problem makes the connections implicit. This method makes use of two design variable vectors termed the *expanded* and *reduced* design variables. Consider a system with n_d devices and n_s routing segments. The expanded design variable vector

$$\mathbf{z}' := [\mathbf{c}_1, \dots, \mathbf{c}_{n_d}, \mathbf{x}_{1_0}, \mathbf{x}_{1_f}, w_1, \dots, \mathbf{x}_{n_{s0}}, \mathbf{x}_{n_{sf}}, w_{n_s}]^T$$

contains all the parameters discussed above for each element in the layout. The reduced design variable vector contains only the reference points of free devices, \mathbf{c}_d^f , locations where routing segments meet, \mathbf{x}_i^f , and width of each connection, w_i . It is assumed that the width of all segments that connect two ports are the same as it is uncommon to change diameters of electrical wiring or piping between two components. The reduced design vector is then

$$\mathbf{z} := [\mathbf{c}_1^f, \dots, \mathbf{c}_{n_{df}}^f, \mathbf{x}_1^f, \dots, \mathbf{x}_{n_{sf}}^f, w_1, \dots, w_{n_c}]^T,$$

where n_{df} is the number of free devices, n_{sf} is the number of points where routing segments meet, and n_c is the number of connections between pairs of ports. With these two vectors set, up a

mapping can be defined from the reduced design variables to the expanded design variables.

$$\mathbf{z}' = \mathbf{M}\mathbf{z} + \mathbf{P} \quad (3.3)$$

The matrix \mathbf{M} , called the *mapping matrix*, contains only ones and zeros. It is straightforward to compute by determining which elements of \mathbf{z} and \mathbf{z}' correspond to each other. The vector \mathbf{P} is used for fixed devices as they are not in the reduced design variable vector, but have a fixed value throughout the optimization. Also, segment endpoints which are connected to device ports are mapped to the device reference point in the reduced design variable vector by including an offset corresponding to the port location \mathbf{p}_i . It should be noted that the mapping only goes in one direction because the mapping matrix is not generally invertible.

By using two design variable vectors, we get the advantage of both easy calculation of sensitivities and an implicit connection between pipes and devices. All calculations of objective functions and constraints, and their sensitivities are done in terms of the expanded design variable vector. Then, the sensitivities are computed in terms of the reduced design variables by using the chain rule.

$$\frac{df}{dz} = \frac{\partial f}{\partial z'} \frac{dz'}{dz} \quad (3.4)$$

$$\frac{dz'}{dz} = \mathbf{M} \quad (3.5)$$

The optimizer uses the reduced design variables as the primary set of variables so connections are enforced without needing additional constraints. To maintain correspondence between reduced and expanded vectors, the mapping (3.3) is applied each time the reduced design variable vector is updated.

Now that the parameterization of the design space has been determined, the optimization problem

can be stated.

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & f \\
 \text{s.t.} \quad & \mathbf{KT} = \mathbf{P} \\
 & g^{phys} \leq 0 \\
 & g^{dd} \leq 0 \\
 & g^{sd} \leq 0 \\
 & g^{ss} \leq 0
 \end{aligned}$$

Here f is the objective function and g are constraints. The function f can be any one of the objectives discussed in section 3.1. The constraints g^{phys} are constraints which depend on solutions to the physics models. The interference constraints g^{dd} , g^{sd} , and g^{ss} prevent interference between two devices, one routing segment and one device, and two routing segments respectively. These constraints are independent of any physics models, so they are all explicit functions in terms of the design variables.

3.1 Objective and physics constraints

This section will formulate objective functions and their derivatives for the optimization problem. Objective functions and physics constraints are discussed here because they both depend on the solution to one of the physics models. Also, the same function could be used either as an objective function or a constraint. The only change would be how it is specified to the optimizer.

The first function will relate to the solution of the lumped parameter flow model. The function is:

$$f_1 = H_L \tag{3.6}$$

The objective is to minimize the head loss in the flow loop as calculated in equation (2.29). When head loss is used as an objective or constraint the radius of each pipe elbow is also a design variable. The elbow radii r_i are added onto the end of expanded and reduced design vectors with a one to

one mapping. The total derivative using the chain rule is

$$\frac{dH_L}{d\mathbf{x}'} = \frac{\partial H_L}{\partial \mathbf{x}'} + \sum_{i=1}^{n_{pe}} \frac{\partial H_L}{\partial K_i} \frac{dK_i}{d\mathbf{x}'}. \quad (3.7)$$

The design variable vector contains device reference point coordinates, bar end coordinates, bar widths, and elbow radii. For each routing segment, the pipe diameter will be equal to the bar width. The only nonzero elements of the explicit derivative $\frac{\partial H_L}{\partial \mathbf{x}'}$ are those corresponding to bar width.

$$\frac{\partial H_L}{\partial d_i} = -\frac{\pi d_i \dot{w}^2}{2g\rho_w^2 A_i^2} \sum_{i=1}^{n_{pe}} K_i \quad (3.8)$$

For each lumped parameter element, whether a straight section or elbow, the following equation applies:

$$\frac{\partial H_L}{\partial K_i} = \frac{1}{A_i^2} \frac{\dot{w}^2}{2g\rho_w^2} \quad (3.9)$$

The final derivative in equation (3.7), $\frac{dK_i}{d\mathbf{x}'}$, depends whether the element is a straight or elbow section. For a straight section design variables are segment end points \mathbf{x}_{i_0} and \mathbf{x}_{i_f} and segment diameter d_i . The sensitives are given below.

$$\frac{dK_i^s}{d\mathbf{x}_{i_0}} = \frac{f_i}{d_i l_i} (\mathbf{x}_{i_0} - \mathbf{x}_{i_f}) \quad (3.10)$$

$$\frac{dK_i^s}{d\mathbf{x}_{i_f}} = \frac{f_i}{d_i l_i} (\mathbf{x}_{i_f} - \mathbf{x}_{i_0}) \quad (3.11)$$

$$\frac{dK_i^s}{dd_i} = -\frac{f_i l_i}{d_i^2} + \frac{l_i}{d_i} \frac{df_i}{dd_i} \quad (3.12)$$

$$\frac{df_i}{dd_i} = 0.0791 Re^{-1.25} \frac{4\dot{m}}{\pi \mu d_i^2} \quad (3.13)$$

If the lumped parameter element is an elbow, the design variables are the four end points of connected segments, \mathbf{x}_{a_0} , \mathbf{x}_{a_f} , \mathbf{x}_{b_0} , and \mathbf{x}_{b_f} , diameter, d_i , and radius of the elbow, r_i . It is assumed that the diameters of connected segments are the same so there is only one diameter variable. The sensitivity of the elbow loss coefficient with respect to pipe diameter is

$$\frac{dK_i^e}{dd_i} = \frac{\partial K_i^e}{\partial d_i} + \frac{\partial K_i^e}{\partial f_i} \frac{df_i}{dd_i} \quad (3.14)$$

Equation (3.13) can be used again here. Partial derivatives required for equation (3.14) are

$$\frac{\partial K_i^e}{\partial d_i} = -f_i \alpha_i \frac{r_i}{d_i^2} + 6.6 \left(\frac{4\alpha_i}{\pi} \right) f_i \left(\sin \left(\frac{\alpha_i}{2} \right) + \sqrt{\sin \left(\frac{\alpha_i}{2} \right) + \varepsilon_s} \right) r^{(-4\alpha_i/\pi)} d^{(4\alpha_i/\pi)-1} + f_i \frac{2l_c}{d_i^2} \quad (3.15)$$

$$\frac{\partial K_i^e}{\partial f_i} = \alpha_i \frac{r_i}{d_i} + 2.4 \sin \left(\frac{\alpha_i}{2} \right) + \frac{6.6 \left(\sin \left(\frac{\alpha_i}{2} \right) + \sqrt{\sin \left(\frac{\alpha_i}{2} \right) + \varepsilon_s} \right)}{\left(\frac{r}{d} \right)^{4\alpha_i/\pi}} - \frac{2l_c}{d_i} \quad (3.16)$$

With respect to elbow radius, the sensitivity is:

$$\frac{dK_i^e}{dr_i} = f_i \alpha_i \frac{1}{d_i} - 6.6 \frac{4\alpha_i}{\pi} f_e \left(\sin \left(\frac{\alpha_i}{2} \right) + \sqrt{\sin \left(\frac{\alpha_i}{2} \right) + \varepsilon_s} \right) r^{(-4\alpha_i/\pi)-1} d^{(4\alpha_i/\pi)} - \frac{2f_i}{d_i} \frac{dl_c}{dr_i} \quad (3.17)$$

The chain rule can be used to calculate sensitivities with respect to the four segment end points.

$$\frac{dK_i^e}{d\mathbf{x}_{a_0}} = \frac{\partial K_i^e}{\partial \alpha_i} \frac{d\alpha_i}{d\mathbf{x}_{a_0}} - \frac{2f_i}{d_i} \frac{dl_c}{d\mathbf{x}_{a_0}} \quad (3.18)$$

with

$$\begin{aligned} \frac{\partial K_i^e}{\partial \alpha_i} &= f_i \frac{r_i}{d_i} + \frac{1}{2} (0.1 + 2.4f_e) \cos \left(\frac{\alpha_i}{2} \right) \\ &\quad + \frac{26.4}{\pi} f_i \left(\sin \left(\frac{\alpha_i}{2} \right) + \sqrt{\sin \left(\frac{\alpha_i}{2} \right) + \varepsilon_s} \right) \left(\frac{d}{r} \right)^{(4\alpha_i/\pi)} \ln \left(\frac{d}{r} \right) \\ &\quad + 6.6 f_i \left(\frac{d}{r} \right)^{(4\alpha_i/\pi)} \left(\frac{1}{4} \frac{1}{\sqrt{\sin \left(\frac{\alpha_i}{2} \right) + \varepsilon_s}} \cos \left(\frac{\alpha_i}{2} \right) + \frac{1}{2} \cos \left(\frac{\alpha_i}{2} \right) \right) \end{aligned} \quad (3.19)$$

The derivatives of angle α are

$$\frac{d\alpha_i}{d\mathbf{x}_{a_0}} = \frac{\partial \alpha_i}{\partial v} \frac{dv}{d\mathbf{x}_{a_0}}, \quad (3.20)$$

with

$$\frac{\partial \alpha_i}{\partial v} = \frac{1}{\sqrt{1-v^2}}. \quad (3.21)$$

The required derivatives of the clipped length are as follows.

$$\frac{dl_c}{dr_i} = \sqrt{\frac{1+v}{1-v}} \quad (3.22)$$

$$\frac{dl_c}{d\mathbf{x}_{a_0}} = \frac{\partial l_c}{\partial v} \frac{dv}{d\mathbf{x}_{a_0}} \quad (3.23)$$

where

$$\frac{\partial t_c}{\partial v} = \frac{r_i}{(1-v)^2} \sqrt{\frac{1-v}{1+v}}. \quad (3.24)$$

The calculations in equations (3.20) and (3.23) require derivatives of v with respect to the four bar end points.

$$\frac{dv}{d\mathbf{x}_{a_f}} = -\frac{dv}{d\mathbf{x}_{a_0}} = (1 - \varepsilon_c) \left[\frac{\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} - \frac{1}{\|\mathbf{a}\|^3\|\mathbf{b}\|} (\mathbf{a} \cdot \mathbf{b}) \mathbf{a} \right] \quad (3.25)$$

$$\frac{dv}{d\mathbf{x}_{b_f}} = -\frac{dv}{d\mathbf{x}_{b_0}} = (1 - \varepsilon_c) \left[\frac{\mathbf{a}}{\|\mathbf{a}\|\|\mathbf{b}\|} - \frac{1}{\|\mathbf{a}\|\|\mathbf{b}\|^3} (\mathbf{a} \cdot \mathbf{b}) \mathbf{b} \right] \quad (3.26)$$

A second objective function is to minimize one of the device temperatures. Temperature at the device reference point will be used, so the objective equation is:

$$f_2 = T(c_d) \quad (3.27)$$

This objective is a function of temperature which is the solution to the thermal finite element problem. The finite element solution (2.8) is a function of the design variables. Explicit derivatives, denoted $\frac{\partial f}{\partial x}$, capture the direct sensitivity of a function with respect to the variable x . Implicit derivatives, denoted $\frac{df}{dx}$, capture indirect sensitivities of the objective through the finite element solution. Implicit derivatives are more difficult to calculate in closed form. The total derivative of a function with respect to a design variable x is

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \mathbf{y}} \frac{d\mathbf{y}}{dx} \quad (3.28)$$

The adjoint method will be used to eliminate the need to calculate implicit derivatives. In order to do this equation (2.8) is rearranged into a residual equation and partitioned into blocks p and f . Blocks denoted by superscript p correspond to *prescribed* temperature degrees of freedom, and blocks denoted by superscript f correspond to *free* (unknown) temperature degrees of freedom.

$$\begin{Bmatrix} \mathbf{R}^p \\ \mathbf{R}^f \end{Bmatrix} = \begin{bmatrix} \mathbf{K}^{pp} & \mathbf{K}^{pf} \\ \mathbf{K}^{fp} & \mathbf{K}^{ff} \end{bmatrix} \begin{Bmatrix} \mathbf{T}^p \\ \mathbf{T}^f \end{Bmatrix} - \begin{Bmatrix} \mathbf{P}^p \\ \mathbf{P}^f \end{Bmatrix} = \mathbf{0} \quad (3.29)$$

A state variable vector will be defined $\mathbf{y} = [\mathbf{P}^p, \mathbf{T}^f]^T$ which are the values solved for in the finite

element problem. The vector \mathbf{P}^p is the flux at nodes where prescribed temperature boundary conditions are applied, and \mathbf{T}^f are temperatures at all remaining nodes.

A new function is defined including the objective function, f , residual, \mathbf{R} , and an adjoint vector, Ψ .

$$\Pi = f + \Psi^T \mathbf{R} \quad (3.30)$$

Since the residual is zero, Π is equal to f and their total derivatives will also be equal. The derivative of Π is

$$\frac{d\Pi}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \mathbf{y}} \frac{d\mathbf{y}}{dx} + \Psi^T \left(\frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{R}}{\partial \mathbf{y}} \frac{d\mathbf{y}}{dx} \right) \quad (3.31)$$

Implicit derivatives $\frac{d\mathbf{y}}{dx}$ are grouped together.

$$\frac{d\Pi}{dx} = \frac{\partial f}{\partial x} + \Psi^T \frac{\partial \mathbf{R}}{\partial x} + \left(\frac{\partial f}{\partial \mathbf{y}} + \Psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{y}} \right) \frac{d\mathbf{y}}{dx} \quad (3.32)$$

Then an adjoint vector is chosen such that the implicit derivative is eliminated. The resulting equation for the adjoint vector is

$$\Psi = - \left[\frac{\partial \mathbf{R}}{\partial \mathbf{y}} \right]^{-T} \left[\frac{\partial f}{\partial \mathbf{y}} \right]^T. \quad (3.33)$$

The remaining terms in equation (3.32) are used to calculate the total derivative.

$$\frac{df}{dx} = \frac{d\Pi}{dx} = \frac{\partial f}{\partial x} + \Psi^T \frac{\partial \mathbf{R}}{\partial x} \quad (3.34)$$

The derivative of the residual with respect to state variable vector results in a matrix which can be constructed from the global stiffness matrix.

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{y}} \right] = \begin{bmatrix} -I & \mathbf{K}^{pf} \\ \mathbf{0} & \mathbf{K}^{ff} \end{bmatrix} \quad (3.35)$$

The temperature at an arbitrary location in the domain needs to be calculated by interpolation from nodal temperatures.

$$T(\mathbf{c}_d) = \mathbf{N}^T(\xi, \eta) \mathbf{T}_{el} \quad (3.36)$$

Where ξ and η are the location of \mathbf{c}_d in the local element coordinate system, and \mathbf{T}_{el} is the vector

of element nodal temperatures. When a structured mesh is used, local element coordinates can easily be calculated. The only nonzero derivative with respect to design variables \mathbf{x}' is for the device reference point. Using the definition of the matrix $\mathbf{B} = \frac{d\mathbf{N}}{d\mathbf{x}}$ explicit derivatives of f_2 can then be calculated.

$$\frac{\partial f_2}{\partial \mathbf{c}_d} = \mathbf{B}^T \mathbf{T}_{el} \quad (3.37)$$

$$\frac{\partial f_2}{\partial \mathbf{y}} = \mathbf{N}(\xi, \eta) \quad (3.38)$$

The derivative of the residual with respect to design variables is calculated using the chain rule.

$$\frac{d\mathbf{R}}{dx'_i} = \frac{\partial \mathbf{R}}{\partial \rho_e} \frac{d\rho_e}{dx'_i} \quad (3.39)$$

Finally, taking into account (2.9) and (2.8) the total derivative is

$$\frac{df_2}{dx'_i} = \frac{\partial f_2}{\partial x'_i} + \sum_{e=1}^{N_e} p(1 - \rho_{min})\rho_e^{(p-1)} \boldsymbol{\Psi}_e^T \mathbf{k}_0 \mathbf{T}_e \frac{d\rho_e}{dx'_i} + \sum_{d=1}^{N_d} Q_d \sum_{e=1}^{N_e} p\rho_{de}^{(p-1)} \boldsymbol{\Psi}_e^T \mathbf{p}_0 \frac{d\rho_{de}}{dx'_i}. \quad (3.40)$$

Where N_e is the number of finite elements in the mesh, ρ_e is the element density from equation (2.23), ρ_{de} is device element density from equation (2.22), and $\boldsymbol{\Psi}_e$ and \mathbf{T}_e are the adjoint and temperature vectors corresponding to element degrees of freedom.

The derivative of the geometric projection is the final part of equation (3.40). The derivative of density resulting from the merge in equation (2.23) is:

$$\frac{d\rho}{dx'_i} = \sum_{q=1}^{N_q} \left(\frac{\rho_q}{\rho} \right)^{p-1} \frac{d\rho_q}{dx'_i} + \sum_{d=1}^{N_d} \left(\frac{\rho_d}{\rho} \right)^{p-1} \frac{d\rho_d}{dx'_i} \quad (3.41)$$

The derivatives of the routing segment projection are calculated with the following equation.

$$\frac{d\rho_q}{dx'_i} = \frac{\partial \rho_q}{\partial \phi_q} \frac{d\phi_q}{dx'_i} \quad (3.42)$$

with

$$\frac{\partial \rho_q}{\partial \phi_q} = \begin{cases} -\frac{2}{\pi r^2} \sqrt{r^2 - d_q^2} & \text{if } -r \leq d_q \leq r \\ 0 & \text{otherwise} \end{cases} \quad (3.43)$$

$$\frac{d\phi_q}{d\mathbf{x}_{q_0}} = \begin{cases} -\frac{\mathbf{b}}{\|\mathbf{b}\|} & \text{if } \mathbf{a} \cdot \mathbf{b} \leq 0 \\ \frac{1}{\|\mathbf{g}\|} \left[\frac{1}{\|\mathbf{a}\|^2} ((\mathbf{a} \otimes \mathbf{b})^T + (\mathbf{a} \cdot \mathbf{b})\mathbf{I}) - \mathbf{I} \right] \mathbf{g} & \text{if } 0 < \mathbf{a} \cdot \mathbf{b} < \mathbf{a} \cdot \mathbf{a} \\ \mathbf{0} & \text{if } \mathbf{a} \cdot \mathbf{b} \geq \mathbf{a} \cdot \mathbf{a} \end{cases} \quad (3.44)$$

$$\frac{d\phi_q}{d\mathbf{x}_{q_f}} = \begin{cases} \mathbf{0} & \text{if } \mathbf{a} \cdot \mathbf{b} \leq 0 \\ -\frac{1}{\|\mathbf{g}\|\|\mathbf{a}\|^2} ((\mathbf{a} \otimes \mathbf{b})^T + (\mathbf{a} \cdot \mathbf{b})\mathbf{I}) \mathbf{g} & \text{if } 0 < \mathbf{a} \cdot \mathbf{b} < \mathbf{a} \cdot \mathbf{a} \\ -\frac{\mathbf{e}}{\|\mathbf{e}\|} & \text{if } \mathbf{a} \cdot \mathbf{b} \geq \mathbf{a} \cdot \mathbf{a} \end{cases} \quad (3.45)$$

$$\frac{d\phi_q}{dw} = -\frac{1}{2}. \quad (3.46)$$

For devices, the design variable is the device reference point location.

$$\frac{d\rho_d}{d\mathbf{c}_d} = \rho_d^{1-p} \sum_{e=1}^{N_e} (\tilde{\rho}_e)^{p-1} \frac{\partial \tilde{\rho}_e}{\partial \mathbf{c}_d}, \quad (3.47)$$

where

$$\frac{\partial \tilde{\rho}_e}{\partial \mathbf{c}_d} = \frac{\partial \tilde{\rho}_e}{\partial \mathbf{x}_{e_0}} \frac{\partial \mathbf{x}_{e_0}}{\partial \mathbf{c}_d} + \frac{\partial \tilde{\rho}_e}{\partial \mathbf{x}_{e_f}} \frac{\partial \mathbf{x}_{e_0}}{\partial \mathbf{c}_d}. \quad (3.48)$$

Derivatives with respect to edge ends appear here as $\frac{\partial \tilde{\rho}_e}{\partial \mathbf{x}_{e_0}}$ and $\frac{\partial \tilde{\rho}_e}{\partial \mathbf{x}_{e_f}}$. These can both be calculated using equation (3.42) because routing and device edges are projected with the same equation. From the definition of edge endpoints in equations (2.20) and (2.21), derivatives $\frac{\partial \mathbf{x}_{e_0}}{\partial \mathbf{c}}$ and $\frac{\partial \mathbf{x}_{e_f}}{\partial \mathbf{c}}$ are the identity matrix. The density derivative of any element inside the device polygon is set to $\frac{\partial \rho_d}{\partial \mathbf{c}} = \mathbf{0}$. To justify making the sensitivity of interior elements zero, imagine perturbing the reference point of the device by a small amount. Most elements inside the polygon are still inside the polygon so there is no change in the density. Few elements near the edges may have switched from being inside the polygon to in the edge bar or vice versa. These elements will be near the bar center and should still have a density of 1.

3.2 Geometric constraints

The interference constraint equations will be presented below along with sensitivity analysis. Any constraint involving a device is not enforced on the device boundary but on a bounding circle with radius r^b centered at the device reference point. The radius can be found by taking the maximum

vertex distance from the device reference point, see figure 3.1a and 3.1b. The constraint between two devices i and j , and its sensitivity are shown below. See figure 3.1a for an illustration of the constraint.

$$g^{dd} = (r_i^b + r_j^b)^2 - \|\mathbf{c}_j - \mathbf{c}_i\|^2 \leq 0 \quad (3.49)$$

$$\frac{dg^{dd}}{d\mathbf{c}_i} = 2(\mathbf{c}_j - \mathbf{c}_i) \quad (3.50)$$

$$\frac{dg^{dd}}{d\mathbf{c}_j} = -2(\mathbf{c}_j - \mathbf{c}_i) \quad (3.51)$$

For the constraint between a device i and a segment of routing j we will use previous results from projecting a bar onto the mesh. As an intermediate step in the projection, the distance between a line segment and a point was found. Here the distance will be found between the line segment and a device reference point rather than a mesh element center. The constraint function is:

$$g^{sd} = \frac{w_j}{2} + r_i^b - d_{ij} \leq 0 \quad (3.52)$$

Sensitivities $\frac{dd_{ij}}{d\mathbf{x}_{j_0}}$ and $\frac{dd_{ij}}{d\mathbf{x}_{j_f}}$ are already known from previous results as well as $\frac{dg^{sd}}{dw_j} = \frac{1}{2}$. The device reference point is a design variable, whereas the element centers were not treated as design variables in previous results, so an additional sensitivity needs to be calculated.

$$\frac{dd_{ij}}{d\mathbf{c}_i} = \begin{cases} \frac{\mathbf{b}}{\|\mathbf{b}\|} & \mathbf{a} \cdot \mathbf{b} \leq 0 \\ \frac{1}{\|\mathbf{g}\|} \left(\mathbf{I} - \frac{1}{\|\mathbf{a}\|^2} (\mathbf{a} \otimes \mathbf{a}) \right) \mathbf{g} & 0 < \mathbf{a} \cdot \mathbf{b} < \mathbf{a} \cdot \mathbf{a} \\ \frac{\mathbf{e}}{\|\mathbf{e}\|} & \mathbf{a} \cdot \mathbf{b} \geq \mathbf{a} \cdot \mathbf{a} \end{cases} \quad (3.53)$$

A constraint to prevent interference between two routing segments requires finding the distance between two line segments. Reference [22] describes an algorithm for calculating the distance between two line segments. First the two segments are extended into infinite lines and the minimum distance is found. If the minimum occurs at a point on the line outside of the segment endpoints, then a series of cases must be tried to find distance between endpoints and the other segment. MATLAB code to compute the minimum distance between two segments, and the derivative with respect to the segment endpoints is included in the appendix. The constraint to avoid interference between

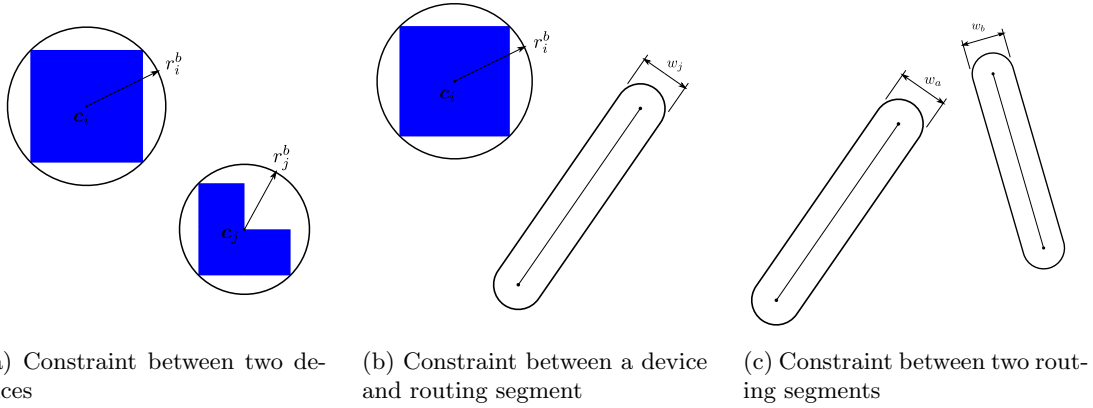


Figure 3.1: Geometric constraints

routing segments a and b with minimum distance between them d_{ab} is presented below.

$$g^{ss} = \left(\frac{w_a}{2} + \frac{w_b}{2} \right)^2 - d_{ab}^2 \leq 0 \quad (3.54)$$

The squared distance is used to avoid undefined derivatives due to a square root operation when the distance is zero. Sensitivity with respect to segment end points can be found in the MATLAB code in the appendix. Sensitivity with respect to the bar widths are:

$$\frac{dg^{ss}}{dw_a} = \frac{dg^{ss}}{dw_b} = \left(\frac{w_a}{2} + \frac{w_b}{2} \right) \quad (3.55)$$

Chapter 4

Results

The above method will be used to optimize the packing and routing of two different systems. The first system consists of three identical devices connected in a loop. A comparison of the results using three different objective functions will be made. The second system optimization will incorporate complexity which would be useful for optimizing an actual system such as fixed points and unique devices. Boundary conditions for the thermal problem will be the same for both systems. The boundary conditions for the thermal problem are a fixed temperature of 100 °C on the right edge, insulated on the left edge, and forced convection on the top and bottom edges, see figure 4.1. The forced convection has an environment temperature of 0 °C and convection coefficient $h = 35.4 \text{ W}/(\text{m}^2\text{K})$. The thermal finite element problem was modeled on a structured mesh of 150x150 four node quadrilateral elements. The MATLAB sequential quadratic programming (sqp) optimization method was used for all optimizations below.

The system depicted in figure 4.2 will be optimized according to three different objective functions. The system has three 0.06 m x 0.06 m square devices. Each device generates 3000 W/m² of heat. Routing connects the devices in a loop. Each connection has four segments, although the segments connected to device ports are fixed relative to the device. This was done so that the constraint

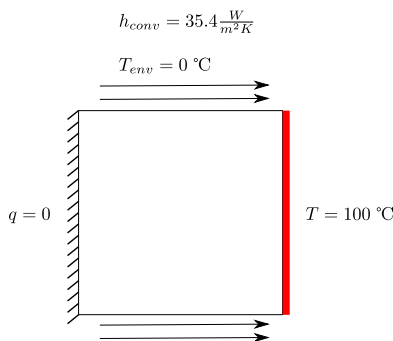


Figure 4.1: Boundary condition for thermal finite element analysis

between devices and routing can be applied to all free routing segments. Each connection between two ports has one free intersection point as a design variable. The thermal conductivity of the solid material is 54 W/(m K) and .032 W/(m K) is used for void. The domain for the optimization is 0.5 m x 0.5 m. A projection radius of 0.00236 m, circumscribing the finite element was used and the penalization parameter was 3. Properties of fluid flow in the routing are listed in table 4.1. Test A, B, and C will minimize head loss, temperature of device 3, and bounding box respectively. The optimization problem statements are below. Test A problem statement is

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & f = H_l \\
 \text{s.t.} \quad & T(\mathbf{c}_1), T(\mathbf{c}_2), T(\mathbf{c}_3) \leq T_c \\
 & g^{geo} \leq 0 \\
 & \mathbf{KT} = \mathbf{P}.
 \end{aligned}$$

Test B problem statement is

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & f = T(\mathbf{c}_3) \\
 \text{s.t.} \quad & T(\mathbf{c}_1), T(\mathbf{c}_2) \leq T_c \\
 & H_l \leq H_{l_c} \\
 & g^{geo} \leq 0 \\
 & \mathbf{KT} = \mathbf{P}.
 \end{aligned}$$

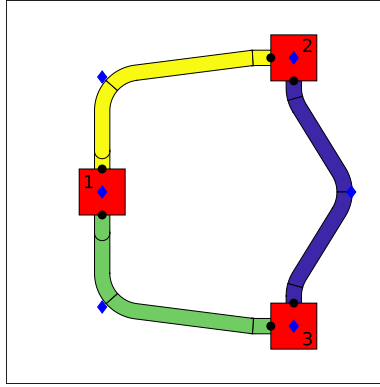


Figure 4.2: Initial system layout for test A, B, and C. Device numbers shown. Blue diamonds are design variable locations.

Test C problem statement is

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & f = (\max(x) - \min(x))(\max(y) - \min(y)) \\
 \text{s.t.} \quad & T(\mathbf{c}_1), T(\mathbf{c}_2), T(\mathbf{c}_3) \leq T_c \\
 & H_l \leq H_{l_c} \\
 & g^{geo} \leq 0 \\
 & \mathbf{KT} = \mathbf{P}.
 \end{aligned}$$

The critical device temperature, T_c , for all three devices is 30 °C. Head loss constraints, H_{l_c} , of 4 m and 5 m will be used to quantify the effect of this constraint on the outcome. Here all geometric constraints from section 3.2 have been lumped together into g^{geo} . The objective function of test C is the area of a bounding box aligned to the x and y axis containing all devices and routing.

The resulting layout from test A is shown in figure 4.3. Head loss in this layout is 2.063 m. In order to satisfy device temperature constraints one of the routing connections touches the convection boundary. This conducts heat from the devices through the routing to the boundary where it can be dissipated. The optimization tries to balance creating smooth bends with reducing the length of pipe to minimize head loss. Table 4.2 contains the objective values for test B with two different head loss constraint values. The optimal layouts are shown in figure 4.4. Both layouts have routing touching the top boundary to dissipate heat from devices. The layout for the 4 m head loss constraint has

	symbol	value
density	ρ_m	1072 kg/m ³
viscosity	μ	0.068 kg/(m s)
flow rate	Q	0.001 m ³ /s
max pipe diameter	-	0.02 m

Table 4.1: Fluid flow properties

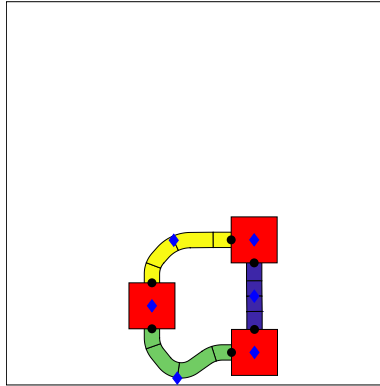


Figure 4.3: Optimal layout of test A

smoother elbows which results in a 80.4 % increase in the temperature of the third device. Results of test C are in table 4.3 and final layouts are in figure 4.5. The layout for the 4 m head loss objective again requires smoother elbows and results in a 16.6 % increase in the bounding box area. Each objective function produces significantly different optimal layouts. It is up to the designer of the system to choose which is most important for that system.

A theoretical power electronics cooling system was optimized using this method. The initial system layout is depicted in figure 4.6 and properties of the devices are given in table 4.4. There are two battery packs and an AC/DC converter which add heat to the thermal problem, and a heat exchanger which removes heat. Connections to the outside are fixed on the right boundary of the domain. Boundary conditions and flow properties are same as the first problem, except the domain has been enlarged to 1 m x 1 m and maximum pipe diameter is 0.03 m. There are two free points in each connection to allow more complex routing. The optimization was run with head loss and

Head Loss (m)	objective (°C)	% increase
≤ 5	10.16	-
≤ 4	18.33	80.4

Table 4.2: Results of test B

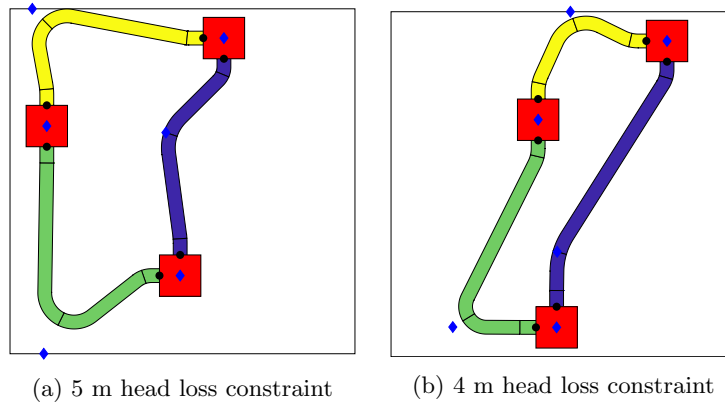


Figure 4.4: Final layouts for test B.

Head Loss (m)	objective (m ²)	% increase
≤ 5	0.0156	-
≤ 4	0.0182	16.6

Table 4.3: Results of test C

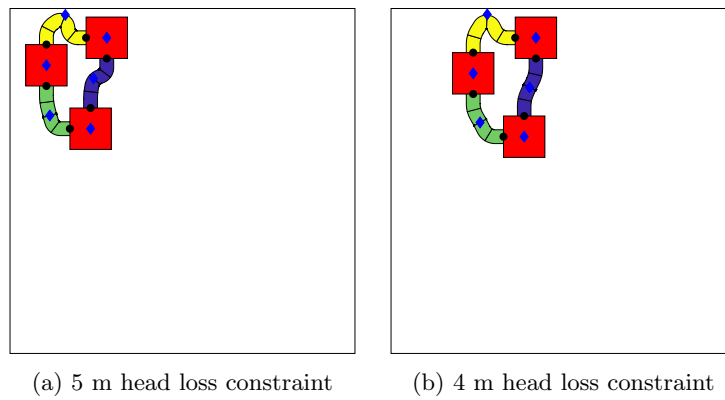


Figure 4.5: Final layouts for test C.

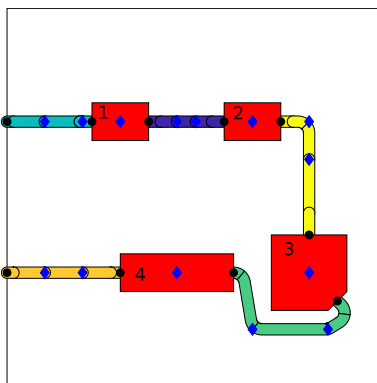


Figure 4.6: Initial layout of power electronics cooling system

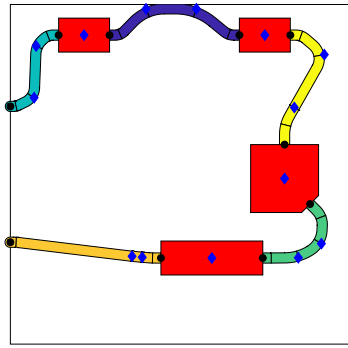
Device number	Description	Q_d (W/m ²)	T_{max} (°C)
1	Battery	5000	30
2	Battery	5000	30
3	AC/DC converter	1000	70
4	Heat exchanger	-2000	-

Table 4.4: Device properties

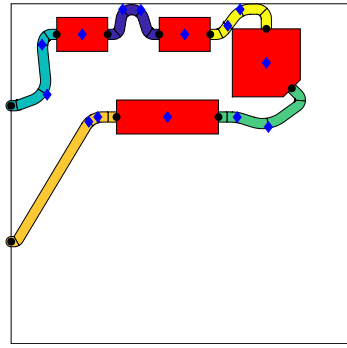
bounding box objective functions. The bounding box optimization had a head loss constraint of 1.5 m in addition to device temperature constraints. Resulting layouts are shown in figure 4.7 and some values from the final layouts are in table 4.5. In the head loss optimization, device 1 and 2 temperature constraints were active, and in the bounding box optimization, device 1 temperature and head loss constraints were active. As expected the layout for the bounding box objective has sharper turns at the elbows with several close to 90 degrees. The head loss objective layout has a higher total piping length but lower head loss. This suggests that elbow geometry is the dominant contributor to head loss.

objective	H_l (m)	bounding box (m ²)	T_1 (°C)	T_2 (°C)	T_3 (°C)
head loss	0.876	0.711	30.0	30.0	27.3
bounding box	1.50	0.311	30.0	22.9	17.2

Table 4.5: Power electronics cooling system optimization results. Objective value highlighted in gray.



(a) Optimal layout for pressure objective



(b) Optimal layout for bounding box objective

Figure 4.7: Optimal layouts of power electronics cooling system

Chapter 5

Conclusion

A method of simultaneously optimizing the placement of devices and the routing which connects them was presented. The optimization problem takes into account physics-based and geometric objective functions and constraints. Finite element and lumped parameter physics models can both be used in the same optimization problem to model different types of physics.

The geometric projection method of Norato *et al* was extended to project objects of arbitrary polygonal shapes rather than just bars. A set of design variables was developed which is applicable to the projection method, lumped parameter model, and geometric constraints. By using a mapping between expanded and reduced design variables sensitivities can be calculated easily, and connections between pipe segments are enforced without needing additional constraints.

The method was used to optimize a simple 3 device system according to different objective functions. Each objective function resulted in a significantly different optimal layout. This highlights the need for the system designer to understand which aspects of the system are important. Another system with fixed interaction points, complex device shapes, and more routing segments was optimized. These features are all useful in designing a real device-routing system. The method presented here could lead to faster development of systems, which are smaller and perform better than those designed by conventional design methods.

References

- [1] M. P. Bendsøe, A. Ben-Tal, and J. Zowe. “Optimization methods for truss geometry and topology design”. In: *Structural optimization* 7.3 (Apr. 1994), pp. 141–159. ISSN: 1615-1488. DOI: 10.1007/BF01742459. URL: <https://doi.org/10.1007/BF01742459>.
- [2] Jaco Dirker and Josua P. Meyer. “Topology optimization for an internal heat-conduction cooling scheme in a square domain for high heat flux applications”. English. In: *Journal of Heat Transfer* 135.11 (2013). ISSN: 00221481. URL: <http://dx.doi.org/10.1115/1.4024615>.
- [3] P.D. Dunning, B.K. Stanford, and H.A. Kim. “Coupled aerostructural topology optimization using a level set method for 3D aircraft wings”. English. In: *Struct. Multidiscip. Optim. (Germany)* 51.5 (2015), pp. 1113–32. ISSN: 1615-147X. URL: <http://dx.doi.org/10.1007/s00158-014-1200-1>.
- [4] K. Hormann and A. Agathos. “The point in polygon problem for arbitrary polygons”. English. In: *Comput. Geom., Theory Appl. (Netherlands)* 20.3 (2001), pp. 131–44. ISSN: 0925-7721. URL: [http://dx.doi.org/10.1016/S0925-7721\(01\)00012-8](http://dx.doi.org/10.1016/S0925-7721(01)00012-8).
- [5] Thomas J. R. Hughes. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. English. Mineola, NY: Dover Publications, 2000. ISBN: 978-0-486-41181-1.
- [6] A. Iga et al. “Topology optimization for thermal conductors considering design-dependent effects, including heat conduction and convection”. English. In: *International Journal of Heat and Mass Transfer* 52.11-12 (2009), pp. 2721–2732. ISSN: 00179310. URL: <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2008.12.013>.
- [7] K.A. James, G.J. Kennedy, and J.R.R.A. Martins. “Concurrent aerostructural topology optimization of a wing box”. English. In: *Comput. Struct. (UK)* 134 (2014), pp. 1–17. ISSN: 0045-7949. URL: <http://dx.doi.org/10.1016/j.compstruc.2013.12.007>.

- [8] A. Klarbring et al. “Topology optimization of flow networks”. English. In: *Comput. Methods Appl. Mech. Eng. (Netherlands)* 192.35-36 (2003), pp. 3909–32. ISSN: 0045-7825. URL: [http://dx.doi.org/10.1016/S0045-7825\(03\)00393-1](http://dx.doi.org/10.1016/S0045-7825(03)00393-1).
- [9] N. de Kruijf et al. “Topological design of structures and composite materials with multiobjectives”. English. In: *Int. J. Solids Struct. (UK)* 44.22-23 (2007), pp. 7092–109. ISSN: 0020-7683. URL: <http://dx.doi.org/10.1016/j.ijsolstr.2007.03.028>.
- [10] Chunjiang Liu. “Optimal design of high-rise building wiring based on ant colony optimization”. English. In: *Cluster Computing* (2018), pp. 1–8. ISSN: 13867857. URL: <http://dx.doi.org/10.1007/s10586-018-2195-y>.
- [11] Qiang Liu and Guoshuai Jiao. “A Pipe Routing Method Considering Vibration for Aero-Engine Using Kriging Model and NSGA-II”. English. In: *IEEE Access (USA)* 6 (2018), pp. 6286–92. ISSN: 2169-3536. URL: <http://dx.doi.org/10.1109/ACCESS.2018.2789361>.
- [12] H. P. Mlejnek. “Some aspects of the genesis of structures”. In: *Structural optimization* 5.1 (Mar. 1992), pp. 64–69. ISSN: 1615-1488. DOI: 10.1007/BF01744697. URL: <https://doi.org/10.1007/BF01744697>.
- [13] Yasuki Nakayama. “Chapter 7 - Flow in Pipes”. In: *Introduction to Fluid Mechanics (Second Edition)*. Ed. by Yasuki Nakayama. Second Edition. Butterworth-Heinemann, 2018, pp. 135–161. ISBN: 978-0-08-102437-9. DOI: <https://doi.org/10.1016/B978-0-08-102437-9.00007-3>. URL: <http://www.sciencedirect.com/science/article/pii/B9780081024379000073>.
- [14] J.A. Norato, B.K. Bell, and D.A. Tortorelli. “A geometry projection method for continuum-based topology optimization with discrete elements”. English. In: *Comput. Methods Appl. Mech. Eng. (Netherlands)* 293 (2015), pp. 306–27. ISSN: 0045-7825. URL: <http://dx.doi.org/10.1016/j.cma.2015.05.005>.
- [15] E. Oktay, H.U. Akay, and O. Merttopcuoglu. “Parallelized structural topology optimization and CFD coupling for design of aircraft wing structures”. English. In: *Comput. Fluids (UK)* 49.1 (2011), pp. 141–5. ISSN: 0045-7930. URL: <http://dx.doi.org/10.1016/j.compfluid.2011.05.005>.
- [16] Jin-Hyung Park and R.L. Storch. “Pipe-routing algorithm development: case study of a ship engine room design”. English. In: *Expert Syst. Appl. (UK)* 23.3 (2002), pp. 299–309. ISSN: 0957-4174. URL: [http://dx.doi.org/10.1016/S0957-4174\(02\)00049-0](http://dx.doi.org/10.1016/S0957-4174(02)00049-0).

- [17] YanFeng Qu et al. “Pipe Routing Approach for Aircraft Engines Based on Ant Colony Optimization”. English. In: *J. Aerosp. Eng. (USA)* 29.3 (2016), p. 04015057. ISSN: 0893-1321. URL: [http://dx.doi.org/10.1061/\(ASCE\)AS.1943-5525.0000543](http://dx.doi.org/10.1061/(ASCE)AS.1943-5525.0000543).
- [18] Tao Ren et al. “A new pipe routing method for aero-engines based on genetic algorithm”. English. In: *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng. (UK)* 228.3 (2014), pp. 424–34. ISSN: 0954-4100. URL: <http://dx.doi.org/10.1177/0954410012474134>.
- [19] Donald C. Rennels and Hobart M. Hudson. *Pipe Flow: A Practical and Comprehensive Guide*. English. 2012. URL: <http://dx.doi.org/10.1002/9781118275276>.
- [20] G. I. N. Rozvany, M. Zhou, and T. Birker. “Generalized shape optimization without homogenization”. In: *Structural optimization* 4.3 (Sept. 1992), pp. 250–252. ISSN: 1615-1488. DOI: 10.1007/BF01742754. URL: <https://doi.org/10.1007/BF01742754>.
- [21] O. Sigmund. “A 99 line topology optimization code written in Matlab”. In: *Structural and Multidisciplinary Optimization* 21.2 (Apr. 2001), pp. 120–127. ISSN: 1615-1488. DOI: 10.1007/s001580050176. URL: <https://doi.org/10.1007/s001580050176>.
- [22] Dan Sunday. *Distance between 3D Lines & Segments*. English. 2012. URL: http://geomalgorithms.com/a07-_distance.html.
- [23] A. Takezawa et al. “Structural topology optimization with strength and heat conduction constraints”. English. In: *Comput. Methods Appl. Mech. Eng. (Netherlands)* 276 (2014), pp. 341–61. ISSN: 0045-7825. URL: <http://dx.doi.org/10.1016/j.cma.2014.04.003>.
- [24] Anthony J. Tarquin and Jack Dowdy. “Optimal Pump Operation in Water Distribution”. In: *Journal of Hydraulic Engineering* 115.2 (1989), pp. 158–168. DOI: 10.1061/(ASCE)0733-9429(1989)115:2(158). URL: <https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9429%281989%29115%3A2%28158%29>.
- [25] Ghanshyam G. Tejani et al. “Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics”. In: *Journal of Computational Design and Engineering* 5.2 (2018), pp. 198–214. ISSN: 2288-4300. DOI: <https://doi.org/10.1016/j.jcde.2017.10.001>. URL: <http://www.sciencedirect.com/science/article/pii/S2288430017300763>.

- [26] Christian Van der Velden et al. “An intelligent system for automatic layout routing in aerospace design”. English. In: *Innovations in Systems and Software Engineering* 3.2 (2007), pp. 117–128. ISSN: 16145046. URL: <http://dx.doi.org/10.1007/s11334-007-0021-4>.
- [27] Lei Wu et al. “Improved ant colony optimization algorithm and its application to solve pipe routing design”. English. In: *Assem. Autom. (UK)* 39.1 (2019), pp. 45–57. ISSN: 0144-5154. URL: <http://dx.doi.org/10.1108/AA-02-2018-022>.
- [28] Tomas Zegard and Glaucio H. Paulino. “Bridging topology optimization and additive manufacturing”. English. In: *Structural and Multidisciplinary Optimization* 53.1 (2016), pp. 175–192. ISSN: 1615147X. URL: <http://dx.doi.org/10.1007/s00158-015-1274-4>.
- [29] Shanglong Zhang et al. “A geometry projection method for the topology optimization of plate structures”. English. In: *Structural and Multidisciplinary Optimization* 54.5 (2016), pp. 1173–1190. ISSN: 1615147X. URL: <http://dx.doi.org/10.1007/s00158-016-1466-6>.
- [30] Xiaojia Shelly Zhang, Glaucio H. Paulino, and Adeildo S. Ramos. “Multi-material topology optimization with multiple volume constraints: a general approach applied to ground structures with material nonlinearity”. English. In: *Structural and Multidisciplinary Optimization* 57.1 (2018), pp. 161–182. ISSN: 1615147X. URL: <http://dx.doi.org/10.1007/s00158-017-1768-3>.
- [31] Ji-Hong Zhu et al. “Integrated layout and topology optimization design of multi-frame and multi-component fuselage structure systems”. English. In: *Structural and Multidisciplinary Optimization* 56.1 (2017), pp. 21–45. ISSN: 1615147X. URL: <http://dx.doi.org/10.1007/s00158-016-1645-5>.
- [32] Jihong Zhu et al. “Simultaneous design of components layout and supporting structures using coupled shape and topology optimization technique”. English. In: *Struct. Multidiscip. Optim. (Germany)* 36.1 (2008), pp. 29–41. ISSN: 1615-147X. URL: <http://dx.doi.org/10.1007/s00158-007-0155-x>.

Appendix A

Distance between segments MATLAB code

```
function [sqDist, dd_dx] = sqDistance(xa0, xaf, xb0, xbf)
    %segmentDistance calculates minimum squared distance between
    % two line segments
    % xa0, xaf - end points of first segment - row vector
    % xb0, xbf - end points of second segment - row vector
    % returns dd_dx in order [xa0, xaf, xb0, xbf]

    dim = size(xa0,2);
    I = eye(dim,dim);
    z = zeros(dim,dim);

    u = xaf - xa0;
    dudx = [-I I z z];
    v = xbf - xb0;
    dvdx = [z z -I I];
    w = xa0 - xb0;
    dwdx = [I z -I z];
    a = u*u';
    b = u*v';
    c = v*v';
    d = u*w';
    e = v*w';
    dadx = 2*u*dudx;
    dbdx = v*dudx + u*dvdx;
    dcdx = 2*v*dvdx;
    dddx = w*dudx + u*dwdx;
    dedx = w*dvdx + v*dwdx;

    D = a*c-b^2;
    dDdx = a*dcdx + c*dadx - 2*b*dbdx;

    sD = D;
    tD = D;
    dsddx = dDdx;
    dtddx = dDdx;

    eps = 1e-5; % tolerance for nearly parallel
    if D < eps
```



```

% segments are nearly parallel
sN = 0;
sD = 1;
tN = e;
tD = c;
dsndx = zeros(1,4*dim);
dsddx = zeros(1,4*dim);
dtndx = dedx;
dtddx = dcdx;
else
sN = (b*e-c*d);
tN = (a*e-b*d);
dsndx = e*dbdx + b*dbdx - d*dcdx - c*dddx;
dtndx = a*dedx + e*dadx - d*dbdx - b*dddx;
if sN < 0
sN = 0;
tN = e;
tD = c;
dsndx = zeros(1,4*dim);
dtndx = dedx;
dtddx = dcdx;
elseif sN > sD
sN = sD;
tN = e+b;
tD = c;
dsndx = dsddx;
dtndx = dedx + dbdx;
dtddx = dcdx;
end
end

if tN < 0
tN = 0;
dtndx = zeros(1,4*dim);
if -d < 0
sN = 0;
dsndx = zeros(1,4*dim);
elseif -d > a
sN = sD;
dsndx = dsddx;
else
sN = -d;
sD = a;
dsndx = -dddx;
dsddx = dadx;
end
elseif tN > tD
tN = tD;
dtndx = dtddx;
if (-d+b) < 0
sN = 0;

```

```

    dsndx = zeros(1,4*dim);
elseif (-d+b) > a
    sN = sD;
    dsndx = dsddx;
else
    sN = -d + b;
    sD = a;
    dsndx = -ddd + dbdx;
    dsddx = dadx;
end
end

if abs(sN) < eps
    sC = 0;
    dsndx = zeros(1,4*dim);
else
    sC = sN/sD;
    dsndx = (1/sD)*dsndx - (sN/(sD^2))*dsddx;
end

if abs(tN) < eps
    tC = 0;
    dtndx = zeros(1,4*dim);
else
    tC = tN/tD;
    dtndx = (1/tD)*dtndx - (tN/(tD^2))*dtddx;
end

dP = w+(sC*u)-(tC*v);
dPdw = I;
dPdu = sC*I;
dPdv = -tC*I;
dPdsc = u';
dPdte = -v';
ddP_dx = [dPdw dPdu dPdv dPdsc dPdte]*[dwdx; dudx; dvdx; dsndx; dtndx];
sqDist = dP*dP';
dd_dx = 2*dP*ddP_dx;
end

```