Optimal Magnetic Shield Design with Second–Order Cone Programming

Takashi Sasakawa[*]
Takashi Tsuchiya[†]

October, 2000
(Final Revision: September 2002)

Abstract

In this paper, we consider a continuous version of the convex network flow problem which involves the integral of the Euclidean norm of the flow and its square in the objective function. A discretized version of this problem can be cast as a second-order cone program, for which efficient primal-dual interior-point algorithms have been developed recently. An optimal magnetic shielding design problem of the MAGLEV train, a new bullet train under development in Japan, is formulated as the continuous convex network flow problem, and is solved with the primal-dual interior-point algorithm. Taking advantage of its efficiency and stability, the algorithm is further applied to robust design of the magnetic shielding.

1 Introduction

Let us consider the following optimization problem:

$$\begin{aligned} &\text{minimize } \int_{\Theta} \left\{ \|a_0(x)v(x)\|^2 + \|a_1(x)v(x)\| + a_2(x)^T v(x) \right\} dx \\ &\text{subject to } \quad \text{div } v(x) = b(x), \quad \|v(x)\| \le c(x), \ x \in \Theta \subset R^2, \end{aligned} \tag{1}$$

where a_0, a_1, b, c are continuous scalar functions and a_2 is a two-dimensional continuous vector function defined over a compact region Θ. v is a continuously differentiable vector field defined over Θ with respect to which the objective function is minimized. We assume appropriate boundary conditions. In the context of applications, v may be regarded as a flow of physical quantities over the region Θ. The constraint div $v = b$ represents the conservation law and $\|v\| \le c$ represents a constraint on the maximum capacity at each point. The objective function typically represents energy or cost of the flow. This problem

[*] Railways Technical Research Institute, Kokubunji, Tokyo 185-8540 Japan (sasakawa@rtri.or.jp)

[†] The Institute of Statistical Mathematics, 4–6–7 Minami-Azabu, Minato-ku, Tokyo 106-8569 Japan (tsuchiya@sun312.ism.ac.jp)

is a continuous version of convex network flow problems and have a number of possible applications [28]. A discretized version of (1) becomes an example of convex optimization problems called second-order cone programs, to which efficient polynomial interior-point algorithms have been developed recently.

In this paper, a real-world optimal magnetic shielding design problem which arises in the development of a new bullet train in Japan [22] is formulated as a special case of (1), and will be solved with the help of polynomial-time primal-dual interior-point algorithms for second-order cone programming. The algorithm is further applied to robust optimization of the magnetic shielding. Through this real-world example, we demonstrate how this type of optimization problems can be reasonably treated by combination of finite element methods and interior-point algorithms. For example, such problems as optimal design of a draining system protecting an area from floods, estimation of a flow from observed data etc. are also formulated as (1) or its slight variation, and can be solved with the techniques developed here. Contact problems with tangential frictional force in mechanics can be cast as similar problems with different linear equality constraints [15].

The bullet train, which is called MAGLEV (superconducting MAGnetically LEVitated vehicle), is held in the air by strong magnetic fields and propelled by linear synchronous motors. Each car is equipped with several super-conducting magnet units which generate the magnetic field. Passengers inside the car need to be shielded from the magnetic field outside. The optimal design problem of the magnetic shielding is to minimize the weight of the shielding by adjusting the thickness of the shielding taking into account the external magnetic field. Intuitively, the shield needs to be thick at a point where the field is strong while it can be thinner at a point where the field is weaker.

After some appropriate simplification, this optimization problem is formulated as a convex program of minimizing the sum of Euclidean norm under linear equality constraints. Plausibility of the model has been confirmed through previous research by one of the authors [23], where the problem was solved with a conventional iterative method and the result was compared with physical experiments. While the computational results seemed plausible, there was no guaranteed bound on the optimal value.

In this paper, we cast this problem into a second-order cone program which is a special case of linear programs over symmetric cones, and solve it with the primal-dual path-following interior-point algorithms developed recently. With the new approach, we can solve the problem in a much more efficient and stable way, providing a nice lower bound on the optimal weight automatically. Furthermore, we are able to handle problems with linear inequality constraints. This allows more flexibility to design, and is crucial in some situations. For example, minimum thickness constraints can be required on some part of the shielding to keep enough strength of the body.

The interior-point algorithm solves optimization problems by tracing vector fields defined in the interior of the feasible region. The algorithm enjoys polynomial-time convergence and has been studied extensively as one of the central topics in the field of optimization since 1984 when Karmarkar proposed the projective scaling method for linear programming (LP). Primal-dual algorithms, which generate sequences in the space of both primal and dual problems, turned out to be the method of choice for LP (see, e.g., [32]). In 1990's, primal-dual algorithms were extended to an important class of convex programming called “linear programming over symmetric cones.” In particular, semidefinite programming (SDP), linear function optimization over the intersection of an affine space and the cone of positive semidefinite matrices, was studied extensively from the viewpoint of algorithms, theories, and applications [1, 20, 31].

The second-order cone programming (SOCP) is another example of linear programming over symmetric cones which optimizes a linear objective function over intersection of an affine space and a direct product of second-order cones. Recently, the polynomial primal-dual path-following algorithms for LP and SDP are extended to SOCP by Nesterov and Todd and one of the authors [21, 30] (See also [19, 26]). We apply these algorithms to the optimal magnetic shield design problem of the MAGLEV train. The number of design variables y in the problems we solved ranges from about 1600 to 60000, and the number of primal variables x (= the number of dual variables s) is $4800 \sim 180000$. The result is considered from the physical point of view, and the performance of the primal-dual algorithm is compared with the previous algorithm. Then the primal-dual algorithm is applied to robust optimization of the magnetic shielding [4, 5, 7, 8, 12]. A robust design is obtained by solving perturbed optimization problems 10000 times in several hours.

This paper is organized as follows. In Section 2, we introduce the optimal magnetic shielding design problem and formulate the problem as a second-order cone program. In Section 3, we introduce SOCP and explain the primal-dual interior-point algorithms. In Section 4, we solve the problem and show the optimized design of shielding. We analyze the results from an engineering point of view. In Section 5, the primal-dual algorithm is applied to robust design of the magnetic shielding. Section 6 is conclusion.

2 Static Magnetic Shield Design Problem and MAGLEV Train

Optimal design of static magnetic shielding arises naturally in several areas including magnetically levitated (MAGLEV) train design, MRI (Magnetic Resonance Image) and low magnetic field measurement etc. In this paper, we focus on the problem which arises in the development of the MAGLEV train [22]. We introduce a simplified formulation of the optimal magnetic shielding design problem and cast this problem into a second-order cone program through discretization by the Finite Element Method.

2.1 Formulation of a Simplified Optimal Magnetic Shielding Design Problem and Second-Order Cone Programming

In MAGLEV train, each car is equipped with two SuperConducting Magnet units (SCM units) which produce magnetic field. Each SCM unit consists of four SuperConducting Coils (SCCs) placed sequentially on each side of the corridor, as is shown in Figure 1. Thus, a total of eight SCCs are laid along the corridor connecting two coaches. Each SCC has racetrack shape with the length 1.07m and the height 0.5m, and it is energized at 700k Ampere. The pair of SCCs facing each other on the both sides are magnetized in the same direction. The directions of magnetization of the four sequential SCCs on the SCM unit are S-N-S-N along the direction of travel.

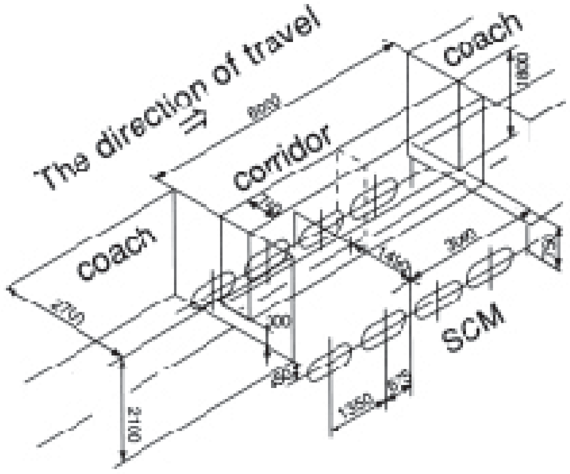


Figure 1: MAGLEV train and configuration of SCCs (unit: mm).

Our purpose is to design a magnetic shielding with the minimum weight which has enough thickness at each point to shield passengers inside from the magnetic field generated by the SCM units. In the previous paper [23], one of the authors proposed a simplified design problem for magnetic shielding which minimizes the required magnetic materials for shielding. We explain it briefly in this subsection.

For magnetic shielding, we enclose the region Ω (interior region) where we want to shield magnetic field by ferromagnetic material, which typically is iron or an iron-based alloy. Our purpose is to reduce the weight of this ferromagnetic material. Magnetic field generated by the current sources which are placed in $R^3 - \Omega$ (exterior region) is bypassed through this ferromagnetic material and does not leak into the interior region in principle. We show this situation in Figure 2. We make the following assumptions regarding this magnetic shielding problem.

[A1] Interior region Ω is completely enclosed by ferromagnetic material, i.e., Ω is enclosed by ferromagnetic material which is placed on all points of $\partial\Omega$ (surface of Ω) and has a sheet-like shape.

[A2] Ferromagnetic material has *infinite* initial permeability.

[A3] Ferromagnetic material has saturated magnetic flux density B_s (unit: Tesla = Wb/m^2).

The assumed *B*-*H* curve of ferromagnetic material is shown in Figure 3. In the above assumptions [A2] and [A3], we take into account only saturated magnetic flux density of

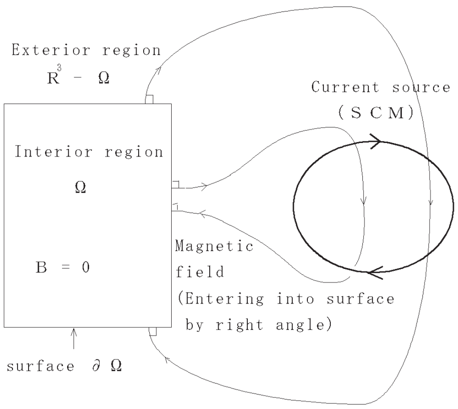


Figure 2: Situation of the problem

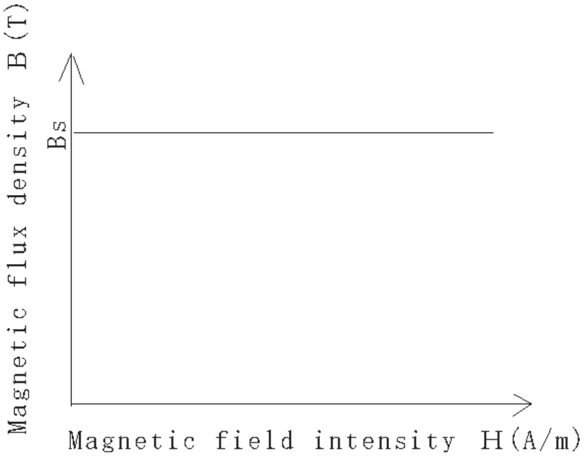


Figure 3: Assumed *B*-*H* curve

shielding material and ignore the detailed features of its B-H curve (in general, nonlinear features). With this simplification, we obtain a convex optimization problem, which is easier to solve but does not lose the essence of the original shielding problem. We explain it in the following.

Due to the nature of the shielding material, the external field generated by the SCM units would not be influenced by changing the thickness of the shielding as long as magnetic saturation does not occur in the shielding. This enables us to solve the problem of determining the external magnetic field separately. Under the assumptions [A1], [A2] and also the prescribed current distribution in $R^3 - \Omega$, we can solve Maxwell equations for the static magnetic field in the exterior region ($R^3 - \Omega$) with the help of boundary conditions on $\partial\Omega$ (tangential component of $\vec{B}$ s.t. $\vec{B} \times \vec{n} = \mathbf{0}$, where $\vec{n}$ is a unit normal on $\partial\Omega$). After computation, we obtain the normal component of magnetic flux density B_n on $\partial\Omega$ (unit: Tesla) which flows into the shield through $\partial\Omega$ at right angles. We can use any analytic or numerical method to solve this boundary value problem (we call it an exterior field problem) and the obtained value of B_n is used later.

Due to the assumption [A2] of infinite permeability of initial B-H curve for shielding material, magnetic flux does not leak into the interior region Ω enclosed by the shielding material (surface). We also assume that magnetic flux density $\vec{B}$ inside the shield is uniform with respect to thickness. Taking this assumption into account, now we introduce a two-dimensional vector field $\vec{F} = (F_1, F_2)$ (unit: Wb/m) as the integral of $\vec{B}$ with respect to thickness (the unit of thickness is meter). Then we have the following equation about the balance of magnetic flux "on" $\partial\Omega$:

$$\mathrm{div}\vec{F} \;=\; B_n \text{ on } \partial\Omega. \tag{2}$$

The two-dimensional divergence operator div in (2) is also taken on $\partial\Omega$. So if $\partial\Omega$ is a plane (at least locally) and coordinates x_1 and x_2 are Cartesian, we have

$$\mathrm{div}\vec{F} = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2}.$$

Secondly, from the assumption [A3], ferromagnetic material has a saturated magnetic flux density B_s and we must use ferromagnetic material at a magnetic flux density of less than B_s. Otherwise ferromagnetic material would be "saturated" and the magnetic field would leak into the interior region Ω. Therefore, we need at least

$$\frac{\| \vec{F} \|}{B_s} \text{ meter}$$

as the thickness at every point on $\partial\Omega$. We use $B_s = 1.5$ Tesla throughout the paper. This value reflects the physical nature of the shielding material which typically is iron or iron-based alloy as was mentioned above, and was used in [23]. Since the total weight of the shield material is proportional to its volume and the volume is given by integrating the thickness over the surface, the problem of minimizing the total weight of the shield material is formulated as the following optimization problem, which is a continuous version of minimizing the sum of Euclidean norm problem:

$$\begin{aligned} \text{minimize } & \frac{1}{B_s}\iint_{\partial\Omega} \| \vec{F} \| \ dS, \\ \text{subject to } \mathrm{div}\vec{F} \ &= \ B_n \text{ on } \partial\Omega, \end{aligned}$$

where the unknown variable $\vec{F}$ is defined on $\partial\Omega$ and B_n is given on $\partial\Omega$. This way, the optimal magnetic shielding design problem can be formulated as a convex network flow problem introduced in (1).

Introducing the auxiliary variable F_0, the problem is rewritten as follows:

$$\text{minimize } \frac{1}{B_s}\iint_{\partial\Omega} F_0 \ dS, \tag{3a}$$

$$\text{subject to } \mathrm{div}\vec{F} \ = \ B_n \text{ on } \partial\Omega, \tag{3b}$$

$$F_0 \geq \| \vec{F} \| \ \text{ on } \partial\Omega. \tag{3c}$$

This is the formulation that we adopt to solve the magnetic shielding design problem. In this formulation, the objective function (3a) and the constraint (3b) are linear with respect to F_0 and $\vec{F}$, respectively. The second constraint (3c) is not linear but is convex with respect to F_0 and $\vec{F}$. The constraint of this type is called the second-order cone constraint and plays a main role in this paper. Thus the magnetic shielding design problem is formulated as a convex optimization problem (3). This problem is a continuous version of second-order cone programs which will be formally introduced in the next section.

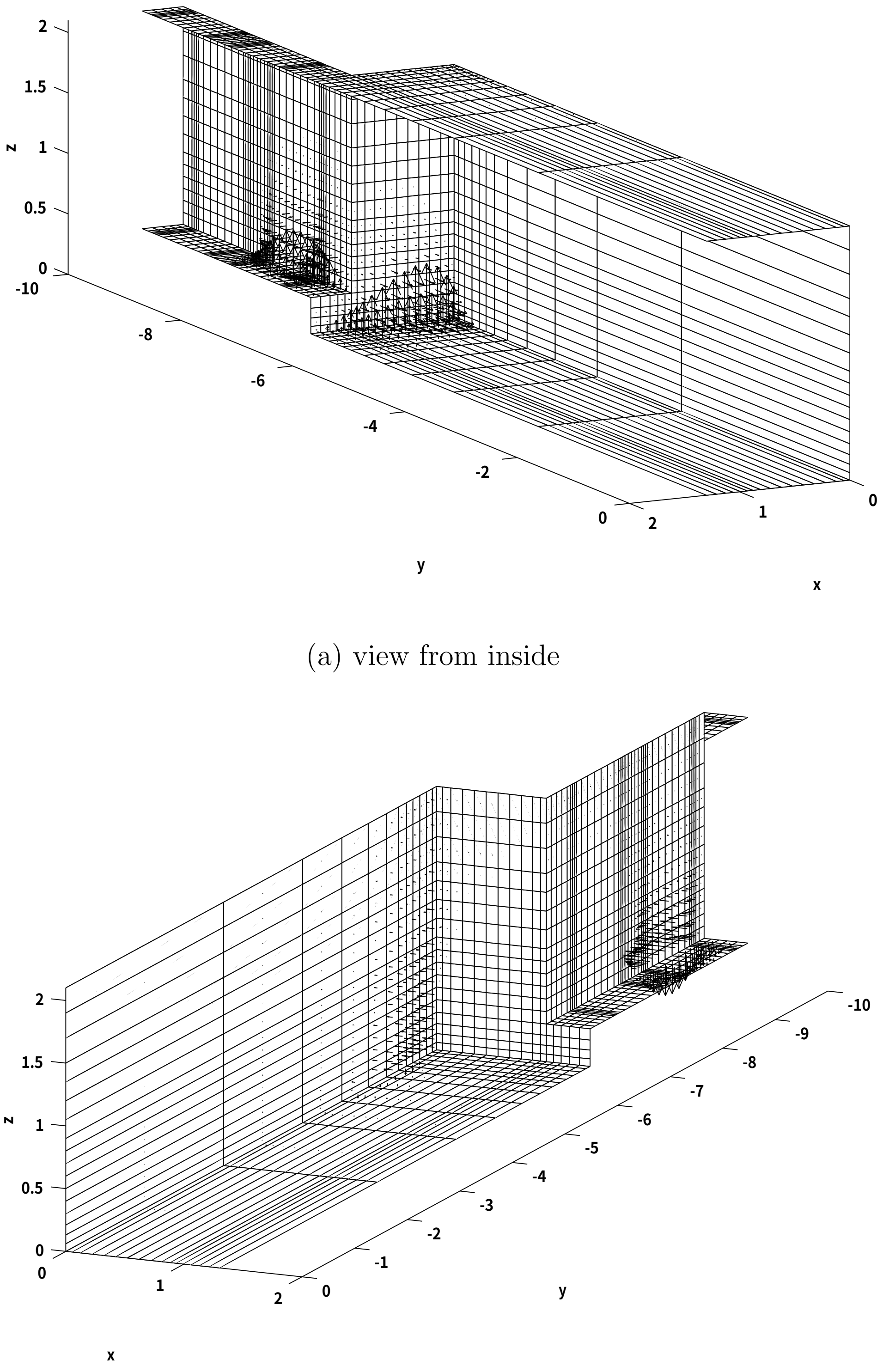
2.2 Discretization of the Problem by Finite Element Method

In this subsection, we discretize the optimization problem (3) by Finite Element Method (FEM) and cast it into a second-order cone program. We consider quarter of the body of the car as Ω in consequence of taking advantage of symmetry, and discretize $\partial\Omega$ (frontier of interior region) into 1669 rectangular finite elements (subregions) $\partial\Omega_i$ (i=1,..., 1669). The discretization is shown in Figures 4(a) and (b), together with (approximated) B_n obtained by solving the exterior field problem. As is seen in the figures, the coach and corridor are modeled as two different-sized bricks. We see that the mesh becomes dense at four places along the corridor corresponding to the ends of the two SCCs, where the change of the magnetic field is supposed to be large.

A main part of the discretization of (3) is the linear equality constraint (3b). By using the theorem of Gauss and the boundary condition on $\partial\partial\Omega$, a weak form of (3b) is given by

$$\iint_{\partial\Omega} \left[(\vec{F}, \mathrm{grad}\lambda') \ + \ \lambda' B_n\right] dS = 0,$$

where (,) means inner product of two-dimensional vectors, and operator grad is taken on two-dimensional plane (or more generally surface) $\partial\Omega$. "Test function" λ' is arbitrary function taken from space $H^1(\partial\Omega)$. We use the bilinear finite element over a rectangular domain for λ' [6]. The degree of freedom of each element is associated with the nodes of the mesh. As to $\vec{F} = (F_1, F_2)$ and F_0, we adopt constant elements. We denote by F_{0j}^* and



(a) view from inside

(b) view from outside

Figure 4: Discretization of shield surface $\partial\Omega$ and the normal component of the (approximated) external field $\widehat{B}_n^*$. (Each rectangular area is a finite element. The arrow growing from the center of each element is proportional to $\widehat{B}_n^*$.)

$\vec{F}_j^*$ the discretized F_0 and $\vec{F}$ associated with the finite element j. Degrees of freedom of all finite elements for F_0 and $\vec{F}$ are attached to center of gravity of each element and the second-order cone condition is evaluated at this point. The external field B_n in our case is an elementwise constant scalar function as an approximate solution of the external problem. We denote this approximated solution by $\widehat{B}_n^*$.

In order to complete the discretization, we have to determine the boundary conditions. We consider the following boundary conditions on $\partial\partial\Omega$. Below $\vec{n}'$ denotes outward normal of $\partial\partial\Omega$ on $\partial\Omega$.

[B1] Dirichlet-type boundary condition $\vec{F} \times \vec{n}' = \mathbf{0}$: To satisfy this condition, we set (discretized) λ' to be zero.

[B2] Neumann-type boundary condition $\vec{F} \cdot \vec{n}' = 0$: To satisfy this condition, we just let (discretized) λ' free and impose no boundary condition on $\vec{F}$ explicitly.

We applied the Dirichlet-type boundary condition to all the parts of the boundary except for the section cut by the plane $y = 0$, where we applied the Neumann-type boundary condition (cf. Figure 4). (It is not relevant to the problem which boundary condition is applied to that center part of the coach, because the magnetic field is weak there.)

The Neumann-type boundary condition [B2] corresponds to the natural boundary condition in usual finite element procedure. Concerning the Dirichlet-type boundary condition [B1], it is not apparent whether the condition is satisfied by just setting λ' to zero. However, the boundary condition is indeed satisfied “after optimization.” This is an interesting aspect of this optimization problem which we observed in the original version [24] of this paper, but we do not go into details here.

Using the quantities introduced above, (3) is discretized as follows:

$$\begin{aligned} &\text{minimize} && \frac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, && \text{(4a)}\\ &\text{subject to} && \sum_{j\in E} D_{ij}\vec{F}_j^* = (\widetilde{B}_n^*)_i, \quad i \in V, && \text{(4b)}\\ &&& \|\vec{F}_j^*\| \le F_{0j}^*, \quad j \in E, && \text{(4c)} \end{aligned}$$

where E and V are the sets of indices of finite elements and nodes of the mesh, respectively, w_j is the area of the element j, D is the FEM discretization of div operator, and $(\widetilde{B}_n^*)_i$ is the inner product of the test function with $\widehat{B}_n^*$ associated with the node i. F_{0j}^* of the optimal solution of (4) represents thickness of the optimized shield at the element j.

3 Second-Order Cone Programming and Primal-Dual Interior-point Algorithms

In this section, we formally introduce second-order cone programming (SOCP) and explain the primal-dual interior-point algorithms.

3.1 Second-Order Cone Programming

The second-order cone $\mathcal{K}(p)$ is a cone in R^p defined as follows.

$$\mathcal{K}(p) = \{x = (x_0, x_1) \in R \times R^{p-1} \mid x_0^2 - x_1^T x_1 \geq 0,\ x_0 \geq 0\}.$$

This cone is known as one of the typical examples of symmetric cones, i.e., self-dual and homogeneous cones [9]. As a convention, by $x \succeq 0$ and $x \succ 0$ we mean $x \in \mathcal{K}(p)$ and $x \in \mathrm{Int}(\mathcal{K}(p))$, respectively. We denote by $\mathcal{K}^*(p)$ the dual cone of $\mathcal{K}(p)$. Since $\mathcal{K}(p)$ is self-dual (with respect to "the ordinary inner product"), we have

$$\begin{aligned} \mathcal{K}^*(p) &= \{s \in R^p \mid x^T s \geq 0, \quad x \in \mathcal{K}(p)\} \\ &= \{s = (s_0, s_1) \in R \times R^{p-1} \mid s_0^2 - s_1^T s_1 \geq 0,\ s_0 \geq 0\} = \mathcal{K}(p). \end{aligned}$$

A second-order cone program is an optimization problem of minimizing a linear function over the intersection of an affine space and the direct product of second-order cones, and is written as follows:

$$\begin{array}{ll} \text{(P)} & \text{minimize } \displaystyle\sum_{i=1}^{n} c_i^T x_i, \\ & \text{subject to } \displaystyle\sum_{i=1}^{n} A_i x_i = b,\ x_i = (x_{i0}, x_{i1}) \succeq 0, \quad i = 1, ..., n, \end{array}$$

where $A_i \in R^{m \times k_i}$ $(i = 1, ..., n), b \in R^m, c_i \in R^{k_i}$. n denotes the number of second-order cones. Like LP and SDP, SOCP has a number of applications in many areas [16]. Obviously, our problem (4) formulated in the last section is a second-order cone program. The dual problem is defined by

$$\begin{array}{ll} \text{(D)} & \text{maximize } b^T y, \\ & \text{subject to } s_i = c_i - A_i^T y,\ s_i = (s_{i0}, s_{i1}) \succeq 0, \quad i = 1, ..., n. \end{array}$$

To simplify the notation, let

$$\begin{aligned} &A = (A_1\ A_2\ ...\ A_n) \in R^{m \times K}, \quad c = (c_1, \cdots, c_n) \in R^K, \\ &x = (x_1, ..., x_n) \in R^K, \quad s = (s_1, ..., s_n) \in R^K, \\ &\mathcal{K} = \mathcal{K}_1 \times ... \times \mathcal{K}_n, \end{aligned}$$

where $K = \sum_{i=1}^{n} k_i$. K is the number of primal (and dual) variables. Then (P) and (D) are represented in a form similar to LP:

$$\begin{array}{ll} \text{(P)} & \text{minimize } c^T x, \\ & \text{subject to } Ax = b, \quad x \succeq 0, \end{array}$$

$$\begin{array}{ll} \text{(D)} & \text{maximize } b^T y, \\ & \text{subject to } s = c - A^T y, \quad s \succeq 0, \end{array}$$

where we abuse the notation $\succeq$ in an obvious way. In fact, linear inequality constraints can be handled formally just by regarding the half-line as the one-dimensional second-order cone $\mathcal{K}(1)$. Observe that, for any feasible solutions x and (s, y) of (P) and (D), we have

$$c^T x - b^T y = x^T(c - A^T y) = x^T s \geq 0,$$

where the last equality follows from $x \in \mathcal{K}$ and $s \in \mathcal{K}^* = \mathcal{K}$. Thus, the primal objective value is always greater than or equal to the dual objective value for any feasible primal and dual solutions. The quantity $x^T s (= c^T x - b^T y)$ is referred to as "duality gap". Furthermore, if we can find the feasible solutions x and (s, y) of (P) and (D) with no duality gap, i.e, (x, s, y) satisfying the following conditions:

$$\text{(PD)} \quad s^T x = 0, \ Ax = b, \ s = c - A^T y, \ x \succeq 0, \ s \succeq 0,$$

then x and (s, y) are the optimal solutions of (P) and (D), respectively. The existence of such optimal solutions is always ensured if both (P) and (D) have interior feasible solutions, i.e., feasible solutions such that $x \succ 0$ and $s \succ 0$. In our case, we can check that this condition is satisfied and hence (P) and (D) have optimal solutions with no duality gap.

Before going to the next section, we explain how the problem (1), which contains the quadratic term $\|a_0 v\|^2$ as integrand, can be converted to (a continuous version of) a second-order cone program. It is known [16, 20] that a quadratic constraint $z^T z \leq \alpha$ is equivalent to

$$\left\| \begin{pmatrix} \alpha - 1 \\ 2z \end{pmatrix} \right\| \leq \alpha + 1, \quad \alpha \geq 0.$$

By using this result, (1) can be rewritten as the follows:

$$\begin{aligned} &\text{minimize} && \int_\Theta \left\{ u + w + a_2^T v \right\} dx, \\ &\text{subject to} && \text{div} v = b, \quad \|v\| \leq c, \\ & && \|a_1 v\| \leq w, \quad \left\| \begin{pmatrix} u - 1 \\ 2a_0 v \end{pmatrix} \right\| \leq u + 1, \quad u \geq 0, \end{aligned}$$

where u and w are continuous scalar functions in x. This way, (1) is represented as a continuous version of a second-order cone program.

3.2 Central Trajectory and Primal-dual Path-following Algorithms

The primal-dual interior point algorithms solve optimization problems by following a trajectory called central trajectory. This trajectory is defined in the interiors of feasible regions of (P) and (D), and we approach the optimal solutions of (P) and (D) along the trajectory. A formulation of the central trajectory is given on the basis of the Euclidean

Jordan algebra [9, 10, 11, 30]. We introduce the following product defined between the two elements x_i and s_i of R^{k_i}:

$$x_i \circ s_i = (x_i^T s_i, x_{i0}s_{i1} + s_{i0}x_{i1}).$$

The vector space R^{k_i} equipped with this product is the Euclidean Jordan algebra associated with the second-order cone $\mathcal{K}(k_i)$ [30]. In terms of this product, $\mathcal{K}(k_i)$ is represented as $\mathcal{K}(k_i) = \{v \mid v = w \circ w, \ \ w \in R^{k_i}\}$. The element $e_i = (1, 0, ..., 0)$ is a unit element of this algebra. The Euclidean Jordan algebra can be easily extended to the whole space by defining the product

$$x \circ s = (x_1 \circ s_1, ..., x_n \circ s_n),$$

where $e = (e_1, ..., e_n)$ is the unit element.

Observe that $x \circ s = 0$ implies that $x^T s = 0$, because

$$e^T(x \circ s) = \sum_i e_i^T(x_i \circ s_i) = \sum_i x_i^T s_i = x^T s. \tag{6}$$

Thus, the problems (P) and (D) are formulated as finding feasible solutions x and (s, y) of (P) and (D) satisfying the condition $x \circ s = 0$.

The central trajectory of (P) and (D) is defined as the set of the solutions of the following parameterized system of bilinear equations with the parameter $\nu \in (0, \infty]$.

$$x \circ s = \nu e, \quad Ax = b, \quad A^T y + s = c, \quad x \succeq 0, \quad s \succeq 0, \tag{7}$$

where $e = (e_1, ..., e_n)$. Under the assumption of existence of interior feasible solutions of (P) and (D), it is known that (7) defines a smooth path in the interiors of the feasible regions of (P) and (D) which approaches the optimal sets as ν tends to zero [11]. The solution of (7) is referred to as the center point (with parameter ν). Due to (6), (7) and the fact that $e^T e = n$, the relation $\nu = x^T s/n$ holds on the central trajectory.

The primal-dual path-following algorithms solve second-order cone programs by following the central trajectory with the Newton method (or its variant) based on this formulation. We generate a sequence in the interior of the cone $\mathcal{K} \times \mathcal{K}$ by solving approximately the equation (7) repeatedly, reducing ν to zero. Typically, we start from an initial point $(x, s, y) = (\nu_0 e, \nu_0 e, 0)$ with $\nu_0 > 0$, which is in the interior of the primal-dual cone, and restrict the iterates to stay inside the cone in the subsequent iterations. The Newton direction for (7) is called AHO direction [2]. Many primal-dual interior-point algorithms adopt variants of the Newton direction called the scaled Newton directions. There are three well-known scaled Newton directions called the Helmberg-Rendl-Vanderbei-Wolkowicz/Kojima-Shindoh-Hara/Monteiro (HRVW/KSH/M) direction [13, 14, 18], the HRVW/KSH/M dual direction [18] and the Nesterov-Todd (NT) direction [21].

A generic primal-dual path-following algorithm for SOCP is described as follows.

Generic Algorithm:

Let $\varepsilon \in (0, 1)$ and $\theta \in (0, 1)$ be precision parameter and step-size parameter, and let $(x^0, s^0, y^0) \in R^K \times R^K \times R^m$ be a point such that $(x^0, s^0) \in \text{Int}(\mathcal{K}) \times \text{Int}(\mathcal{K})$. Let $\mu^0 := ((s^0)^T x^0)/n$ and set $k := 0$.

```
Repeat until $\mu^k \leq \varepsilon\mu^0$ do
    1. Let $(x, s, y) := (x^k, s^k, y^k)$ and $\mu := \mu^k$.
    2. Determine $\sigma \in (0, 1)$.
    3. Compute a Newton-type search direction $(\Delta x, \Delta s, \Delta y)$ at $(x, s, y)$ for the
       center point with the parameter $\nu := \sigma\mu$.
    4. Choose the step-size $\alpha > 0$ which brings the iterate to the fraction $\theta$ of the
       way to the boundary of the primal-dual cone, and let $(x^{k+1}, s^{k+1}, y^{k+1}) :=$
       $(x, s, y) + \alpha(\Delta x, \Delta s, \Delta y) \in \mathrm{Int}(\mathcal{K}) \times \mathrm{Int}(\mathcal{K}) \times R^m$.
    5. Set $\mu^{k+1} := ((x^{k+1})^T s^{k+1})/n$ and increment $k$ by 1.
End
```

Note that the search direction in Step 3 aims at the center point which would reduce the duality gap by a factor of σ.

There are two well-known versions of algorithms in implementation. The first one is the algorithm which we call “Basic algorithm.” In this algorithm we take σ to be a constant and use the Newton direction or the scaled Newton directions as the search direction without modification. The other one is called “Mehrotra Predictor-Corrector (MPC) algorithm” and incorporates with adaptive update of σ and modification of search directions taking account of a second-order correction. The MPC algorithm is a standard technique to speed up the primal-dual interior-point algorithms [17, 32, 29].

The major task in one iteration of the primal-dual path following algorithms is computation of search directions. Like in other interior-point algorithms for LP and SDP, this part consists of (i) computation of the Shur complement matrix which is the $m \times m$ coefficient matrix of a system of linear equations to determine a search direction, and (ii) solution of the system of linear equations. They require $O(m^2K)$ and $O(m^3)$ arithmetic operations, respectively, assuming that A is dense. However, the number of arithmetic operations can be much less when we deal with such a sparse problem like the one treated in this paper. We do not go into details of computation of the search directions. We refer interested readers to [30] and [3].

4 Computational Results and Analysis

We implemented the primal-dual algorithms to solve the optimal magnetic shielding problems explained in Sections 2 and 3. All experiments are conducted on a personal computer with dual Pentium III 700MHz CPUs, 1GB main memory and Windows NT 4.0 operating system. (But only one processor was used in the computation.) The code is written in Fortran and compiled with Microsoft Visual Fortran Vers. 5. for Windows. All floating-point computations are executed with double-precision.

The problem is as explained in Section 2 and Section 3. The number of second-order cone constraints is 1669, where the dimension of each cone is three, and hence the number of primal and dual variables are $5007 (= 1669 \times 3)$ for each. The number of y variables is 1646. In this problem, the matrix A is sparse. The number of second-order cone constraints is large but their dimensions are the same and small. The techniques for taking advantage of sparsity in solving second-order cone programs of this type are very similar to the ones in LP. We exploited these special structures in computing the search direction, namely, both in forming the Shur complement matrix and solution of the resulting sparse system of linear equations. We employed the sparse Cholesky factorization routines in IMSL attached to Visual Fortran. In the following, we report the results with a sparse implementation of the primal-dual algorithm with the NT direction.

4.1 Performance of the Algorithm

The algorithm very quickly converged to optimal solutions. The problem was solved in 1.8 sec and 21 iterations with MPC algorithm, and 3.3 sec and 41 iterations with Basic algorithm. (The iteration was stopped when the duality gap was smaller than 10^{-12}. Feasibility was also satisfied with the same level in order.) In order to observe dependency of performance of the algorithm on the size of the problem, we constructed larger problems by dividing each rectangular finite element by $k \times k$ ($k = 2, ..., 6$). We report the result for the largest problem with $k = 6$. The number of second-order cones of this problem is 60084, the number of primal/dual variables of this extended problem is 180252 ($= 60084 \times 3$), and the number of y variables is 59850. This problem was solved in 725 sec and 34 iterations with MPC algorithm, and 1940 sec and 110 iterations with Basic algorithm, with the same stopping condition as above.

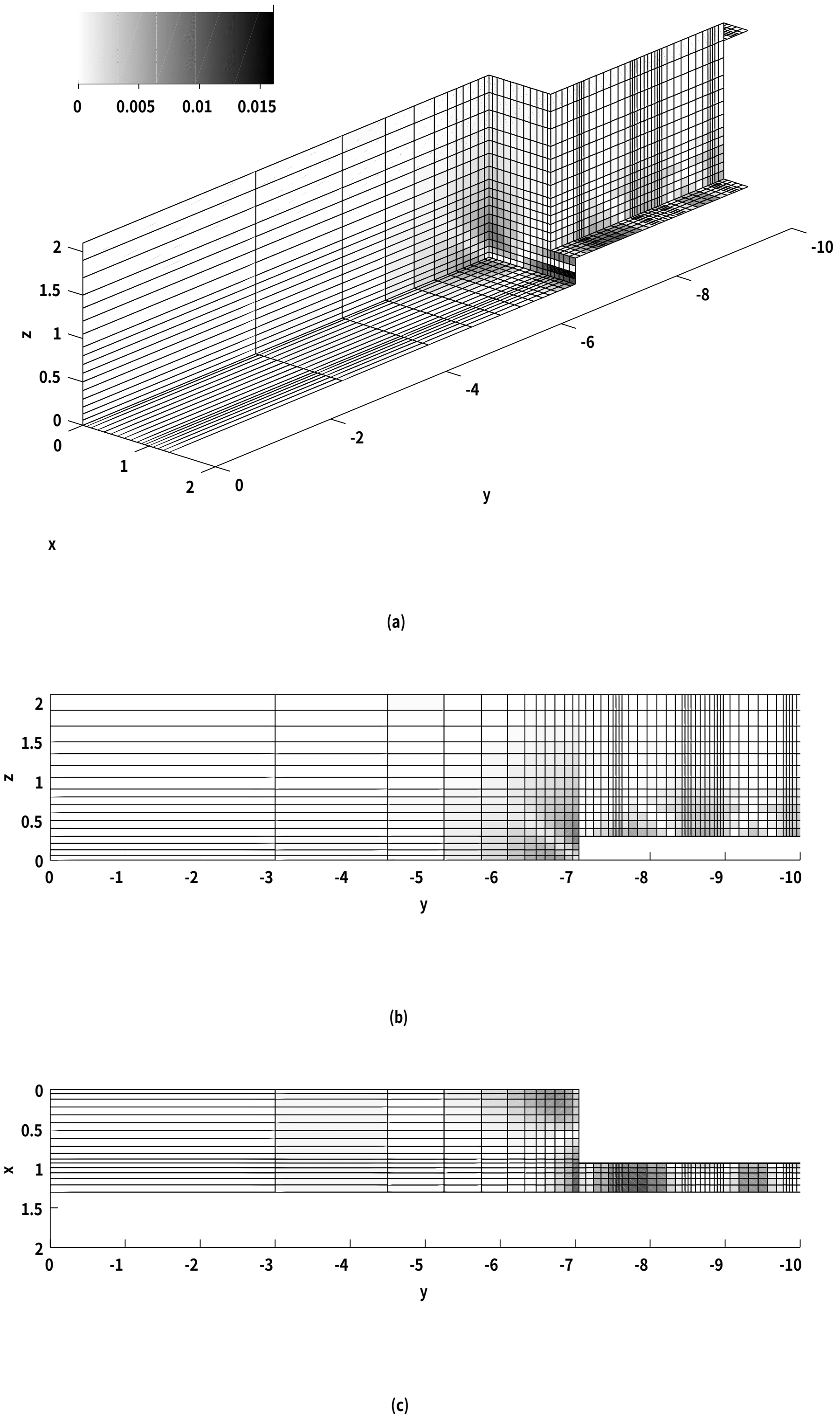
Our code is specialized to this design problem, but the timing data reported here suggests that it is at least comparable in speed with other well-known SOCP codes like MOSEK [3] and SeDuMi [27] for this type of problem. See [24] for more detailed results of numerical experiments where we compared performance of the four major search directions AHO, HRVW/KSH/M, HRVW/KSH/M dual and NT.

4.2 Optimized Design

Now we analyze the optimized design from the physical point of view. As was mentioned in Section 2, we took $B_s = 1.5$ Tesla. The optimal value, which represents one quarter of the volume of the shield, is $2.525069751 \times 10^{-2}\text{m}^3$. In reference to the stopping criteria we adopted, this value is considered to be correct up to the order of 10^{-12}. The volume of shielding of the car is $1.0100279004 \times 10^{-1}\text{m}^3$ ($= 4 \times 2.525069751 \times 10^{-2}\text{m}^3$). The obtained design is shown in Figure 5. The units of x, y and z-axis and bargraph in Figure 5 are meter.

Before explaining this figure, we describe the magnetic field $\widehat{B}_n^*$ generated by the SCM units. As was explained in Section 2.2, the area consists of the coach and the corridor (cf. Figures 1 and 4). The coach is enclosed by the bottom, side, ceiling and end plates, whereas the corridor is enclosed by the bottom, side and ceiling plates.

First we explain the incoming/outgoing magnetic flux in the coach. See Figure 4(a). The flux goes into the shield at the corridor side of the bottom plate and the lower part of the end plate. On the other hand, as is seen in Figure 4(b), the flux goes out from the part of the side plate closer to the end plate.



(a)

(b)

(c)

Figure 5: Optimized shielding design (The rectangular on the upper-left of the figure shows gray-scale representing thickness in meter.) (a) view from lower left side; (b) view from left side; (c) view from the bottom.

The magnetic flux in the corridor is as follows. See Figure 4(a). The lower parts of the side plate and the bottom plate on the coach side are sinkers of the magnetic flux. On the other hand, as is seen in Figure 4(b), the magnetic flux goes out from the further lower part of the side plate and the bottom plate (seen from the coach). This is a rough sketch of the magnetic flux $\vec{F}^*$ on the surface generated by the SCM units.

Now we explain the magnetic flux inside the optimized shield. See Figure 5. Basically, at the bottom plate of the corridor, the magnetic flux flows along the guidance direction (x-direction), because the SCCs placed on this side and the opposite side of the corridor are magnetized in the same direction, as was explained in Section 2.1. The shield becomes thicker two times at the bottom of the corridor along the direction of travel (y-direction) around $y = -7.8$ and $y = -9.5$. These thick parts correspond to the place where the flow is strong. The direction of the magnetic flux $\vec{F}^*$ is opposite to each other at the both thick parts.

Secondly in the coach, the magnetic flux flows from the bottom plate to the side plate and also flows from the end plate to the side plate. This is because the end plate and the bottom plate are sinkers of magnetic flux from the SCM units and magnetic flux goes back from the side plate into the SCM units. Reflecting this flow of magnetic flux, the shielding becomes thicker around $(x, y, z) \sim (0, -7, 0.3)$ and $(x, y, z) \sim (0, -7, 0)$. However, the total outgoing flux from the side plate is not sufficient to balance the incoming flux to the end plate and the bottom plate. Therefore, to satisfy the total balance of the flux, there exists a strong flow in the guidance direction to the other side of the coach through the part beneath the corridor. This is why the thickest part exists at the lower and center part of the end plate below the corridor ($(x, y, z) \sim (1.2, -7, 0.1)$). It is interesting to note that the magnetic flux B_n at this place is not very strong as is seen from Figures 4(a) and (b). Nevertheless, the shield has to be thick there to satisfy the conservation law of the flow $\vec{F}^*$.

4.3 Comparison with the Previous Approach

In this subsection, we compare our result with the previous result obtained by one of the authors [23]. As is formulated in Section 2, the magnetic shielding design problem is a problem of minimizing the sum of weighted Euclidean norm $\sum_{j \in E} w_j \|\vec{F}_j^*\|$ subject to the linear constraint (4b). In the previous paper [23], a solution of this problem was obtained by iteratively solving the following weighted least squares problem:

$$\vec{F}^{*(k+1)} = \mathrm{argmin}_{\vec{F}^*} \left\{ \sum_{j \in E} w_j \frac{\|\vec{F}_j^*\|^2}{\|\vec{F}_j^{*(k)}\|} \; : \; \vec{F}^* \text{ satisfies (4b)} \right\}, \qquad k = 0, 1, ...$$

In the following, this method is referred to as the iterative least squares (ILS) algorithm. The number of arithmetic operations per iteration is more or less the same as the primal-dual (PD) algorithm.

We compare the ILS algorithm and the primal-dual (PD) algorithm. With the ILS algorithm, the objective function converged in 184 iterations to $2.525117105 \times 10^{-2}$ before the running process broke down due to numerical difficulty. The obtained optimal value of the ILS algorithm is correct just up to the order of $10^{-7} \sim 10^{-8}$, which is much worse than the accuracy of 10^{-12} attained by the PD algorithm. Furthermore, the number of iterations is much less with the PD algorithm. Thus, the PD algorithm is superior to the ILS algorithm both in efficiency and accuracy. Another advantage of the PD algorithm is availability of a lower bound of the optimal value with the dual objective value. With the ILS algorithm, there is no automatic way to obtain such a lower bound.

We observed that the global structures of the both designs are similar, but thickness considerably differs at some places even though the objective function values of the two designs are very close. In particular, the two designs differ at the bottom part of the end plate beneath the corridor, where the shield becomes thickest. The design by the ILS algorithm is smooth compared with the design by the PD algorithm. This is because the design by the ILS method is not yet optimized. When optimization is complete, the solution tends to be nonsmooth. From an engineering point of view, a smooth solution would be preferable. When we use the interior-point algorithms, we can incorporate such smoothness conditions into the formulation explicitly. For example, we may require upper bounds on $\max \alpha_{ij}|F^*_{0i} - F^*_{0j}|$, $\sum \beta_{ij}|F^*_{0i} - F^*_{0j}|$ or $\sum \gamma_{ij}(F^*_{0i} - F^*_{0j})^2$, where $\alpha_{ij}, \beta_{ij}, \gamma_{ij}$ are appropriate weights and the maximum and summation are taken over all two neighbor elements which share an edge. We may also modify the objective function by adding these terms with appropriate weights. All of these modifications can still be cast as second-order cone programs (with additional linear inequalities).

5 Application to Robust Optimization

We apply the primal-dual interior-point algorithm to robust design of our magnetic shield design problem. Our physical model for the shielding design contains some approximation errors and uncertainty parts. Therefore, ideally, optimization should be done over “the set of shielding which would function even when such errors and uncertainty are taken into account.” This type of meta-optimization approach is called robust optimization in more general context and has been extensively studied recently [4, 5, 7, 8, 12]. The meta-optimization problem taking into account of the worst-case scenario is called “robust counterpart” of the original optimization problem. Tractability of the robust counterpart depends on the original optimization problem and the shape of the region of uncertainty we consider, but these recent studies revealed that there are several interesting and useful cases where the robust counterpart can be formulated as tractable convex programs such as SOCP and SDP. Here we consider a robust counterpart of the magnetic shielding problem, and make an attempt to solve it approximately by solving perturbed problems repeatedly taking advantage of efficiency and stability of the primal-dual interior-point algorithm.

In the following, we use the same notations as in Subsection 2.2. Main sources of errors and uncertainty incurred in the model are the FEM discretization and the approximate solution of the exterior field problem. We denote by B^*_n the approximated elementwise constant function of the exterior field B_n, and focus on uncertainty in B^*_n. Specifically, we assume that uncertainty incurred in B_n^* is represented as the following set Γ:

$$\Gamma \equiv \{B_n^* \in R^{|E|} |\ (B_n^*)_j = (\widehat{B}_n^*)_j + (\Delta B_n^*)_j, \quad |(\Delta B_n^*)_j| \leq 0.05|(\widehat{B}_n^*)_j|, \quad j \in E\}, \tag{8}$$

where $\widehat{B}_n^*$ is the concrete approximated elementwise constant function given in Subsection 2.2 based on which optimization in the previous section is done. Intuitively, we assume that "10% elementwise relative error" can be incurred in $\widehat{B}_n^*$.

Let $G_i(B_n^*)$ be the inner product of B_n^* with the test function associated with the ith node, and define $G(B_n^*) = (G_1(B_n^*), ..., G_{|V|}(B_n^*))$. Then we have $\widetilde{B}_n^* = G(\widehat{B}_n^*)$, where $\widetilde{B}_n^*$ is as defined in Section 2.2 (cf.(4b)). Possible changes on the righthand side of (4b) when B_n^* is assumed to be in the box Γ is given by $G(\Gamma)$. Note that $G(\Gamma)$ is a polyhedral set in $R^{|V|}$.

Now we are ready to present the robust counterpart of our magnetic shielding problem. Designed shielding is represented by the vector $F_0^* \equiv (F_{01}^*, ..., F_{0|E|}^*)$, where F_{0i}^* is thickness of the ith element. Robustness of the shielding F_0^* means that

"For each possible approximated external field $B_n^* \in \Gamma$, there exists a feasible flow of magnetic flux $\vec{F}^* \equiv (\vec{F}_1^*, ..., \vec{F}_{|V|}^*)$ satisfying

$$\sum_{j\in E} D_{ij}\vec{F}_j^* = G_i(B_n^*), \quad i \in V, \quad \text{and} \quad \|\vec{F}_j^*\| \leq F_{0j}^*, \quad j \in E.\text{''} \tag{9}$$

If this condition is satisfied, we say that F_0^* is robust. We denote by Θ the set of robust F_0^*. In the following, the flow $\vec{F}^*$ satisfying (9) is referred to as "a feasible flow with respect to B_n^*," indicating dependency on B_n^* explicitly. Now the robust counterpart of the shielding design problem is written as

$$\text{minimize } \frac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, \quad \text{subject to } F_0^* \equiv (F_{01}^*, ..., F_{0|E|}^*) \in \Theta, \tag{10}$$

where

$$\Theta \equiv \{F_0^* \in R^{|E|} \mid \text{For each } B_n^* \in \Gamma, \text{ there exists a feasible flow } \vec{F}^* \text{ satisfying (9).}\}$$

This robust counterpart is a convex semi-infinite program, where the convexity of Θ is readily verified by using the triangular inequality. Unfortunately, this problem is unlikely to afford a tractable convex program reformulation. Therefore, we consider to solve it approximately. A simple way to solve (10) approximately is to replace Γ with the set of N points $(B_n^*)^1, ..., (B_n^*)^N$ sampled from Γ. Let $\Gamma_N \equiv \{(B_n^*)^1, ..., (B_n^*)^N\}$. Then the approximated robust counterpart is written as

$$\text{minimize } \frac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, \quad \text{subject to } F_0^* \in \Theta_{\text{Finite}}, \tag{11}$$

where

$$\Theta_{\text{Finite}} \equiv \{F_0^* \in R^{|E|} \mid \text{For each } (B_n^*)^k \in \Gamma_N, \text{ there exists a feasible flow } (\vec{F}^*)^k \text{ satisfying (9).}\}$$

Since Γ_N is a finite set, it is not difficult to see that (11) is rewritten as the following second-order cone program with $|E| \times N$ second-order cone constraints:

$$\begin{array}{ll} \text{minimize} & \dfrac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, \\ \text{subject to} & \sum_{j\in E} D_{ij}(\vec{F}_j^*)^k = G_i((B_n^*)^k), \quad i \in V, \quad k = 1,..,N, \\ & \|(\vec{F}_j^*)^k\| \leq F_{0j}^*, \quad j \in E, \quad k = 1,...,N. \end{array} \tag{12}$$

Even though the problem (11) and (12) appear to be a finite point approximation, it is worthwhile to note that the optimal solution of (11) is the optimal solution of the following convex semi-infinite program where Γ in the original robust counterpart (10) is replaced by the convex hull $\text{conv}(\Gamma_N)$ of Γ_N.

$$\text{minimize} \quad \frac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, \quad \text{subject to} \quad F_0^* \in \Theta_{\text{Conv}}, \tag{13}$$

where

$\Theta_{\text{Conv}} \equiv \{F_0^* \in R^{|E|} \mid$ For each $B_n^* \in \text{conv}(\Gamma_N)$, there exists a feasible flow $\vec{F}^*$ satisfying (9).$\}$

Equivalence of these two problems (11) and (13) is seen as follows. $\Theta_{\text{Conv}} \subseteq \Theta_{\text{Finite}}$ is obvious by definition. To show the reverse inclusion, given $F_0^* \in \Theta_{\text{Finite}}$, we will show that, for any $B_n^* \in \text{conv}(\Gamma_N)$, there always exists a feasible flow $\vec{F}^*$ satisfying (9). Let $(\vec{F}^*)^k$ be a feasible flow with respect to $(B_n^*)^k \in \Gamma_N$. Then $\|(\vec{F}^*)^k\| \leq F_0^*$ holds for all $k = 1,...,N$. Let us represent B_n^* as a convex combination of $(B_n^*)^1,...,(B_n^*)^N$ as $B_n^* = \sum_k \beta_k (B_n^*)^k$, where $\beta \in R^N$, $\sum \beta_k = 1$, $\beta \geq 0$. By using the triangular inequality, it is easy to see that $\vec{F}^* = \sum_k \beta_k (\vec{F}^*)^k$ is a feasible flow with respect to B_n^* satisfying $\|\vec{F}^*\| \leq F_0^*$. Thus, it is always possible to find a feasible flow with respect to any $B_n^* \in \text{conv}(\Gamma_N)$ as long as $F_0^* \in \Theta_{\text{Finite}}$. This implies that $F_0^* \in \Theta_{\text{Conv}}$ and completes the proof.

Now, the set $\text{conv}(\Gamma_N)$ converges to Γ as N tends to infinity if the sampling procedure is carried out properly. Then the optimal solution of (11) approaches the optimal solution of the robust counterpart (10). Thus, it makes some sense to work with the approximation (11). Based on the relationship among (10), (11) and (13) exploited here, one might further develop a concrete theoretical and empirical analysis of performance of (11) or (13) as an approximation to (10).

But the problem is that even the approximated robust counterpart (11) is very difficult to solve, because the number of second-order cones involved in (11) (i.e., (12)) is $|E| \times N$ and hence it can be a huge problem. In our case, $|E| = 1669$ and N needs to be at least several thousands. Therefore, we develop a heuristic procedure to solve (12) approximately which we describe below.

For simplicity, we call points in Γ_N as perturbed external field, and $(\Delta B_n^*)^k = (B_n^*)^k - \widehat{B}_n^*$, $k = 1,...,N$ (cf. (8)) as perturbation. We assume that $(\Delta B_n^*)_j^k$ obeys to independent uniform distribution $U[-0.05|(\widehat{B}_n^*)_j|, 0.05|(\widehat{B}_n^*)_j|]$ for each k and j. In the procedure described below, we solve a second-order cone program associated with each perturbed external field one by one repeatedly. The idea is to increase thickness of the elements bit by bit at each iteration so that feasibility of the flow is maintained with the minimum increase of the cost as we solve the perturbed problems. T_j^k is the thickness of the element j of the obtained design at the k-th iteration.

Procedure for Robust Optimization

1. Let N be the total number of iterations. Set $k := 0$, $T_j^0 := 0$ for all $j \in E$, and $(\Delta B_n^*)_j^0 := 0$ for all $j \in E$ (initialization).
2. If $k \geq 1$ then generate $(\Delta B_n^*)_j^k$ $(j \in E)$ by drawing from the uniform distribution $U[-0.05|(\widehat{B}_n^*)_j|,\, 0.05|(\widehat{B}_n^*)_j|]$.
3. Compute the right hand side $G^k \equiv G(\widehat{B}_n^* + (\Delta B_n^*)^k)$ of (4b) associated with the perturbed external field $\widehat{B}_n^* + (\Delta B_n^*)^k$.
4. Solve the second-order cone program

$$\begin{array}{ll} \text{minimize} & \displaystyle\frac{1}{B_s}\sum_{j\in E} w_j F_{0j}^*, \\ \text{subject to} & \displaystyle\sum_{j\in E} D_{ij}\vec{F}_j^* = G_i^k, \quad i \in V, \quad \max\left[T_j^k, \|\vec{F}_j^*\|\right] \leq F_{0j}^*, \quad j \in E. \end{array}$$

   For each $j \in E$, Let $\widehat{F}_{0j}^*$ be the value of F_{0j}^* at the obtained optimal solution.

5. $T_j^{k+1} := \widehat{F}_{0j}^*$ for all $j \in E$.
6. $k := k + 1$.
7. If $k = N$ then stop; else go to Step 2.

The quantity $\sum_j w_j T_j^k$ represents the volume of the shield at the $(k-1)$-th iteration. Obviously, for any $k \geq 1$, $T_j^k, j \in E$ represents a design which is feasible for all perturbations $(\Delta B_n^*)^l$, $l = 0, ...k-1$. This implies that $T^N \equiv (T_1^N, ..., T_{|E|}^N) \in \Theta_{\text{Finite}}(= \Theta_{\text{Conv}})$. As was mentioned before, by taking N large enough, we obtain a robust design taking account of uncertainty incurred in B_n^*.

We implemented this procedure with the primal-dual algorithm (the basic algorithm with NT direction; sparse implementation). The computational environment is the same as in Section 4. The second-order cone program solved at Step 3 contains 1669 three-dimensional second-order cone constraints and the same number of linear inequality constraints. Thus, the number of primal (dual) variables is 6676 $(= 1669 \times 3 + 1669)$. The number of y variables is $3315 (= 1669 + 1646)$. We stop the iteration when the duality gap becomes less than 1.0×10^{-8}, and it is confirmed that feasibility is satisfied to the same level in order.

We run the procedure with $N = 10000$. It took 21457 sec to run the whole procedure by solving 10000 second-order cone programs. The first iteration gives the optimal solution 2.52507×10^{-2} for the original problem because $(\Delta B_n^*)^0 = 0$ and $T^0 = 0$. After 10000 iterations, the volume is increased to 2.55321×10^{-2}, which is 101.1% of the original optimal design. In Figure 6, we plot volume vs. the number of iterations. It is observed that the volume of the designed shield saturates as k increases. We do not show the picture of the obtained robust design because it is almost the same as the original one, but we can say that robustness is to a considerable extent improved at the cost of increase of 1.1% of weight.

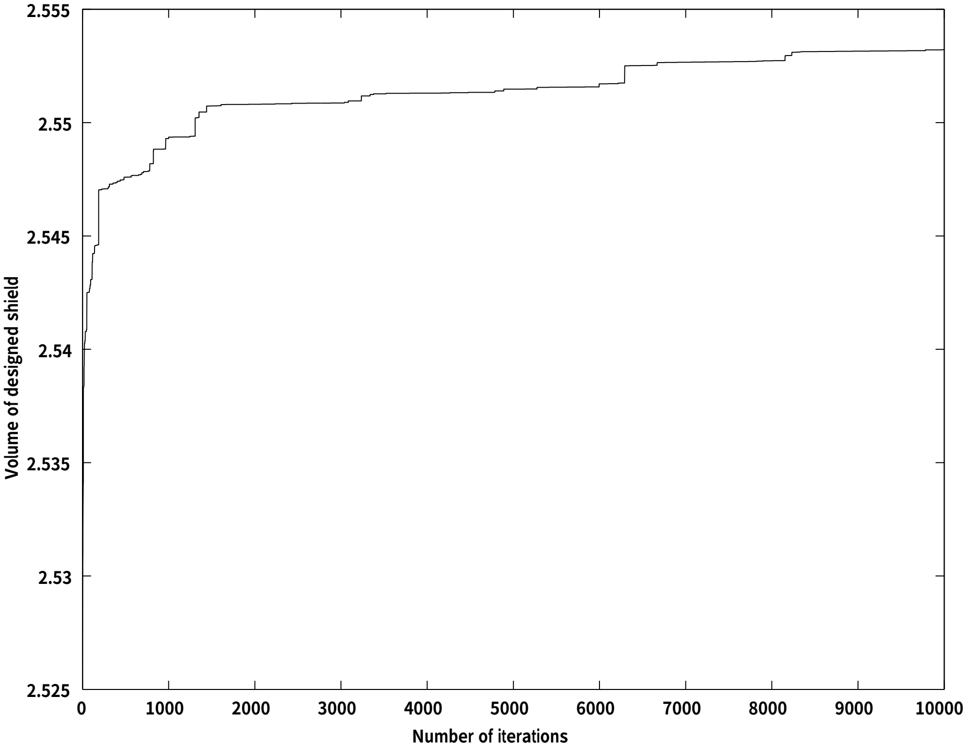


Figure 6: The volume of designed shield (unit :10^{-2}m^3) vs. the number of iterations

One might think that this is a naïve and heuristic approach. It gives a conservative approximate solution to the approximated robust counterpart (11), and the obtained solution can be too optimistic in view of robustness if the finite point (or polyhedral) approximation of the robust counterpart is crude. But on the other hand, from the engineering point of view, one might well be satisfied, since the method provided a fairly good approximate optimal solution whose robustness is guaranteed for perturbations within the convex hull of 10000 uniformly sampled points drawn from the domain of uncertainty. Anyway, this is a robust optimization problem which we need to solve in reality, and there seems no nice way to solve such a problem without any approximation. In this respect, it is a reasonable approach to the problem with which we can be much more confident of the resulting design. We emphasize that this type of robust optimization by simulation cannot be done without a stable and efficient algorithm like the primal-dual interior-point algorithm.

6 Conclusion

In this paper, we dealt with the continuous version of the convex network flow problem which can be formulated as a (continuous version of) second-order cone program. We proposed to apply the primal-dual interior-point algorithms for second-order cone programming to solve the problem after appropriate discretization. In particular, the magnetic shielding design problem for the MAGLEV train was formulated as the continuous version of the sum of Euclidean norm problem, and was solved successfully with finite element discretization and the primal-dual interior-point algorithms. The optimal design was examined from the physical point of view, and was compared with the one obtained by the previous approach. It was confirmed that the method can solve the problem efficiently with high accuracy compared with the previous approach, providing a nice lower bound of the optimal value. As a further application of the primal-dual interior-point algorithm, we developed a heuristic procedure for robust optimization. The procedure, which requires solution of thousands of perturbed design problems, was successfully implemented with the primal-dual algorithm to obtain a reasonable robust design. Further analysis of performance of the proposed procedure and development of a more sophisticated procedure of robust optimization is an interesting topic for further research.

Acknowledgment

The authors would like to thank Mr. Naotsugu Nozue of Mathematical Modelling Co., Japan for his helpful comments and suggestions in early stage of this research. The authors are grateful to Mr. Shingo Horiuchi of Department of Mathematical Engineering, University of Tokyo, Japan (currently NTT corporation, Japan) for his help in providing pictures of this paper.

References

[1] Farid Alizadeh: Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization* Vol. 5 (1995), pp. 13–51.

[2] F. Alizadeh, J.-P. Haeberly and M. Overton: Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM Journal on Optimization*, Vol. 8 (1998), pp. 746–768.

[3] E. D. Andersen, C. Roos and T. Terlaky: On implementing a primal-dual interior-point method for conic quadratic optimization, Manuscript, Helsinki School of Economics and Business Administration, Finland, December, 2000.

[4] A. Ben-Tal and A. Nemirovski: Robust convex optimization. *Mathematics of Operations Research*, Vol. 23 (1998), pp. 769–805.

[5] A. Ben-Tal and A. Nemirovski: *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. SIAM*, Philadelphia, PA, USA, 2001.

[6] P. G. Ciarlet: *The Finite Element Methods for Elliptic Problems*, North-Holland, 1978.

[7] Laurent El Ghaoui, Herve Lebret: Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, Vol. 18 (1997), pp. 1035–1064.

[8] Laurent El Ghaoui, Francois Oustry, and Herve Lebret: Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, Vol. 9 (1999), pp. 33–52.

[9] J. Faraut and A. Korányi: *Analysis on Symmetric Cones*, Oxford University Press, New York, 1994.

[10] L. Faybusovich: Jordan algebras, symmetric cones and interior-point methods. Manuscript, Department of Mathematics, Notre Dame, IN, USA, 1995.

[11] L. Faybusovich: Linear systems in Jordan algebras and primal-dual interior-point algorithms. *Journal of Computational and Applied Mathematics*, Vol. 86 (1997), pp. 149–175.

[12] D. Goldfarb and G. Iyengar: Robust portfolio selection problems. Technical Report, Department of Industrial Engineering and Operations Research, Columbia University, New York, USA, December 2001.

[13] C. Helmberg, F. Rendl, R. J. Vanderbei and H. Wolkowicz: An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, Vol. 6 (1996), pp. 342–361.

[14] M. Kojima, S. Shindoh and S. Hara: Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, Vol. 7 (1997), pp. 86–125.

[15] D. G. Lions: *Inequalities in Mechanics and Physics*. Springer-Verlag, 1976.

[16] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret: Applications of second-order cone programming. *Linear Algebra and its Applications*, Vol. 284 (1998), pp. 193–228.

[17] S. Mehrotra: On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, Vol. 2 (1992), pp. 575-601.

[18] R. D. C. Monteiro: Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization*, Vol. 7, No. 3 (1997), pp. 663–678.

[19] R. D. C. Monteiro and T. Tsuchiya: Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-Family of directions. *Mathematical Programming*, Vol. 88 (2000), pp. 61–83.

[20] Yu Nesterov and A. Nemirovskii: *Interior Point Polynomial Methods in Convex Programming*. SIAM Publications, Philadelphia, Pennsylvania, USA, 1994.

[21] Y. E. Nesterov and M. Todd: Self-scaled barriers and interior-point methods for convex programming; *Mathematics of Operations Research*, Vol. 22 (1997), pp. 1-42.

[22] Railways Technical Research Institute home page: http://www.rtri.or.jp

[23] T. Sasakawa and N. Tagawa : Preliminary design of Magnetic Shielding by FEM. *IEEE Transactions on Magnetics*, Vol.33 (1997), No.2, pp.1951-1954.

[24] T. Sasakawa and T. Tsuchiya : Optimal magnetic shield design with second-order cone programming. Research Memorandum No. 775, The Institute of Statistical Mathematics, October 2000, (Revised: April 2001).

[25] H. Sasaki: Magnetic Shielding. Kawasaki Steel Technical Report (in Japanese), Vol. 25, No. 1 (1993), pp. 64-70.

[26] S. Schmieta and F. Alizadeh: Associative algebras, symmetric cones and polynomial time interior point algorithms. Report RRR 17-98, RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, 1998.

[27] J.F. Sturm: Using SeDuMi 1.02, A matlab toolbox for optimization over symmetric cones. *Optimization Method and Software*, Vol. 11&12 (1999), pp. 625-653.

[28] A. Taguchi and M. Iri: Continuum approximation to dense networks and its application to the analysis of urban road networks. *Mathematical Programming Study*, No. 20 (1982), pp. 178–217.

[29] M. J. Todd, K. C. Toh and R. H. Tütüncü: On the Nesterov-Todd direction in semidefinite programming. *SIAM Journal on Optimization*, Vol. 8 (1998), pp. 769–796.

[30] T. Tsuchiya: A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. *Optimization Methods and Software*, Vol. 11&12 (1999), pp. 141–182.

[31] H. Wolkowicz, R. Saigal and L. Vandenberghe (eds.): *Handbook of semidefinite programming. Theory, algorithms, and applications*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000.

[32] S. Wright: *Primal–dual Interior Point Algorithms*, SIAM Publications, Philadelphia, Pennsylvania, USA, 1997.