SANDIA REPORT

SAND2017-13410
Unlimited Release
Printed September 2017

LDRD Report: Topological Design Optimization of Convolutes in Next Generation Pulsed Power Devices

E. C. Cyr, G. von Winckel, D. P. Kouri, T. Gardiner, D. Ridzal, J. N. Shadid, S. Miller

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology and Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2017-13410
Unlimited Release
Printed September 2017

LDRD Report: Topological Design Optimization of Convolutes in Next Generation Pulsed Power Devices

Eric C. Cyr, John N. Shadid, Sean Miller
Computational Mathematics
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
eccyr@sandia.gov, jnshadi@sandia.gov, seamill@sandia.gov

Greg von Winckel, Drew P. Kouri, Denis Ridzal
Optimization and Uncertainty Quantification
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
gvonwin@sandia.gov, dpkouri@sandia.gov, dridzal@sandia.gov

Thomas Gardiner
HEDP Theory
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
tagardi@sandia.gov

Abstract

This LDRD project was developed around the ambitious goal of applying PDE-constrained optimization approaches to design Z-machine components whose performance is governed by electromagnetic and plasma models. This report documents the results of this LDRD project. Our differentiating approach was to use topology optimization methods developed for structural design and extend them for application to electromagnetic systems pertinent to the Z-machine. To achieve this objective a suite of optimization algorithms were implemented in the **ROL** library part of the **Trilinos** framework. These methods were applied to standalone demonstration problems and the **Drekar** multi-physics research application. Out of this exploration a new augmented Lagrangian approach to structural design problems was developed. We demonstrate that this approach has favorable mesh-independent performance. Both the final design and the algorithmic performance were independent of the size of the mesh. In addition, topology optimization formulations for the design of conducting networks were developed and demonstrated. Of note, this formulation was used to develop a design for the inner magnetically insulated transmission line on the Z-machine. The resulting electromagnetic device is compared with theoretically postulated designs.

Contents

Appendix

List of Figures

Chapter 1

Introduction

This LDRD pursued the development of topology optimization methods motivated by the challenge of reducing current loss incurred in the convolute structure of the Z-machine. The goal of topology optimization is to optimally place material in a volume, subject to engineering constraints, that delivers optimal performance measured by a specific engineering quantity of interest. The optimization process defines the "topology" of the device being designed. This report details the approaches taken to address this challenge and their results.

This project was composed of three interacting research directions needed to advance topology optimization methodology for electromagnetic problems. The first is development of an optimization formulation that results in a physically realizable topology satisfying the engineering ideal and constraints. To ensure efficient and scalable topology optimization methods, the second direction is to develop nonlinear programming algorithms to solve large-scale PDE-constrained optimization problems relevant to topology optimization. Finally, PDE constrained optimization has substantial software overhead. This project worked to develop and demonstrate software approaches that can be applied to these complex problems.

1.1 Motivating Problem: Design of the Convolute on the Z-machine

The Z-machine is an electromagnetic device that uses large currents to create radiation environments. Figure 1.1 shows a profile of the Z-machine with the inset on the right focusing on the magnetically insulated transmission lines (MITLs) and convolute. A high-level description of how the Z-machine works can be found in [24]. Briefly, the Z-machine is arranged as a ring of capacitors whose power is simultaneously released. This current travels down the MITLs, towards the convolute where the current is rearranged before it reaches the target known as the *load*. Along the way in the convolute, owing to the high currents, strong electric fields, complex surface interactions, and material properties a percentage of the input power is lost [15, 16]. Design of the topology associated with the convolute, under ideal (e.g. lossless) conditions, is the ambitious goal of this LDRD.

The convolute is a stainless steel device that combines power transferred over several MITLs to an inner MITL eventually delivering the current to the Z-machine target [28]. The configuration of

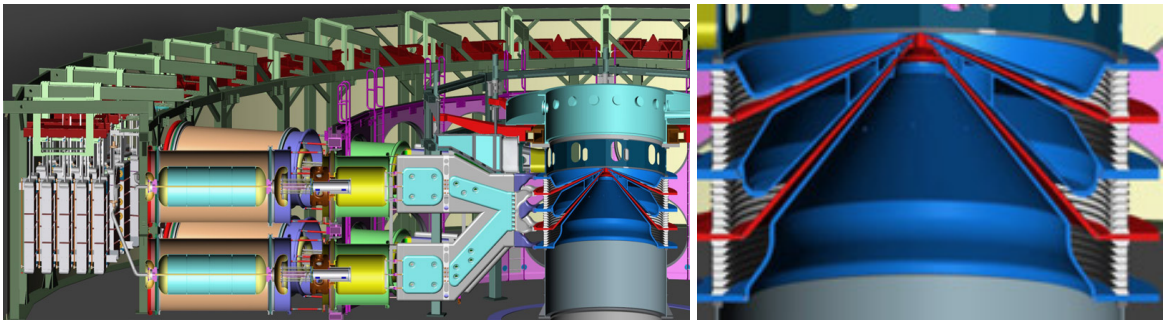

Figure 1.1. On the left is a profile of the Z-machine showing each component. The image on the right, is a close up of the region on the far right of the profile. This shows the Magnetically Insulated Transmission Lines (MITLs) and at its center is the convolute. Further details on how the Z-machine works can be found in [24], the image and further discussion can also be found in [26].

the incoming MITLs is such that the convolute device is fundamentally 3D. Current Sandia designs are based on a "double post hole" topology. Model development to gain insight into current loss is ongoing, but plasma models in the presence of large electromagnetic fields will play a significant role in the loss mechanism. To focus on the required optimization algorithms and formulations this project is focused on design optimization using electromagnetic models.

The project targeted simplified models of the convolute and MITLs. In particular, the MITLs can be viewed as a coaxial cable (concentric cylindrical conductors with a gap between carrying a signal). One extension of the straight coaxial cable model is a transition from a wide radius input port to a narrow radius output port. On the Z-machine this is how the inner MITL that transfer current from the convoluted to the load is structured. There has been some theoretical working designing an idealized inner MITL [19, 38]. These results suggest a dome shape design is more efficient than a cone shape.

1.2 Topology Optimization Methods and Formulations

Topology optimization [35, 1] is a PDE-constrained optimization computational technique for designing an engineering system optimized for a quantity of interest (e.g energy) subject to a constraint (e.g. a prescribed mass). This technique was developed originally to design optimal structures for carrying a specified load. To that end, most of the work on topology optimization has been targeting the linear elasticity equations as a PDE constraint (though more realistic nonlinear structural models have since been pursued). There are a number of different approaches that are used to define the distribution of material over a domain. The distribution of material in this domain is constrained only by explicitly enforced engineering constraints. This can lead to very general

topologies. This is in contrast to shape optimization approaches where a domain is parameterized and optimization is performed over those parameters. The extent of the topology achieved is thus limited by the parameterization. The ability of topology optimization to find designs beyond what is dictated by engineering intuition is one of its strengths.

There are a number of approaches used to define a topology. All of them use a continuous field defined over the domain. This field indicates if there is a material or a void at a particular location. The PDE model (e.g. linear elasticity) is robust to the presence of both material and void. A special treatment of the void may be required. A good review of the options can be found in [35].

Our approach is to use an artificial density field that ranges between 0 (void space) and 1 (material space). The interface between void and material represents the shape of the component being designed. This approach has been successful for problems in structural design [35] and the design of antennas [22] (among others). The optimization problem is initialized with a uniform artificial density. As the optimization algorithm proceeds the objective function and constraints drive the artificial density field to evolve until the shape materializes. This approach has no topological assumptions imposed by the formulation, thus obtaining the complex topologies is possible. To solve the resulting problems nonlinear programming methods are applied. We utilized and extended the Rapid Optimization Library (**ROL**) as the algorithmic tool for solving the optimization problems.

1.3 Summary of Contributions

There were several concrete artifacts produced as a result of this project. This includes two papers on block preconditioning, see references [10], and [11]. In addition to this report, one additional report (see [12]) is progressing towards publication. Five presentations were given at conferences:

1. *G. von Winckel*, E.C. Cyr, T. Gardiner, D.P. Kouri, D. Ridzal, and J.N. Shadid, Reduced and Full Space Methods in Topology Optimization: Applications in Linear Elasticity and Electrical Conduction, Copper Mountain Conference on Iterative Methods, 2016.

2. *E.C. Cyr*, G. von Winckel, T. Gardiner, D.P. Kouri, S. Miller and J.N. Shadid, Topology optimization for design of coaxial cables, SIAM Conference on Computational Science and Engineering, 2017.

3. *G. von Winckel*, E.C. Cyr, and D.P. Kouri, Algorithms for the topology optimization of electrical conductors, SIAM Conference on Computational Science and Engineering, 2017.

4. *E.C. Cyr*, G. von Winckel, T. Gardiner, D. Kouri, S. Miller, D. Ridzal and J.N. Shadid, Topology Optimization: Challenges, Algorithms and Applications, UNM, Mathematics Colloquium and the Applied Mathematics Seminar, 2017.

5. *E.C. Cyr*, G. von Winckel, T. Gardiner, D. Kouri, S. Miller and J.N. Shadid, Methods for Topological Design of Conducting Networks, Coupled Problems, 2017.

Also, a mini-symposium titled *Topology Optimization: Formulations, Methods and Applications* was organized by the PI and Boyan Lazarov (Technical University of Denmark) at the 2017 SIAM Conference on Computational Science and Engineering. Two student visitors were mentored and supported by this project for a summer. They each produced student research papers published in the CSRI proceedings

- Mona Hajghassem, University of Maryland Baltimore County, summer 2015, reference [27],
- Nick Fisher, Colorado School of Mines, summer 2016, reference [30].

In addition several follow on proposals were funded. The CIS LDRD project

- *Distributed-in-time techniques for optimization at extreme scales (DITTO-X)*, PI: Denis Ridzal, 2018-2020

is a collaboration between Ridzal and the PI of the present project that evolved directly out of the student work [27]. The PI of the present project is a principal on the grand challenge LDRD

- *Towards Predictive Plasma Science and Engineering through Revolutionary Multi-Scale Algorithms and Models*, PI: George Laity, 2018-2020.

The inclusion of the current PI on the GCLDRD, is partially a result of the work towards understanding the Z-machine power flow required for this project. The efforts for electromagnetic simulation and optimization developed in this project are planned to be leveraged in the *DARPA-EXTREME (Enabling Extreme New Designs for Optics and Imagers)*, PI Ihab El-kady. A number of ASC projects are also using **ROL** to do their optimization, including Sierra Dynamics, Xyce and Plato.

Much of the labor in this project went to development of sophisticated software tools. The work towards these tools was in the development of libraries and approaches that will be broadly applicable beyond the scope of the present work. A summary of the library development includes development in the **Trilinos** packages **ROL** and **Panzer** . This includes directed usage of the **Sacado** AD library to compute derivatives required for optimization. Finally, the application **Drekar** was modified to include optimization. This includes novel interfaces that could be generalized for usage in other applications. For a more detailed description of these contributions see Chapter 2.

Algorithmically this project considered Hessian based optimization algorithms applied to topology optimization. This includes augmented Lagrangian formulations, interior point methods and the Kelley-Sachs trust region approach. This effort has exposed a fundamental challenge of developing scalable (e.g. mesh independent) algorithms for topology optimization. Namely the non-convexity of the objective imposed by the $0/1$ nature of the density approach. Despite these challenges, progress was made using these algorithms to designing conducting networks using both electrostatic and electromagnetic PDE constraints.

Chapter 2

Software

The development of the optimization formulation is a substantial problem for topology optimization and PDE constrained optimization. A number of important considerations go into defining the objective, PDE formulation and constraints. These considerations include, but are not limited to, physical design goals, design constraints, PDE solution regularity, structure of the objective, and achievable performance of the optimization algorithms. As such, in an exploratory phase of problem formulation (as opposed to a more production capability) the flexibility of the software is essential to rapidly prototyping many problem formulations. This section discusses several of the core software technologies developed as part of this LDRD effort.

2.1 Automatic Differentiation (AD)

The tool used for automatic differentiation is a C++ library called **Sacado** distributed as part of the **Trilinos** framework. The approach taken uses operator overloading and template metaprogramming to automatically compute derivatives efficiently. The application of this technology, combined with other software tools, to implement a flexible finite element application is discussed in [32, 33]. **Sacado** can be efficiently applied when the derivatives are dense, and has been extended to work efficiently on various multicore platforms (Intel Phi, NVIDIA GPUs, threaded CPUs) [39]. This section will present an example **Sacado** program that computes the first and second derivatives of a function, these will be used in computing gradients and Hessians required in PDE-constrained optimization.

The following code listing is based on an example from the **Trilinos** package **Panzer** . The original example can be found in the **Trilinos** distribution[1], and is built frequently with the **Trilinos** continuous integration process.

```
1  typedef Sacado::Fad::DFad<double> FadType;
2  typedef Sacado::Fad::DFad<Sacado::Fad::SFad<double,1> > SecondFadType;
3
4  template <typename Scalar>
5  Scalar func(const Scalar & x,const Scalar & y)
6  {
7    return std::sin(x*y)+0.25*std::cos(y);
8  }
```

[1] See `git` SHA1 d3b096f4f1a984291442564a93f03592dad859b1 in the **Trilinos** master branch. The example file given a checkout of **Trilinos** is: `Trilinos/packages/panzer/disc-fe/example/SacadoExample/example.cpp`.

```
9
10 inline SecondFadType seed_second_deriv(int num_vars,int index,double xi,double vi)
11 {
12   SecondFadType x = SecondFadType(num_vars,index,xi);
13   x.val().fastAccessDx(0) = vi;
14
15   return x;
16 }
17
18 int main(int argc,char * argv[])
19 {
20   double x_val = 0.25, y_val = 0.5;
21   double dx = 2.0, dy = 3.0;
22
23   { // value
24     double x = x_val;
25     double y = y_val;
26     double f = func(x,y);
27
28     std::cout << "Value:  " << f << std::endl;
29   }
30
31   { // first derivative
32     FadType x = FadType(2,0,x_val);
33     FadType y = FadType(2,1,y_val);
34     FadType f = func(x,y);
35
36     std::cout << "First:  " << f << std::endl;
37   }
38
39   { // second derivative
40     SecondFadType x = seed_second_deriv(2,0,x_val,dx);
41     SecondFadType y = seed_second_deriv(2,1,y_val,dy);
42     SecondFadType f = func(x,y);
43
44     std::cout << "Second: " << f << std::endl;
45   }
46
47   return 0;
48 }
```

This listing computes the value (code block at line 23), derivative (code block at line 31), and action of the Hessian (code block at line 39) for the function defined on line 5. The function

$$f(x,y) = \sin(xy) + \frac{1}{4}\cos(y) \tag{2.1}$$

has two dependent variables x, y and is a nonlinear function of each variable.

The first derivative **Sacado** type is defined on lines 1. The "Fad" (Forward AD) class is templated on the value type, in this case `double`. When the `FadType` is used to evaluate the templated function, the expression template mechanism automatically computes the derivatives at the same time as computing the value. The key step is to initialize or *seed* the variables correctly. This is done in the code block starting at line 31 where the variable `x` is defined by calling the `FadType` constructor. This constructor takes three arguments `FadType(num_vars,var_index,var_value)` where the arguments are

- *num_vars*: the number of variables to differentiate with respect to,
- *var_index*: the index of this variable in the array of variables to differentiate with respect to,

- *var_value*: the value of this variable.

For example when computing the gradient with respect to the x and y variables (code block at 31) there are 2 variables. Thus the first argument to the `FadType` constructor is 2. The x variable is defined as the first index and has value `x_val`, thus the constructor call instantiating `x` is `FadType(2,0,x_val)`. Similarly for the y variable, this time its the second index to differentiate with respect to and has value `y_val` yielding the constructor call `FadType(2,1,y_val)`.

The second derivative type is defined on line 2. Note that this templated type nests, two first derivative types. In this case the gradient is computed in the outer "Fad" object and the second derivative is computed by the inner type. In this case the template argument of `1` indicates that this derivative will only be computed with respect to one variable. In fact, the second derivative computed by this type will be a directional derivative. Define $\vec{x} = [x, y]$ and a direction $\vec{v} = [v_x, v_y]$, the direction second derivative is defined as:

$$H(\vec{x})\vec{v} = \frac{\partial}{\partial \tau}\Big(\nabla f(\vec{x} + \tau\vec{v})\Big)\Big|_{\tau=0}. \tag{2.2}$$

Thus in this case, the first derivative is similar to what is computed for the gradient, but the second derivative is actually taken with respect to the τ variable. Again, the key is seeding the variables passed into the templated function. The seeding of the second derivative variables is done by the function defined starting on line 10. The first line of this function seeds the first derivative. This is identical to the line used for initialization of the first derivative AD type. The second line in the function seeds the second derivative. This line explicitly sets the derivative of $\vec{x}$ with respect to τ to be $\vec{v}$. This is used to build up the chain rule computations used by the forward mode automatic differentiation algorithm.

The description of using **Sacado** AD to compute derivatives of a scalar function is easily extended to a vector with many parameters. This is used to compute local element stiffness matrices and Hessian contributions for use in a finite element method. A key point is that these derivatives are typically dense (or can be computed densely). With local element residuals, stiffness and sensitivity values computed a scatter routine is required to distribute these contributions to global vector and sparse matrix data structures. A similar gather routine is used to transfer data from global data to local data. In this step appropriate seeding of the **Sacado** objects is done to ensure proper propagation of the derivatives. See for instance the discussion in [32, 33]

2.2 Objective Calculator

2.2.1 First and Second Derivatives

The form of the objective function is often the subject of formulation exploration. In this case, we assume that the PDE code can compute a set of G "responses" $g_i(u,z)_{i=1^G}$. Each of which maps the state and control to a real number. In general, the objective function can then be written as

$$F(u,z) = f(g_1(u,z), g_2(u,z), \ldots g_N(u,z)). \tag{2.3}$$

Using the automatic differentiation procedure described above, we assume each of the response quantities has first and second derivatives with respect to u and z. To use this objective in second order optimization requires the first and second derivative operators:

$$\begin{bmatrix} \frac{\partial F}{\partial u} \\ \frac{\partial F}{\partial z} \end{bmatrix}, \quad \begin{bmatrix} \frac{\partial^2 F}{\partial u^2} & \frac{\partial^2 F}{\partial u \partial z} \\ \frac{\partial^2 F}{\partial z \partial u} & \frac{\partial^2 F}{\partial z^2} \end{bmatrix}. \tag{2.4}$$

Often, only the action of the second derivative is required. To compute these derivatives, we have developed a run time calculator capability in **Drekar** using embedded automatic differentiation. The input to this capability is managed by the `Teuchos::ParameterList`.

The user of this capability defines the objective function $f : \mathbb{R}^N \to \mathbb{R}$, where each argument will receive a response value. Using the chain rule we define the first derivatives of the objective function F as:

$$\frac{\partial F}{\partial u} = \sum_{i=1}^{N} \frac{\partial f}{\partial g_i} \frac{\partial g_i}{\partial u}, \quad \frac{\partial F}{\partial z} = \sum_{i=1}^{N} \frac{\partial f}{\partial g_i} \frac{\partial g_i}{\partial z}. \tag{2.5}$$

To implement first derivatives, `Sacado` based derivatives are used to compute the first derivative terms $\frac{\partial f}{\partial g_i}$ for $i = 1 \ldots N$. Notice that since a dense derivative array of length N is computed, `Sacado` can be used with reasonable efficiency. To compute the full derivative, a linear combination of the application computed derivatives $\frac{\partial g_i}{\partial u}$ is calculated.

A similar approach can be taken for the second derivatives. Though because in general for an objective, the second derivative is a matrix, the ROL library requires users to compute the action of the Hessian on a vector. While not difficult, this has to be carefully accounted for in the computation of the derivatives. Fortunately, its easy to use the `Sacado` library to compute the required direction derivatives. Before proceeding with the full expression of the second derivative consider the case where the objective is composed of a single response $g(u,z)$, e.g. $F(u,z) = f(g(u,z))$. In the terse notation below, the use of brackets indicates the directional derivative applied at the point (u,z). Now using calculus of variations the directional derivative for one mixed derivative is computed

$$\begin{aligned} \partial^2_{uz} F[v_z] &= \partial_\tau \big(\partial_u F(u, z + \tau v_z) \big|_{\tau=0} \\ &= \partial_\tau \big(\partial_g f(g(u, z + \tau v_z)) \partial_z g(u, z + \tau v_z) \big|_{\tau=0} \\ &= \partial^2_g f \partial_u q[v_z] \partial_u q + \partial_g f \partial^2_{uz} g[v_z]. \end{aligned} \tag{2.6}$$

When discretized, the derivatives $\partial_u q$, $\partial^2_{uz} g[v_z]$ and $\partial^2_{uz} F[v_z]$ are all vectors of dimension u. This

result is generalized for objectives in the form of Eq. 2.3:

$$
\begin{aligned}
\frac{\partial^2 F}{\partial u^2}[v_u] &= \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{\partial^2 f}{\partial g_i \partial g_j}\frac{\partial g_j}{\partial u}[v_u]\right)\frac{\partial g_i}{\partial u} + \frac{\partial f}{\partial g_i}\frac{\partial^2 g_i}{\partial u^2}[v_u], \\
\frac{\partial^2 F}{\partial u \partial z}[v_z] &= \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{\partial^2 f}{\partial g_i \partial g_j}\frac{\partial g_j}{\partial z}[v_z]\right)\frac{\partial g_i}{\partial u} + \frac{\partial f}{\partial g_i}\frac{\partial^2 g_i}{\partial u \partial z}[v_z], \\
\frac{\partial^2 F}{\partial z \partial u}[v_u] &= \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{\partial^2 f}{\partial g_i \partial g_j}\frac{\partial g_j}{\partial u}[v_u]\right)\frac{\partial g_i}{\partial z} + \frac{\partial f}{\partial g_i}\frac{\partial^2 g_i}{\partial z \partial u}[v_u], \\
\frac{\partial^2 F}{\partial z^2}[v_z] &= \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{\partial^2 f}{\partial g_i \partial g_j}\frac{\partial g_j}{\partial z}[v_z]\right)\frac{\partial g_i}{\partial z} + \frac{\partial f}{\partial g_i}\frac{\partial^2 g_i}{\partial z^2}[v_z].
\end{aligned}
\tag{2.7}
$$

In a similar way as the first derivative calculations, the derivatives of the objective function and the responses can be computed efficiently with `Sacado`.

2.2.2 Expression Tree Calculator

To build up an objective function at run time a *factory pattern* [13] is used to build the data structures from a`Teuchos::ParameterList` input. The approach to the calculator is motivated by the generic form of the objective function defined in Eq. (2.3). There are several responses labeled g_i that are automatically included in the calculator. These responses are leaves in an expression tree. The tree is built up from *foundational* terms to build a complex expression. For instance the expression

$$
F(g_0, g_1, g_2) = \frac{1}{2}(g_0 - g_1)^2 + \frac{1}{3}\max(g_2, 0)^3 \tag{2.8}
$$

has two *Sum* terms (e.g. $g_0 - g_1$), a *Power* term (e.g. $(\cdot)^2$), and a Moreau-Yosida-type penalty term (e.g. $\frac{1}{3}\max(g_2,0)^3$). Each term takes as input other terms in the tree. The XML input to build this objective function is:

```
<!-- Computes Objective = DiffSquared + MY-Term -->
<ParameterList name="Objective">
  <Parameter name="Type" type="string" value="Sum"/>
  <Parameter name="Terms" type="string" value="DiffSquared,MY-Term"/>
  <Parameter name="Scaling" type="string" value="1.0,1.0"/>
</ParameterList>

<!-- Computes DiffSquared = Difference^2 -->
<ParameterList name="DiffSquared">
  <Parameter name="Type" type="string" value="Power"/>
  <Parameter name="Term" type="string" value="Difference"/>
  <Parameter name="Power" type="double" value="2.0"/>
  <Parameter name="Scaling" type="double" value="0.5"/>
</ParameterList>

<!-- Computes MY-Term =  \frac{1}{3}\max{(g_2,0)}^3 -->
<ParameterList name="MY-Term">
```

```
    <Parameter name="Type" type="string" value="Moreau-Yosida"/>
    <Parameter name="Term" type="string" value="g_2"/>
  </ParameterList>

  <!-- Computes Difference = g_0-g_1 -->
  <ParameterList name="Difference">
    <Parameter name="Type" type="string" value="Sum"/>
    <Parameter name="Terms" type="string" value="g_0,g_1"/>
    <Parameter name="Scaling" type="string" value="1.0,-1.0"/>
  </ParameterList>
```

Embedded in this XML input is the expression tree used in computing the objective. For instance the *Sum* that comprises the objective term depends on the strings `"DiffSquared"` and `"MY-Term"` whose values are defined further below in the XML input. The dependence on other terms is terminated when a leaf response g_i is found.

Given the input described above, Fig. 2.1 shows the class diagram designed for the calculation of an objective function. The `ObjectiveFunctionFactory`, has two primary methods. One takes the users `Teuchos::ParameterList` that defines the objective function, and terms used to construct it. The second method, `build`, builds a term in the objective function by name. The choices for the string are defined by the names of the individual terms in the parameter list (e.g. `"Objective"`, `"MY-Term"`, or `"Difference"` from the sample input above), or the leaf responses. Notice that this method is also templated on the scalar type. This reveals the templating mechanism that embeds the automatic differentiation into the construction of the objective function. This method returns the second primary class from the UML diagram, an `ObjectiveFunctionBase` object. This object takes a `Teuchos::ParameterList` that describes the sublist for the object. This would correspond to the sublist named `"Difference"` in the example above. With the parameter list set the object can be `initialized` with an `ObjectiveFunctionFactory` object. The initialization phase is where the term being initialized builds any other term, excluding leaves, it may need. The final method, `evaluate`, takes as an argument a map between the name of the leaf terms and their value. The method returns a scalar type with the result of the evaluation.

For first and second automatic derivative evaluation types, the `evaluate` method returns the derivatives with respect to the leaves:

$$\frac{\partial F}{\partial g_i}, \frac{\partial^2 F}{\partial g_i \partial g_j} \text{ for } i,j = 1 \ldots N. \tag{2.9}$$

Given these values, its easy to compute the full derivative with respect to the state or control variables using the precomputed derivatives $\partial g_i/\partial u$, $\partial g_i/\partial z$, $\partial^2 g_i/\partial u \partial z$, $\partial^2 g_i/\partial z \partial u$, $\partial^2 g_i/\partial u^2$, and $\partial^2 g_i/\partial z^2$ and Eqs. (2.5), (2.7). Separating the responses from the objective function limits the extent to which the application code is responsible for building up an objective function, while still automating the evaluation of objective function values and derivatives.

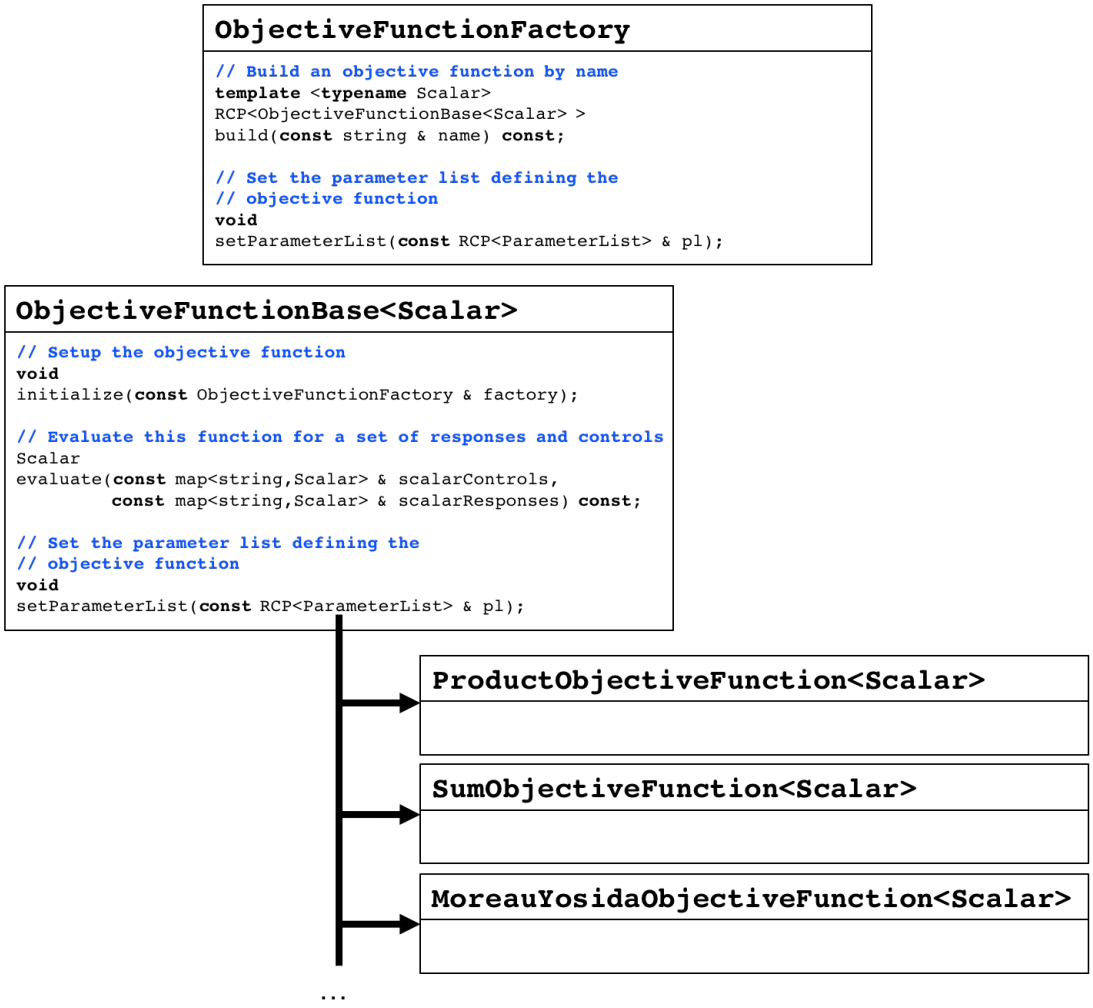

Figure 2.1. A UML diagram of the class hierarchy required for the objective function calculator. The `ObjectiveFunctionFactory` class provides a uniform interface for constructing objective functions. Then using composition a complex calculation of the objective function can be constructed.

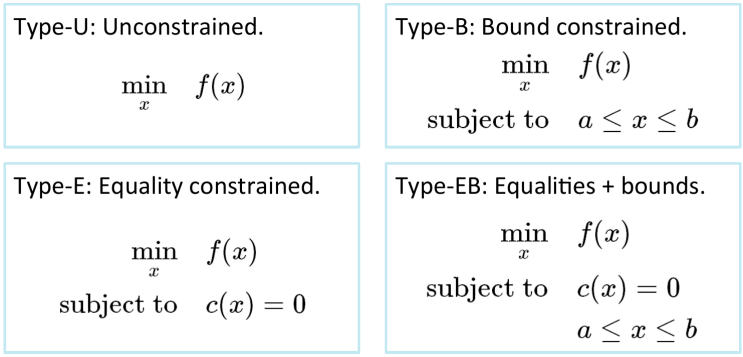

Figure 2.2. The four fundamental optimization problem types that can be solved with ROL

2.3 Rapid Optimization Library (ROL)

The Rapid Optimization Library (ROL) is a modular C++ package for large-scale numerical optimization. This project has provided substantial support for ROL's development. ROL is unique in the field of optimization software in that it provides an abstract interface for the constituent elements of an optimization problem: vectors, objective functions, equality constraints, *etc.* which are then overloaded via virtual inheritence at the application level to encapsulate the specific physics and data structures needed by the problem. For example, ROL provides base classes `ROL::Vector` and `ROL::Objective` which satisfy the function signatures of its algorithm components and the application developer must implement concrete `AppVector` and `AppObjective` that inherit from these classes. ROL solves four basic types of optimization problems, which are unconstrained (U), bound constrainted (B), equality constrained (E), and equality-bound constrained (EB), which can be written in the following forms:

More generally, problems may also have inequality constraints $c_i(x) \leq 0$, however, introduction of slack variables transforms the problem to Type-EB. Direct solution of a problem of Type-EB in which the equality and bound constraints are exactly enforced is computationally prohibitive at the scales needed for PDE-constrained optimization as the lack of convexity makes the problem NP hard. To circumvent this, ROL uses step classes which either incoporate the bound constraints into the objective or the equality constraint by means of a penalty term. These approaches are implemented in `ROL::MoreauYosidaStep` and `ROL::AugmentedLagrangianStep`, respectively.

In the early stages of ROL development, the interface design was limited to a specific subset of vector operations such as linearity that were motivated by the mathematics of vector space operations. Additionally, the algorithms are agnostic to the indexing of vector elements. That is to say that the solver components of ROL do not assume that direct access to the kth element of a given vector is possible. Bound constraints of the form $\ell_i \leq x_i \leq u_i$ can not be imposed using only linear operations on vectors and prior to this project, ROL provided an abstract interface class `ROL::BoundConstraint` and an example implementation when the application vector class con-

tains an `std::vector` object. To solve optimization problems with other concrete vector types required inheriting corresponding classes from `ROL::BoundConstraint`. This project supported the development of a more general and abstract way of formulating entities such as bound constraints in ROL. By adding the capability to evaluate nonlinear elementwise functions on vectors, it became possible to provide a bound constraint interface which was independent of the concrete vector class.

In the context of PDE-constrained optimization problems, the PDE constraint can be written in general as $c(u,z) = 0$ where u dentotes a *simulation* variable which is the solution to the PDE and z is an *optimization* variable to be determined such that a figure of merit involving $u \in \mathscr{U}$ and $z \in \mathscr{Z}$ is maximized or minimized. ROL provides an additional layer for vectors, objectives, and constraints collectively referred to as the SimOpt interface. Neglecting, for the time being, the additional complexities introducted by bound or inequality constraints, the PDE-constrained optimization problem can be written in the full-space representation as

$$\min_{u,z} f(u,z) \quad \text{subject to} \quad c(u,z) = 0 \tag{2.10}$$

In many physical applications, the constraint condition $c(u,z) = 0$ has certain favorable properties such as posessing an unique solution u^* for a given z^*. With appropriate boundary conditions and restrictions on z, this is always the case when the constraint is linear in u as it is for linear elasticity and Maxwell's equations. Whenever the constraint has a solution and its Jacobian with respect to u is nonsingular, the constraint is exactly satisfied and the simulation variable u can be eliminated by recognizing that it is an implicit function of z. This leads to the reduced space problem in terms of the optimization variable z alone.

$$\min_{z} f(u(z),z) \tag{2.11}$$

ROL provides the step classes `ROL::LineSearchStep` and `ROL::TrustRegionStep` for solving problems in the reduced space setting and `ROL::CompositeStep` for the full space setting, which is a sequential quadratic programming solver that simultaneously computes the pair (u,z). There are a few important differences between the full space and reduced space optimization problems. In the reduced space setting, the dimensionality of the solution is typically greatly reduced, for example, u might represent electric and magnetic fields that are time-dependent and z could be some time-independent scalar material property. In the full space setting, it would be necessary to store entire space-time, vector-valued functions evaluated on a finite element grid. The flip side of this expense is that it is readily conceivable that the full-space problem could have better scaling in the sense of less distended level sets of the objective. Additionally, the PDE constraint is not required to be exactly satisfied during the sequence of computing iterates; it is penalized in such a way that the PDE residual is reduced to a prescribed tolerance as part of a stopping criterion. The ROL SimOpt interface allows the application developer to switch between full-space and reduced-space solution methods by modifying a parameter input.

The bound constraints can be directly imposed on reduced space problems. ROL can solve

bound-constrained (Type-B) problems via linesearch using, for example, projected gradient or Newton methods that exactly enforce the bounds. Additionally, ROL has provided a trust region solver that incorporates bound constraints into a local model, which will be referred to as the Kelley-Sachs model [21], of the objective using projection methods in conjunction with Truncated CG, (double) dogleg, and Cauchy point methods. When the Kelley-Sachs model was applied to solving some electrical conduction topology optimization problems in the reduced space setting, very slow convergence was observed. At the time, it was posited that the concavity and disparate scaling of the Solid Isotropic Material with Penalization (SIMP, see Chapter 3 for more discussion) objective may have been impeding progress of iterates to the domain boundary, where the optimal binary solution is sought.

In much of the existing literature on the solution of topology optimization problems, the authors employ a sequential approximation method called the Method of Moving Asymptotes (MMA) or some variation on it. The reason for this is that even topology optimization problems involving linear PDEs tend to be non-convex and exhibit highly nonuniform gradient scaling, particularly in the case of using the SIMP method. MMA construct a sequence of convex model objective functions which locally match the Taylor series of the original objective to low order and implicitly enforce the bound constraints. The MMA objective can be written in the form

$$\tilde{f}^{(k)}(x^k) + \sum_{j=1}^{n} \frac{p_j(k)}{u_j - x_j} + \frac{q_j^{(k)}}{x_j - \ell_j} \tag{2.12}$$

where the numerators under the summation are

$$p_j^{(k)} = (u_j - x_j^{(k)})^2 \max\left\{0, \frac{\partial f}{\partial x_j}(x^{(k)})\right\}, \quad q_j^{(k)} = (x_j^{(k)} - \ell_j)^2 \max\left\{0, -\frac{\partial f}{\partial x_j}(x^k)\right\} \tag{2.13}$$

There are a few significant reasons that MMA was not pursued as a solution strategy for topology optimization problems incolving electromagnetics: MMA yields mesh-dependent solutions, computational cost scales poorly with problem degrees of freedom, and anecdotal evidence from review articles [34, 23] indicated that more standard optimization methods such as the primal-dual interior point methods could yield better performance. At the outset, it was not yet possible to implement MMA in ROL in a generic fashion as MMA requires nonlinear operations on vectors. While the constraint Jacobians could be represented as a vector of vectors, elementwise operations on these are also required. It is also unclear how this method could be generalized to a function space setting when there are primal and dual vectors which are elements of different function spaces and Riesz mappings between spaces.

2.3.1 Trust Region Models

It became clear that ROL would require an augmented set of vector operations to implement an interior point solver, generalize bound constraint handling to arbitrary types, and impose inequality constraints such as volume limits on structural designs. To address these needs, elementwise

operations were added to ROL in the form of unary, binary, and reduce vector methods. These methods do not require access to specific elements, *per se*. ROL's algorithms do not ever need to know how to read or write the ith element of a vector, as doing this in a general way could lead to prohibitively strict assumptions about the application data structures and how they may be manipulated, for example, on a distributed computing platform. The three types of elementwise operations can be represented as

$$\text{Unary: } x_i \leftarrow f(x_i), \quad \text{Binary: } x_i \leftarrow f(x_i, y_i), \quad \text{Reduce: } r = \text{reduce}(r, x_i)\ \forall i \tag{2.14}$$

While the above expressions involve elements of vectors, ROL does not invoke them directly, but rather it is up to the application code to provide ROL with the resulting quantities.

Another consequence of being able to evaluate arbitrary functions of vector elements was that we were able to incorporate an alternative trust region model, originally proposed by Coleman and Li[8], which has in excess of two thousand citations in the literature for bound-constrained nonlinear programming problems. The Coleman-Li model had not been incorporated into ROL earlier as it was not possible to do so in type-invariant way and required leveraging the newly-implemented elementwise vector operations.

As a comparison, the Kelley-Sachs model uses projection methods to ensure that trust region steps remain in the intersection of the trust region hypersphere and the n-orthotope defined by the bound constraints. The Coleman-Li model, in contrast, applies a diagonal transform matrix to the trust region step vector such that the level sets of the model are hyperellipsoids. The transformation matrix, D is constructed according to the multiple case formula

$$D_{ii}(x) := \text{diag}(|v_i(x)|^{-1/2}), \quad v_i(x) = \begin{cases} u_i - x_i & g_i < 0,\ u_i \leq \infty \\ x_i - \ell_i & g_i \geq 0,\ \ell_i \geq -\infty \\ -1 & g_i < 0,\ u_i = \infty \\ 1 & g_i \geq 0,\ \ell_i = -\infty \end{cases} \tag{2.15}$$

where $l_i \leq x_i \leq u_i$ are the bounds on the ith optimization variable and g_i is the ith component of the gradient of the objective. This transformation D scales the step in such a way that it is always strictly interior and only asymptotically reaches the boundary. To benchmark the comparative performance of the Kelley-Sachs (KS) and Coleman-Li (CL) models, as well as the Moreau-Yosida and Augemented Lagrangian approaches to handling bound and (in)equality constraints in tandem, we implemented a test suite which solves the first 41 test problems introduced by Hock and Schittkowski[17] (HS#). The first five of the test problems are Type-B and amenable to ROL's trust region solver with either of the two implemented models. They also serve as visually instructive examples as we can plot the progress of an optimization algorithm in two-dimensional space. As far as we are aware, these two approaches to handling bound constraints in a trust region framework have never been compared.

In the context of PDE-constrained optimzation problems the predominant computational cost is that in satisfactorially solving the constraint PDEs themselves. While the cost of a PDE solve

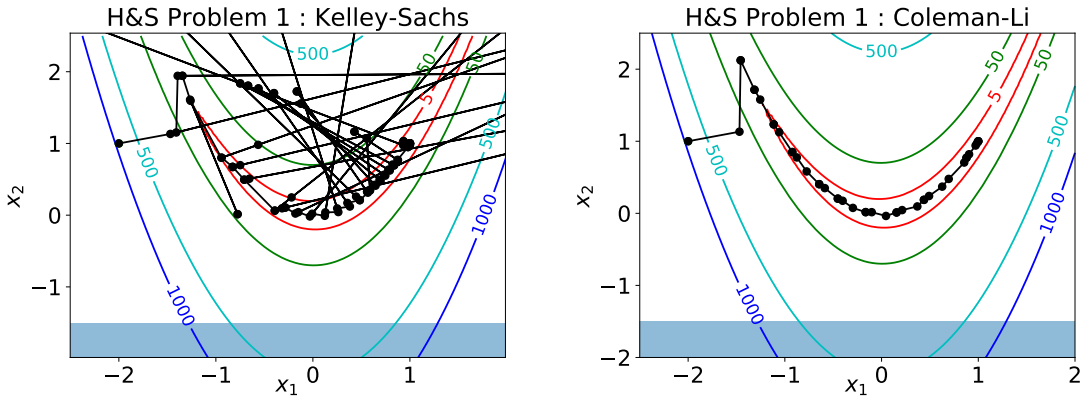

Figure 2.3. In HS1, both models succeed in finding the minimizer, however, CL converges with significantly fewer objective and gradient evaluations. The objective evaluation points are remarkably the same for both models in this example.

is independent of the optimization algorithm used, the number of solves needed to resolve a minimizer is aggregated from the number of objective function and gradient evaluations, as well as the generation of Krylov vectors for methods using Newton-Krylov iterations. For this reason, when we consider comparative performance of the two trust region models, it is important to consider not only the number of optimzation iterations needed, but also the underlying cost of those iterations

Problem HS1 (2.16) shows the CL model leading to convergence that is significantly superior to that of Kelley-Sachs

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \quad -1.5 \leq x_2, \quad x_0 = (-2, 1), \quad x^* = (1, 1) \tag{2.16}$$

The initial guess x_0 is feasible and the gradient can be badly scaled. The optimal solution x^* is strictly interior and the CL solve attains a satisfactory minimizer with 45 objective function (#fval) and 35 gradient (#grad) evaluations, whereas the KS solve takes #fval=123 and #grad=57. When the minimizer is strictly interior, but the trust region model predicts it to be outside the domain, it is plausible that the KS method could be inefficient. The locations of the iterates in the optimization space and the isocontours of the objective are shown in figure 2.3.

In contrast, HS2 is almost identical to HS1 with an infeasible starting guess of $x_0 = (-2, 1)$. KS reaches the minimizer in about 75% the iterations it needed for HS1, while CL fails to converge to the desired solution as it becomes trapped in a local minimizer. KS produces many iterates that are far from the minimizer because in this neighborhood, the objective can only be approximated by the first three terms in its Taylor series within a small radius.

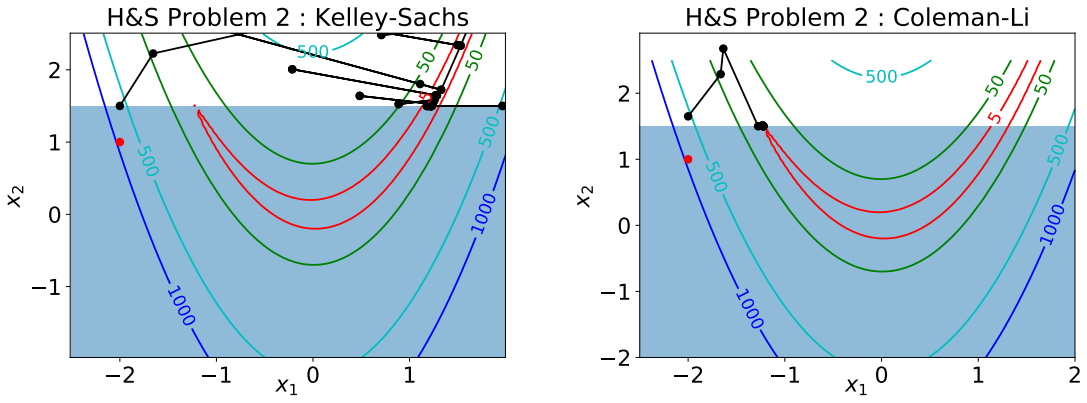

Figure 2.4. The CL model fails to converge due to an infeasible initial guess. KS makes numerous evaluations outside of the plot and far from the minimizer.

$$f(x) = x_2 + 10^{-5}(x_2 - x_1)^2, \quad 0 \le x_2 \tag{2.17}$$

HS3 (2.17) has a convex quadratic objective function, albeit with a Hessian that has highly dissimilar eigenvalues. The CL model excels in this situation as the diagonal transformation of the trust region model and step allows for two CG iterations, which span the entire space, and convergence is obtained within a single iteration. KS, in contrast, takes only one truncated CG iteration per optimization step because the predicted minimizer lies outside the trust region radius. Consequently, KS produces an iterate on the boundary which then proceeds along towards the minimizer2.5.

HS4 is essentially the converse of HS3, where CL is quite slow as it can only take a single CG iteration each step due to the model mimimizer lying outside of the trust region, whereas KS's model allows both CG iterations to be performed and the method converges in one step2.6.

CL has the better convergence in the last test problem 2.18 for similar reasons as HS3. Again, the CL model allows for two CG iterates while KS gets only one for most of the time. The approach of the minimizer is much more uniform as well and the problem is nearly solved within a couple of iterations.

$$f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1, \quad -1.5 \le x_1 \le 4, \quad -3 \le x_2 \le 3 \tag{2.18}$$

The comparative performance in these test problems of the CL and KS approaches to handling bound constraints with trust regions leads to a few observations about their potential merits. The CL method looks like the better choice for problems where the initial guess and the minimizer are both interior to the bounded set. This is, of course, not going to be the case with topology optimization problems where a solution is desired to be on the surface of the bounded set as part of the binary design.

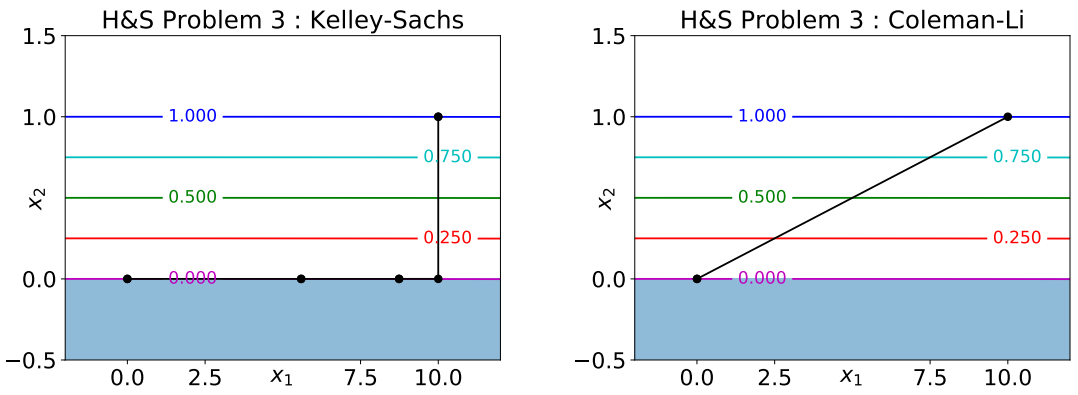

Figure 2.5. Results for a quadratic objective with highly elliptical level sets.

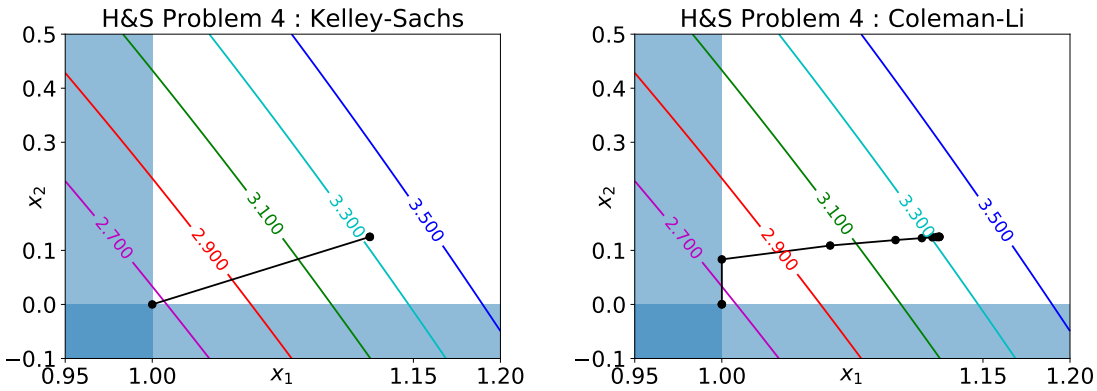

Figure 2.6. CL exhibits slower convergence than KS.

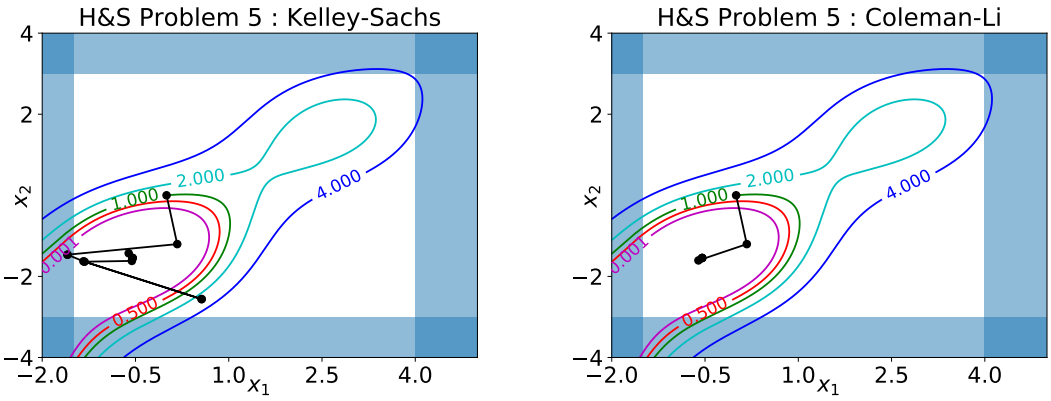

Figure 2.7. CL finds the neighborhood of the minimizer almost immediately, but then must take several small steps to satisfy the gradient norm requirements.

2.3.2 Constrained Optimization Algorithms

All topology optimization problems considered in this work have the abstract form

$$\min_{x\in\mathcal{X}} \; f(x) \tag{2.19a}$$

$$\text{subject to} \quad g(x) = 0 \tag{2.19b}$$

$$h_\ell \le h(x) \le h_u \tag{2.19c}$$

$$x_\ell \le x \le x_u. \tag{2.19d}$$

Upon first glance, optimization problem (2.19) does not fit into the problem types handled by ROL. However, by adding "slack" variables to (2.19c), we arrive at a Type-EB problem. That is, let s be a vector in the range of h, denoted by $\mathcal{H}$, then we can equivalently reformulate (2.19c) as

$$h(x) - s = 0 \quad \text{and} \quad h_\ell \le s \le h_u,$$

resulting in the Type-EB problem

$$\min_{x\in\mathcal{X},\, s\in\mathcal{H}} \; f(x) \tag{2.20a}$$

$$\text{subject to} \quad c(x,s) := \begin{pmatrix} g(x) \\ h(x) - s \end{pmatrix} = 0 \tag{2.20b}$$

$$x_\ell \le x \le x_u \tag{2.20c}$$

$$h_\ell \le s \le h_u. \tag{2.20d}$$

Under this project, we implemented this slack variable transformation in the new ROL class `ROL::OptimizationProblem`. In addition to transforming problems of type (2.19) to (2.20), `ROL::OptimizationProblem` also has general capabilities to check the input vectors and the

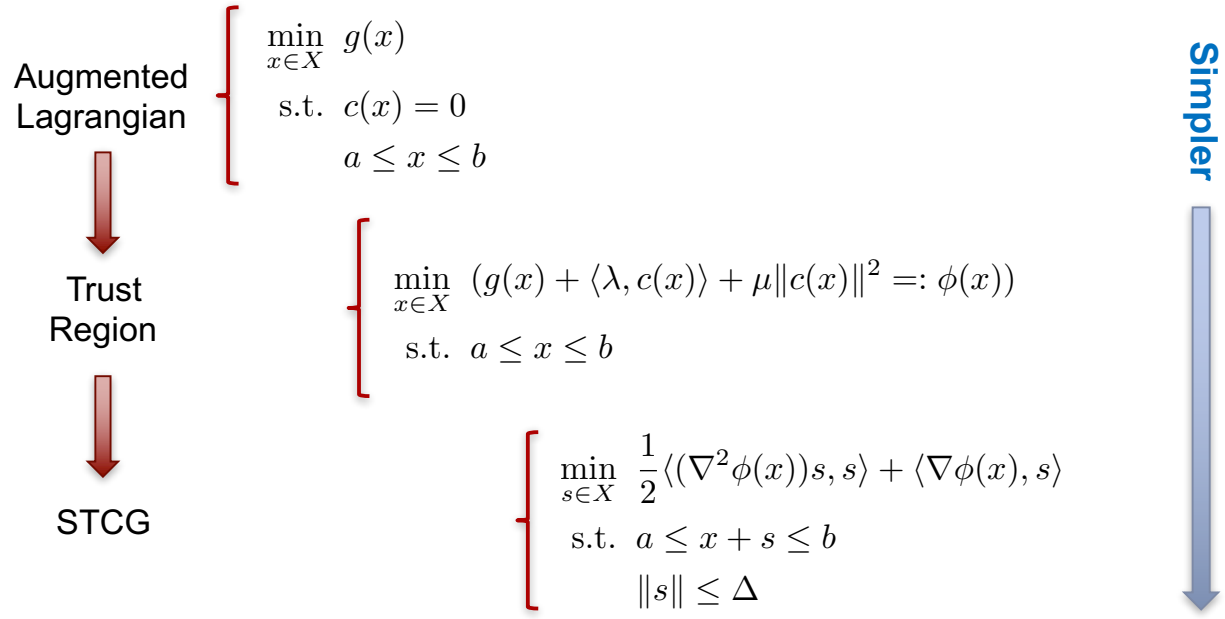

Figure 2.8. Some optimization algorithms are constructed by reducing an initially hard problem hierarchically into subproblems that can be addressed by existing algorithms. This figure demonstrates how a constrained optimization algorithm can be translated into an augmented Lagrangian formulation, utilizing a trust region solver, that in turn utilizes a Steihaug-Toint CG (STCG) algorithm to solve the quadratic programming problem.

derivatives of the input functions using finite differences. This capability to perform a derivative check is a critical tool in ensuring correct implementation of (2.20). To better streamline ROL's interface, we additionally implemented the new class `ROL::OptimizationSolver`. `ROL::OptimizationSolver` constructs a `ROL::Algorithm` based on user input or constructs the default for the problem type defined by `ROL::OptimizationProblem`. Additionally, `ROL::OptimizationSolver` provides the capability for the user to provide their own `ROL::StatusTest` for terminating the optimization.

To solve problems of type (2.20), we have developed three primary capabilities: Augmented Lagrangian, Moreau-Yosida penalty and interior points. Each of these approaches penalizes different constraints in (2.20) and solves subproblems of Type-E or Type-B. We depict this penalization and reduction to simpler subproblems in Figure 2.8 for the case of Augmented Lagrangian. Throughout the subsequent discussions, we assume the range of g is a reflexive Banach space, denoted $\mathscr{G}$, $\mathscr{H} = L^2(\Omega, \mathscr{B}, m)$ and $\mathscr{X} = L^2(\Xi, \mathscr{F}, n)$ where $(\Omega, \mathscr{B}, m)$ and $(\Xi, \mathscr{F}, n)$ are measure spaces. We denote the duality pairing on $\mathscr{G}$ by $\langle\cdot,\cdot\rangle_{\mathscr{G}^*,\mathscr{G}}$ where $\mathscr{G}^*$ denotes the topological dual space of $\mathscr{G}$. We further denote the inner products on $\mathscr{H}$ by

$$\langle s,t\rangle_{\mathscr{H}} = \int_\Omega s(\omega)t(\omega)\,\mathrm{d}m(\omega)$$

and similarly for $\mathscr{X}$.

Augmented Lagrangian. The Augmented Lagrangian algorithm [2, 9, 31] is popular due to its simplicity. Our implementation of Augmented Lagrangian first constructs the so-called agumented Lagrangian functional

$$L_r(x,s,\lambda,\mu) := f(x) + \langle \lambda, g(x)\rangle_{\mathscr{G}^*,\mathscr{G}} + \langle \mu, h(x)-s\rangle_{\mathscr{H}} + \frac{r}{2}\left(\|g(x)\|_{\mathscr{G}}^2 + \|h(x)-s\|_{\mathscr{H}}^2\right).$$

The traditional Augmented Lagrangian algorithm then minimizes $L_r(\cdot,\cdot,\lambda,\mu)$ for fixed $r>0$, $\lambda \in \mathscr{G}^*$ and $\mu \in \mathscr{H}$. That is, at iteration k, we have a current penalty parameter $r_k > 0$ and multiplier estimates λ_k and μ_k. Using this information, we approximately solve

$$\min_{x\in\mathscr{X}, s\in\mathscr{H}} \; L_{r_k}(x,s,\lambda_k,\mu_k) \tag{2.21a}$$

$$\text{subject to} \quad x_\ell \le x \le x_u \tag{2.21b}$$

$$h_\ell \le s \le h_u \tag{2.21c}$$

using ROL's Type-B capabilities. This produces the new iterates x_{k+1} and s_{k+1}, and using these values we update the multipliers using the steepest ascent steps

$$\lambda_{k+1} = \lambda_k + r_k R_g g(x_{k+1}) \quad \text{and} \quad \mu_{k+1} = \mu_k + r_k(h(x_{k+1}) - s_{k+1})$$

where R_g is the linear injection operator mapping the range of g into its topological dual space. Once the multipliers are updated, the penalty parameter r_k is either increased or decreased depending on constraint feasibility. The Augmented Lagrangian approach is implemented in ROL as `ROL::AugmentedLagrangianStep`.

Moreau-Yosida Penalty. The Moreau-Yosida penalty method is closely related to the Augmented Lagrangian method [20]. For this method, we first consider the partially augmented Lagrangian

$$L_r(x,s,\mu) := f(x) + \langle \mu, h(x)-s\rangle_{\mathscr{H}} + \frac{r}{2}\|h(x)-s\|_{\mathscr{H}}^2,$$

for fixed x, μ and r. Minimizing $L_r(x,\cdot,\mu)$ over $h_\ell \le s \le h_u$ results in the optimal slack variable

$$s^\star = \min\{h_u, \max\{h_\ell, r^{-1}\mu + h(x)\}\}$$

where the min and max operators are understood "pointwise" in Ω. Now, evaluating $L_r(x,\cdot,\mu)$ at $s^\star$ yields

$$\begin{aligned} L_r(x,s^\star,\mu) =& f(x) - \frac{1}{2r}\|\mu\|_{\mathscr{H}}^2 \\ &+ \frac{r}{2}\|\max\{0,\,(h_\ell - r^{-1}\mu) - h(x)\}\|_{\mathscr{H}}^2 + \frac{r}{2}\|\max\{0,\,h(x) - (h_u - r^{-1}\mu)\}\|_{\mathscr{H}}^2. \end{aligned}$$

We can produce the same penalization for the bound constraints $x_\ell \le x \le x_u$ resulting in the penalized objective function

$$\begin{aligned} \Phi_r(x,\mu,\nu) =& f(x) - \frac{1}{2r}\|\mu\|_{\mathscr{H}}^2 - \frac{1}{2r}\|\nu\|_{\mathscr{X}}^2 \\ &+ \frac{r}{2}\|\max\{0,\,(h_\ell - r^{-1}\mu) - h(x)\}\|_{\mathscr{H}}^2 + \frac{r}{2}\|\max\{0,\,h(x) - (h_u - r^{-1}\mu)\}\|_{\mathscr{H}}^2 \\ &+ \frac{r}{2}\|\max\{0,\,(x_\ell - r^{-1}\nu) - x\}\|_{\mathscr{X}}^2 + \frac{r}{2}\|\max\{0,\,x - (x_u - r^{-1}\nu)\}\|_{\mathscr{X}}^2. \end{aligned}$$

Therefore, at iteration k of the Moreau-Yosida penalty algorithm, we are given the current penalty parameter $r_k > 0$ and the current multiplier estimates μ_k and ν_k. Using this information we approximately solve the subproblem

$$\begin{aligned} &\min_{x\in\mathscr{X}} \ \Phi_{r_k}(x,\mu_k,\nu_k) \\ &\text{subject to} \quad g(x)=0 \end{aligned}$$

using ROL's Type-E capabilities. This produces the new iterate x_{k+1} which we then use to update he multiplier estimates as

$$\mu_{k+1} = \max\{0,\,(h_\ell - r^{-1}\mu_k) - h(x_{k+1})\} + \max\{0,\,h(x_{k+1}) - (h_u - r^{-1}\mu_k)\}$$

and

$$\nu_{k+1} = \max\{0,\,(x_\ell - r^{-1}\nu) - x\} + \max\{0,\,x - (x_u - r^{-1}\nu)\}.$$

The penalty parameter is either increased or decreased according to the progress of the algorithm. Notice that this algorithm is not required to produce feasible iterates x_k as we are not strictly enforcing the bound constraints. The Moreau-Yosida penalty algorithm is implemented in ROL as `ROL::MoreauYosidaPenaltyStep`.

Interior Point Methods The large production codes for the solution of numerical optimzation problems generally make use of sequential quadratic programming or primal-dual interior point methods (PDIP). Among the most heavily cited in the literature are KNITRO[6], LOQO[36] and IPOPT[37]. These codes have a long history and date back anywhere between a dozen and nearly 30 years. They do not, however, use application data structures as ROL does, leading to cumbersome and inefficient interfaces for PDE-constrained problems. Additionally, they may make use of factorizations of matrices which are prohibitively computationally expensive or even impossible to obtain in large-scale multiphysics problems. For example, primal-dual systems are almost always indefinite meaning their solution vectors are not not generally descent directions that would be of use to a globalizing line-search method. A typical remedy to this problem, taken by IPOPT, for example, is to add a multiple or block multiple of the identity to the primal-dual KKT matrix so as to make it positive definite. Appropriate weighting factors are determined using information gathered from matrix factorizations, which is not possible in a matrix-free setting.

In consideration of how interior point methods could be implemented in ROL, we reviewed the general approach taken by some of these production codes. ROL's first foray into implementing interior points roughly follows the primal approach taken in NITRO, the progenitor to KNITRO, which uses barrier penalities with SQP.

Consider the following nonlinear programming problem

$$\min_x f(x) \tag{2.22a}$$

$$\text{subject to} \quad c_e(x) = 0 \tag{2.22b}$$

$$c_i(x) \geq 0 \tag{2.22c}$$

$$\ell_k \leq x_k \leq u_k. \tag{2.22d}$$

To solve (2.22) using interior points and ROL, we rely on `ROL::OptimizationProblem` to generate the slack variables and create or modify equality and bound constraints. The bounds are then taken into the objective by means of a log barrier function leading to the Type-E problem

$$\min_{x,s} f(x) - \mu_s \sum_j^m \ln(s_j) - \mu_x \sum_k^n \{\ln(x_k - \ell_k) + \ln(u_k - x_k)\} \tag{2.23a}$$

$$\text{subject to} \quad c_e(x) = 0 \tag{2.23b}$$

$$c_i(x) - s = 0. \tag{2.23c}$$

Where m and n are the dimensions of the inequality constraint and optimization spaces, respectively. The logarithms of vectors are implemented using the elementwise operations described earlier.

Problems of this form for fixed penalty parameters μ_x and μ_s can be solved using ROL's Type-E capabilities such as `ROL::CompositeStep` or `ROL::AugmentedLagrangianStep`, which are encapsulated by the primal solver `ROL::IteriorPointStep`. This IP step reduces the objective through solving a sequence of penalized problems. Testing of this primal IP method on the HS test problems demonstrated slow convergence, typically comparable to `ROL::MoreauYosidaStep` or, in some cases, worse. The primal IP approach using log barriers is problematic when minimizers lie on boundaries as the unbounded objective forces iterates to remain strictly interior. This point is clearly in opposition to the needs of binary material design in topology optimization. Combining this observation with recent findings[34, 23] that have benchmarked the performance of a variety of general purpose and structural optimization problems motivates implementation of PDIP methods in ROL. A Type-EB problem with upper and lower bounds has the following structure for the linearized KKT system.

$$\begin{pmatrix} W_k & J_k & -I & I \\ J_k^\top & 0 & 0 & 0 \\ Z_k^\ell & 0 & X_k - L & 0 \\ -Z_k^u & 0 & 0 & U - X_k \end{pmatrix} \begin{pmatrix} \delta x_k \\ \delta \lambda_k \\ \delta z_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + J_k \lambda_k - z_k \\ c(x_k) \\ z_k^\ell (x_k - \ell) - \mu e \\ z_k^u (u - x_k) - \mu e \end{pmatrix} \tag{2.24}$$

where λ, z^ℓ, and z^u are the multipliers for the equality, lower, and upper bound constraints respectively, $J_k = \nabla c(x_k)$, and $W_k = \nabla_x^2 \mathscr{L}(x, \lambda, z_\ell, z_u)$. Use of the capital letter representation X_k denotes a diagonal matrix containing the iterate vector x_k as its diagonal band. The same applies to Z_k^ℓ and Z_k^u.

IPOPT solves the primal-dual KKT system and performs a line-search filter method that balances reduction in objective value and constraint norms with switching conditions. Many of the necessary components for a PDIP solver in the style of IPOPT have been implemented in ROL. These components include objects for the primal-dual residual, KKT hessian, symmetrization operators, merit function as well alternate method of introducing boundary constraints into objective penalties. As could be expected, however, when solving PDIP systems with conjugate gradient methods terminate after only a few iterations due to negative eigenvalues and GMRES typically requires nearly as many Arnoldi vectors as the dimensionality of the system, to only to produce

a non-descent vector. These results and the observation that the convergence of the KS and CL trust region models were contingent on the number of Arnoldi vectors produced before terminating due to negative eigenvalues highlight a need for greater ability to handle nonconvexity of the type arising in topology optimization problems.

Chapter 3

Topology Optimization for Structural Design

3.1 Overview

In this chapter, we consider the problem of designing minimally compliant elastic structures. Throughout, we let D denote the physical domain with boundary ∂D. We denote the Dirichlet (fixed) boundary as $\emptyset \neq \Gamma_d \subset \partial D$ and the Neumann (traction) boundary as $\Gamma_n = \partial D \setminus \Gamma_d$. We consider the minimum compliance topology optimization problem

$$\min_{0\leq z\leq 1} \ \ell(S(z)) \quad \text{subject to} \quad \int_D z(x)\,\mathrm{d}x \leq v_0|D| \tag{3.1}$$

where $Z = L^2(D)$, $v_0 \in (0,1)$ is the volume fraction, $\ell \in (H^1(D)^2)^*$ is a prescribed load and $u = S(z) \in H^1(D)^2$ solves the weak form of the linear elasticity equations

$$-\operatorname{div}(\rho(z)E : \varepsilon(u)) = \ell \qquad \text{in } D \tag{3.2a}$$

$$\varepsilon(u) = \tfrac{1}{2}(\nabla u + \nabla u^\top) \qquad \text{in } D \tag{3.2b}$$

$$u = 0 \qquad \text{on } \Gamma_d \tag{3.2c}$$

$$(\rho(z)E : \varepsilon(u))n = 0 \qquad \text{on } \Gamma_n. \tag{3.2d}$$

Here, n denotes the outward normal vector of Γ_n, E is the stiffness tensor and $\rho(z)$ is the the density

$$[\rho(z)](x) = (\rho_u - \rho_\ell)[F(z)](x)^3 + \rho_\ell, \quad 0 < \rho_\ell < \rho_u < \infty.$$

This specific form of ρ is known as the Solid Isotropic Material with Penalization (SIMP) model [1]. In the SIMP model, $F(z)$ is the filtered density. To filter, we utilize the Helmholtz filter described in [25]. The goal of filtering, in this context, is to enforce a length scale in the optimal design. We will additionally consider the volume topology optimization problem

$$\min_{0\leq z\leq 1} \int_D z(x)\,\mathrm{d}x \quad \text{subject to} \quad \ell(S(z)) \leq c_0\ell(S(z_0)) \tag{3.3}$$

where $z_0 \equiv 1$ and $c_0 > 0$. The constraint in optimization problem (3.3) can be viewed as a "safety" requirement. For example, any feasible design can not be more than twice (if $c_0 = 2$) as compliant as the solid beam.

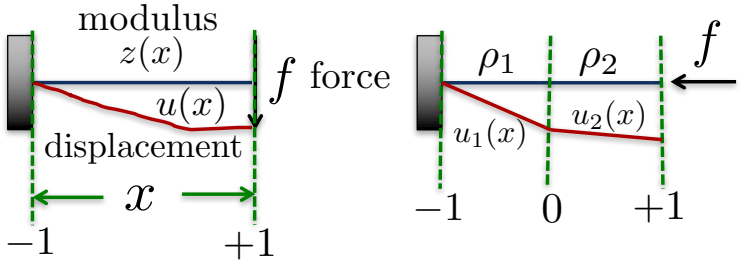

Figure 3.1. One-dimensional compressed beam and its two-element discretization.

3.2 Analytic Results

As topology optimization has most often been applied to the design of support structures, it is valuable to investigate in simplified limiting cases how the form of the problem leads to numerical challenges. Furthermore, problems with two-dimensional optimization spaces afford the additional advantage of being able to easily visualize where and why algorithms may exhibit slow convergence.

3.2.1 Two-element compressive beam model

A simple example that shares the bilinearity of the constraint equation is that of a compressed beam with a spatially varying modulus. This model is, admittedly, not physically realistic, however, it lends itself to a simple analytical solution, which makes it worth considering before proceeding to more physical models which can not be so easily visualized.

In this example, one aims to find the distributed material parameter that minimizes compression due to a fixed load and constraints on weight. Let the beam, depicted in 3.2.1, be modeled by a one-dimensional Poisson equation:

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\rho(x)\frac{\mathrm{d}u}{\mathrm{d}x}\right)=0, \quad u(-1)=0, \quad u'(1)=f \tag{3.4}$$

Using the standard cubic SIMP parameterization of the modulus $\rho(x)=\rho_l+(\rho_u-\rho_l)z^3(x)$ where $\rho_{l,u}$ is the lower and upper bounds of the modulus with $0\leq z\leq 1$, we can exactly solve the discretized boundary value problem when we assume a piecewise linear displacement and piecewise constant modulus. Supposing the domain is partitioned into two equal subintervals,

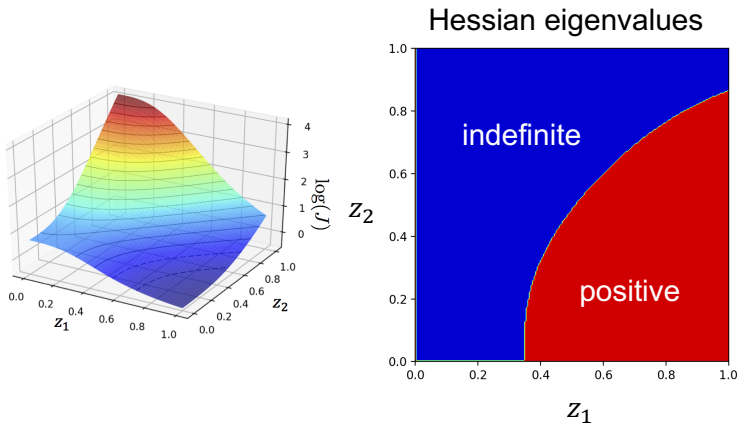

Figure 3.2. Logarithm of the objective function when $\rho_u = 10\rho_l$ and regions in which the objective is locally convex.

$\mathscr{I}_1 = [-1,0]$ and $\mathscr{I}_2 = [0,1]$, we have the optimization variables

$$z(x) = z_1\chi_1(x), \quad z_2\chi_2(x), \quad \chi_i(x) = \begin{cases} 1 & x \in \mathscr{I}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

and the exact displacement is

$$u(x) = \begin{cases} f\frac{\rho_l+(\rho_u-\rho_l)z_2^3}{\rho_l+(\rho_u-\rho_l)z_1^3} & x \in \mathscr{I}_1 \\ f\left(\frac{\rho_l+(\rho_u-\rho_l)z^3}{\rho_l+(\rho_u-\rho_l)z_2^3}+x\right) & x \in \mathscr{I}_2 \end{cases} \tag{3.6}$$

Since the displacement is exactly known in terms of the moduli, it is possible to write an objective function which minimizes compressive (or tensile) displacement as

$$J(u(z)) = \frac{1}{2}[u_2(+1)]^2 = \frac{1}{2}\left(1+\frac{\rho_u+(\rho_u-\rho_l)z_2^3}{\rho_u+(\rho_u-\rho_l)z_1^3}\right)^2 \tag{3.7}$$

and we can also write its gradient, Hessian, and inverse Hessian in closed form. Given the bound constraints, the objective is minimized when $z_2 = 1$ and $z_1 = 0$. The objective is plotted on a log scale in Figure 3.2. The negative curvature can be clearly seen and the regions in which the Hessian is indefinite or positive definite are shown as well.

When the ratio of ρ_u/ρ_l is made progressively larger, the region in which the Hessian has two positive eigenvalues becomes smaller. Applying the Kelley-Sachs (KS) and Coleman-Li (CL) trust region models to solving the optimization problem, we find that KS converges much faster than CL

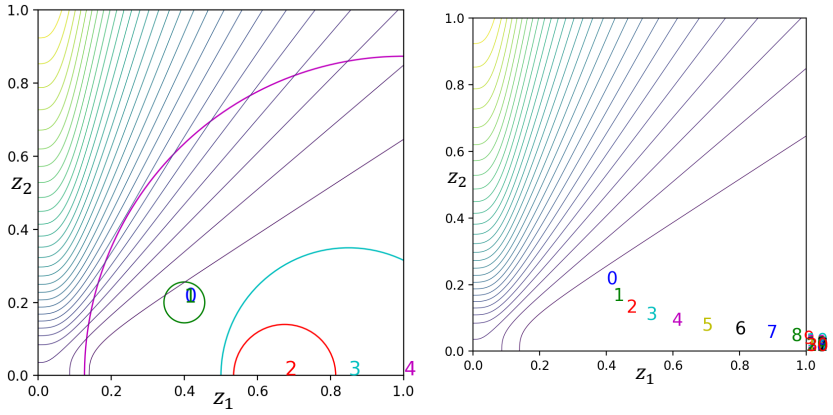

Figure 3.3. Locations of the optimization iterates for the Coleman-Li model (*left*) and Kelley-Sachs model (*right*) for the compressive beam problem.

due to quickly finding the lower boundary and proceeding along the boundary to the minimizer. CL, by contrast, follows a central path which approaches the $z = (1,0)$ corner asymptotically. The reason for this difference is that while both methods only generate one Arnoldi vector per iteration, this is sufficient for KS as the problem has become a one-dimensional minimization by the second iteration. Since CL remains strictly interior its truncated CG iteration terminates early and can not take as large of a step. The convergence of both KS and CL is depicted in Figure 3.3.

3.2.2 Euler-Bernoulli beam model

A more physically realistic model of a beam subjected to a load is the Euler-Bernoulli beam. The Euler-Bernoulli beam equation is a linear model of a beam with uniform cross-section, so that the resulting equation is a two-point boundary value problem. Like the above example and the electromagnetics design problems of interest, the PDE constraint is linear and the optimization variable (modulus) appears multiplying the simulation variable (deflection), however, the constraint equation is fourth order.

On the computational domain $\mathscr{I} = [0,L]$ let the beam be described by the boundary value problem

$$\frac{d^2}{dx^2}\left(E(x)\frac{d^2}{dx^2}w(x)\right) = 0, \quad w(0) = w'(0) = E(L)w''(L) = 0, \quad [Ew'']'(L) = F \tag{3.8}$$

where $w(x)$ is the deflection of the beam from equilibrium, $E(x)$ is a spatially-varying modulus,

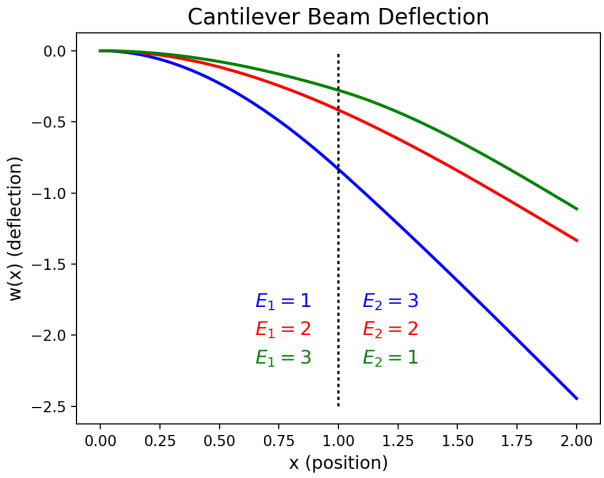

Figure 3.4. Deflection of a two-element Euler-Bernoulli beam subjected to a downward shear point force for three element modulus combinations.

and the boundary conditions indicate that the beam is clamped at the left end and subjected to a point shear force without torque at the right end. Let the domain be uniformly partitioned into n subintervals and the modulus be constant on each so that $E(x) = E_1\chi_1(x) + \cdots + E_n\chi_n(x)$ where

$$\chi_k(x) = \begin{cases} 1 & x \in \mathscr{I}_k \\ 0 & \text{otherwise} \end{cases}, \quad \mathscr{I}_k = [x_{k-1}, x_k], \quad x_k = \frac{kL}{n} \tag{3.9}$$

Since the problem is homogeneous, the deflection must be piecewise cubic.

$$w(x) = \sum_{j=1}^{n} \chi_k(x)(a_k + b_k x + c_k x^2 + d_k x^3), \quad \chi_k(x) = \begin{cases} 1 & x_{k-1} < x < x_k \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The coefficients can be determined from the boundary conditions and the requirement that $w \in C^1(\mathscr{I})$ and by satisfying jump conditions at the interfaces between subdomains. Given a test function $\phi \in C^1$, it is required that

$$\int_{x_k-\varepsilon}^{x_k+\varepsilon} \phi \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(E\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right)\mathrm{d}x = \phi \frac{\mathrm{d}}{\mathrm{d}x}\left[E\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right]_{x_k-\varepsilon}^{x_k+\varepsilon} - \int_{x_k-\varepsilon}^{x_k+\varepsilon} \frac{\mathrm{d}\phi}{\mathrm{d}x} E\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\,\mathrm{d}x = 0 \tag{3.11}$$

Surprisingly, it happens that one can write out the exact deflection and the objective.

$$J(E) = F^2 L F^3 \sum_{k=1}^{n} \frac{c_k^n}{E_k}, \quad c_k^n = \frac{3(n-k)(n-k-1)+1}{3n^3} \tag{3.12}$$

Similarly, the gradient and Hessian are

$$\frac{\partial J}{\partial E_i}(E) = -F^2 L^3 \frac{c_i^n}{E_i^2}, \quad \frac{\partial^2 J}{\partial E_i \partial E_j}(E) = 2F^2 L^3 \frac{c_j^n}{E_j^3} \delta_{ij} \tag{3.13}$$

This remarkable result shows that the bound constrained problem is everywhere convex. The Hessian is diagonal, so its inverse can be trivially applied for Newton's method. However, the Euler-Bernoulli beam constraint is illustrative of scaling problems that can arise in the gradient as the number of elements increases. The ratio of the largest to smallest element of the Hessian increases quadratically meaning that although the objective is convex, its level sets become severely distended with mesh refinement. It is also clear that the Newton step does not exhibit this mesh dependence

$$\delta E_k^{\text{Newton}} = -\left(\frac{\partial^2 J(\mathbf{E})}{\partial E_j E_k}\right)^{-1} \frac{\partial J(\mathbf{E})}{\partial E_k} = \frac{E_k}{2} \tag{3.14}$$

Linear elasticity constraints in two or three dimensions, unfortunately, do not give rise to a diagonal Hessian and so Newton-Krylov methods must be used instead. Additionally, this analysis has not yet included either a mass constraint or the requirement for a binary design. Applying SIMP to this problem removes its convexity and the above terms can not be written so succinctly.

The fourth-order beam problem can be solved using the finite element method with globally C^1 basis functions. We use Hermite interpolating polynomials. An implementation of arbitrary order Hermite interpolating basis functions was developed and added to the finite element package Intrepid [4].

The modulus can be taken to be piecewise constant on the elements. Let L be tall matrix containing the second derivatives of the global basis functions, $u \in \mathbb{R}^N$ be a vector containing the pointwise values and first derivatives of the displacement, $z \in \mathbb{R}^m$ be a vector of the modulus parameter with $0 \leq z \leq 1$. Let $p : [0,1] \to [\rho_\ell, \rho_u]$ be the interpolating material penalty function. The equality constraint can be written in matrix-vector form as

$$c(u,z) = L^\top \text{diag}[p(z)] L u - f \tag{3.15}$$

The constraint Jacobian terms can be expressed as

$$c_u v = L^\top \text{diag}[p(z)] L v, \quad c_z v = L^\top \text{diag}[p'(z) v] L u \tag{3.16}$$

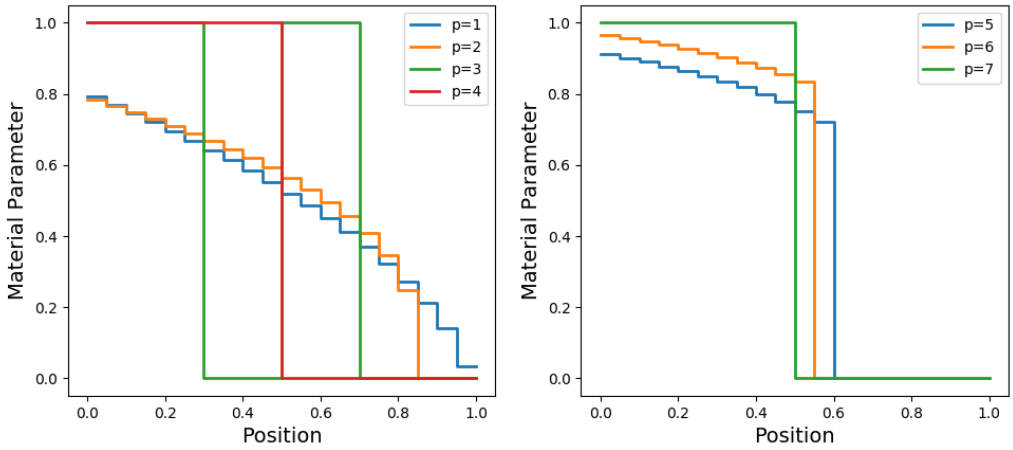

Figure 3.5. Effect of power p in the SIMP parameterization $E(z) = E_1 + (E_2 - E_1)z^p$ for computing the optimal beam composition subject to $0 \leq z \leq 1$ and that at most half of the structure be composed of the heavier, less compliant material.

The adjoint Hessian components are

$$c_{uu}(v,w) = 0, \quad c_{uz}(v,w) = L^\top \mathrm{diag}[p'(z)w]L^\top v, \quad c_{zz}(v,w) = L^\top [p''(z)vw]Lu \tag{3.17}$$

A small application code using ROL, Intrepid, Sacado, and Teuchos to formulate and solve the beam design problem has been developed. The Beam solver takes in a `Teuchos::ParameterList` which allows the user to select number of cells, polynomial order, and a variety of physical, SIMP, and optimization parameters to study the comparative performance of various solution methods in full and reduced space. Running the solver under numerous combinations with an inequality constraint on the mass such that its volume fraction may be at most half composed of the heavier, less compliant material, revealed that the most rapidly convergent method was to solve the reduced space problem using the augmented Lagrangian step. This is reasonable as the mass/volume inequality constraint is only a rank-one modification. While the Moreau-Yosida step worked well for small numbers of elements (¡10), its convergence rapidly deteriorated for increasing mesh refinement. This is reasonable because as the mesh is refined, more bound constraint variables are injected into the Moreau-Yosida penalty, whereas the scalar mass constraint relatively perturbs the augmented Lagrangian objective less with increasing mesh refinement.

The optimal material distribution was computed for a 20-element beam subjected to a unit downward shear point force on the free end with different powers in the SIMP formula. It was found that while the usual cubic power did lead to a binary solution, it did not lead to the least compliant solution: a bar that has the less compliant subdomain at the fixed end and the more compliant solution at the free end. Instead, cubic SIMP produces a design with a more compliant

subdomain to the left of the midpoint. In contrast, a quartic SIMP power gives exactly the idea solution, as does a power of 7, however, intermediate and greater powers produce less ideal results. It should be noted that this illustrates that the SIMP power is highly heuristic and problem-dependent and we have observed that the optimal power varies when other parameters in the solver are altered, suggesting that parameter studies are likely to be necessary whenever attempting to a solve a problem with different physics and optimization algorithms. Typical convergence histories for the 20 element beam with a SIMP penalty power $p = 7$ are given for Moreau-Yosida and Augmented Lagrangian below.

```
 Moreau-Yosida Penalty solver
  iter  fval           cnorm          gnorm          ifeas          snorm          penalty   #fval  #grad  #cval  subIter
  0     1.868626e+01   1.110223e-16   4.243629e+00   0.000000e+00                  1.00e+07
  1     9.146490e+00   5.551115e-17   8.686863e+00   7.158914e-06   1.704540e+00   1.00e+07  403    403    3      200
  2     9.136299e+00   1.110223e-16   3.845573e+00   0.000000e+00   2.692180e-03   1.00e+07  805    805    5      200
  3     9.126872e+00   1.110223e-16   1.308698e+01   1.943322e-05   2.523928e-03   1.00e+07  1207   1207   7      200
  4     9.113823e+00   1.110223e-16   3.971217e+01   1.195061e-04   3.539839e-03   1.00e+07  1609   1609   9      200
  5     9.082965e+00   2.220446e-16   6.374760e+01   4.269913e-04   8.390998e-03   1.00e+07  2011   2011   11     200
  6     9.064375e+00   1.665335e-16   3.845752e+01   1.505151e-04   5.019290e-03   1.00e+07  2413   2413   13     200
  7     9.047617e+00   0.000000e+00   2.591743e+01   6.854177e-05   4.567540e-03   1.00e+07  2815   2815   15     200
  8     9.036813e+00   1.110223e-16   2.611726e+01   5.786174e-07   2.973982e-03   1.00e+07  3217   3217   17     200
  9     9.026860e+00   1.110223e-16   4.316148e+01   1.917340e-04   2.763313e-03   1.00e+07  3619   3619   19     200
  10    9.016044e+00   0.000000e+00   2.649843e+01   7.191634e-05   2.973169e-03   1.00e+07  4021   4021   21     200

Material Vector:
0.999997   0.999997     1          0.997137  0.983323  0.941833  0.860208  0.740685   0.573279
0.433412   0.342481     0.278127   0.228509  0.187354  0.151102  0.117671  0.0858217  0.0548196
0.0242401  6.99213e-07
```

```
 Augmented Lagrangian solver
  iter  fval           cnorm          gLnorm         snorm          penalty   feasTol   optTol    #fval  #grad  #cval  subIter
  0     1.868626e+01   1.110223e-16   2.006954e+00                  1.00e+01  1.26e-01  2.01e-02
  1     2.000500e+00   5.000000e-01   0.000000e+00   2.236068e+00   1.00e+03  1.26e-01  1.00e-01  9      9      12     3
  2     6.493453e+00   3.288875e-02   3.206684e-02   3.574827e+01   1.00e+03  6.31e-02  1.00e-04  31     22     41     7
  3     6.729477e+00   1.262067e-02   1.304146e-06   1.378702e+01   1.00e+03  3.16e-02  1.00e-07  51     37     69     8
  4     6.563757e+00   4.068084e-03   4.471448e-06   4.139070e+00   1.00e+03  1.58e-02  1.00e-10  83     51     115    15
  5     6.525077e+00   1.709119e-03   1.293325e-09   1.735526e+00   1.00e+03  7.94e-03  1.00e-12  146    80     198    20
  6     6.512132e+00   8.497433e-04   3.752768e-13   8.618968e-01   1.00e+03  3.98e-03  1.00e-12  158    89     214    4
  7     6.507143e+00   5.039206e-04   1.109113e-13   5.094841e-01   1.00e+03  2.00e-03  1.00e-12  170    98     230    4
  8     6.504662e+00   3.264123e-04   4.380396e-11   3.299625e-01   1.00e+03  1.00e-03  1.00e-12  205    107    274    9
  9     6.504662e+00   3.264123e-04   6.528246e-03   3.264123e-01   1.00e+03  5.01e-04  1.00e-12  208    109    277    1
  10    6.504662e+00   3.264123e-04   6.528246e-03   3.264123e-01   1.00e+03  2.51e-04  1.00e-12  211    111    280    1

Material Vector:
1  1  1  1  1  1  1  1  1  0.993472  0  0  0  0  0  0  0  0  0  0
```

3.3 Scaling Results

When numerically solving infinite-dimensional optimization problems, such as the structural topology optimization problems discussed in this chapter, it is important to use algorithms that scale with problem size. That is, the number of optimization iterations, function and evaluations, etc., remain constant as the underlying discretization is refined. This scaling property is often called *mesh independence*. In this section, we investigate the mesh independent performance of ROL's Augmented Lagrangian implementation.

3.3.1 2D Results

We consider (3.1) constrained by (3.2) with $D = (0,2)\times(0,1)$, $\Gamma_d = \{0\}\times[0,1]$, and $v_0 = 0.4$. The load ℓ in this example is a Gaussian load of width 10^{-2} applied at $x = (2, 0.5)$. As with any penalty method for optimization, the performance of Augmented Lagrangian degrades when the penalty parameter $r > 0$ becomes large. Additionally, the performance depends on the initial choice of r. For example, if r is chosen to be too large, then the constraint violation is emphasized over minimizing the objective function. This imbalance can lead to poor performance. To circumvent some of these issues, we scale the objective function. Using Körn's inequality, we have (under appropriate assumptions on E) that there exists $e_0 > 0$ such that

$$\rho_\ell e_0 \|v\|^2_{H^1(D)^2} \leq \int_D (\rho(z) E\varepsilon(v)) : \varepsilon(v)\,\mathrm{d}x \quad \forall v \in H^1(D)^2. \tag{3.18}$$

The bilinear form on the right hand side of (3.18) is the bilinear form of the linear differential operator in (3.2). By continuity of the right hand side of (3.2) and the fact that u solves the weak form of (3.2), we additionally have that

$$\int_D (\rho(z) E\varepsilon(u)) : \varepsilon(v)\,\mathrm{d}x \leq \|\ell\|_{(H^1(D)^2)^*} \|v\|_{H^1(D)^2} \quad \forall\, v \in H^1(D)^2. \tag{3.19}$$

Combining (3.18) and (3.19) gives the following bound on the objective function in (3.1)

$$0 \leq \ell(S(z)) \leq (\rho_\ell e_0)^{-1} \|\ell\|^2_{(H^1(D)^2)^*} =: c_0.$$

We can then use c_0^{-1} to scale the objective function in (3.1) which ensures that the objective will be between zero and one. We similarly scale the volume constraint in (3.1) using the reciprocal of $\max\{v_0, 1-v_0\}|D|$ to ensure that it also remains between zero and one.

With the objective and constraint scaling balanced, we achieve near dimension-independent performance of ROL's Augmented Lagrangian method. In the following tables, we list the iteration histories for various discretization levels. Throughout, we discretize using continuous, piecewise linear finite elements on quadrilateral meshes. We start with a coarse discretization on a 32×16 mesh and refine to 64×32, 128×64, 256×128, 512×256 and 1024×512. Figure 3.3.1 depicts the optimal designs for the different mesh sizes. As demonstrated in these figures, our optimal solutions appear to be independent of mesh size. In addition, the subsequent tables demonstrate that, to

reach the desired stopping tolerances, the Augmented Lagrangian method required seven or fewer iterations for each mesh. For this example, ROL's Augmented Lagrangian approach appears to be mesh independent. However, the number of subproblem solver iterations (i.e., the `subIter` column) exhibits mild mesh dependence. The subproblem solver used for the Augmented Lagrangian subproblems (2.21) is ROL's Kelley-Sachs bound constrained trust-region solver. It is unknown why this algorithm is performing poorly and improving the performance of this solver is the subject of future investigation. We postulate that the inherent non-convexity of (2.21) may contribute to this poor performance since the conjugate gradients subproblem solver within ROL's Kelley-Sachs bound constrained trust-region solver typically detects negative curvature after a single iteration.

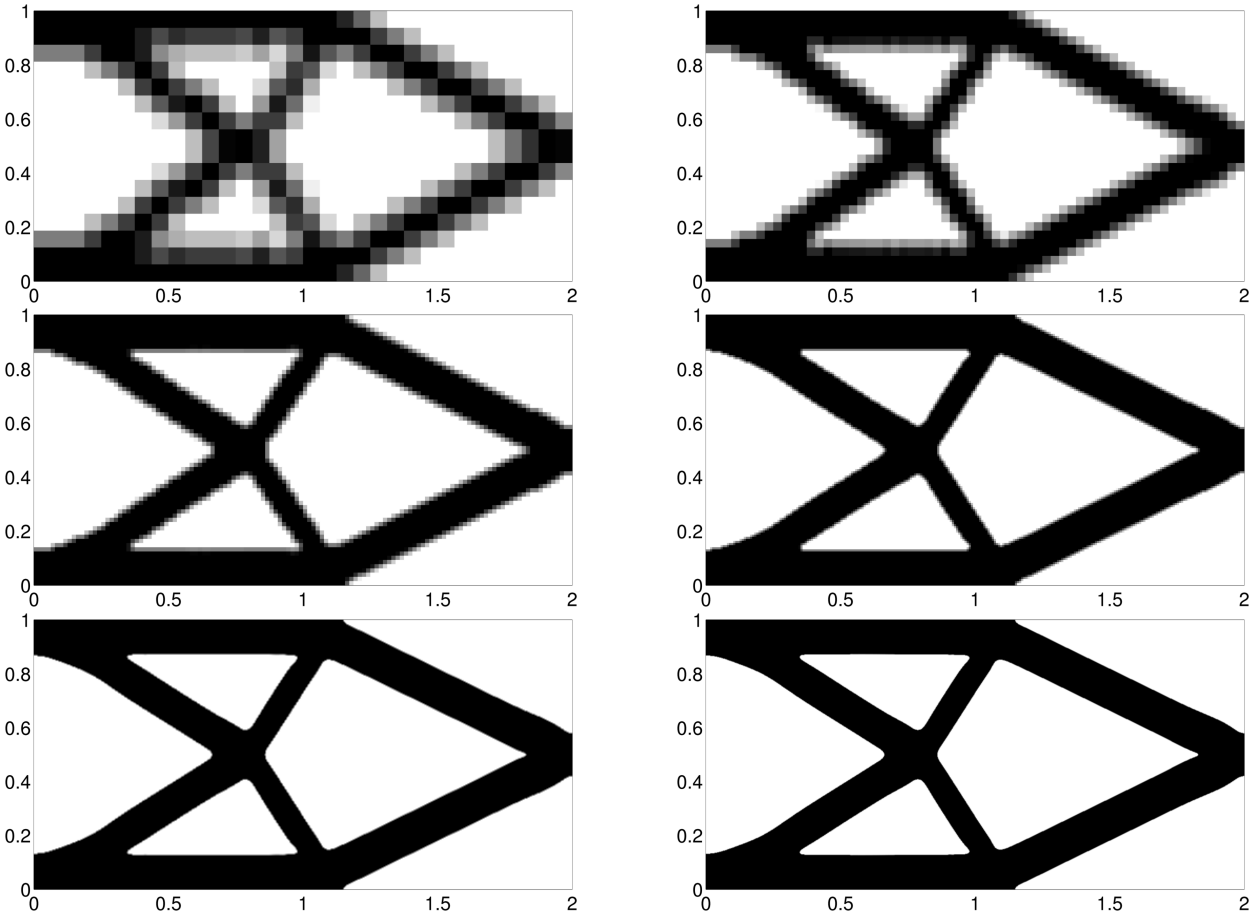

Figure 3.6. Averaged "unfiltered" density optimization variables z. Mesh refinement moves left to right and top to bottom: 32×16, 64×32, 128×64, 256×128, 512×256 and 1024×512.

32×16 **Mesh.**

```
Total optimization time = 5.69829 seconds.
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	8.184235e+00	2.312965e-17	8.443461e-01		1.00e+01	1.26e-01	8.44e-03				
1	6.432870e-01	5.945460e-01	5.829618e-03	7.727374e-01	1.00e+02	1.26e-01	1.00e-01	20	20	29	9
2	1.505951e+00	2.598534e-02	5.905832e-02	3.400903e+00	1.00e+02	7.94e-02	1.00e-03	62	42	82	11
3	1.586642e+00	4.867299e-03	3.466893e-03	6.650050e-01	1.00e+02	5.01e-02	1.00e-05	125	76	165	20
4	1.604720e+00	6.748430e-04	6.594977e-06	1.217652e-01	1.00e+02	3.16e-02	1.00e-05	160	98	212	12
5	1.607248e+00	9.473762e-05	1.849492e-06	1.309413e-02	1.00e+02	2.00e-02	1.00e-05	176	108	235	7
6	1.607603e+00	1.323183e-05	1.083312e-07	1.828044e-03	1.00e+02	1.26e-02	1.00e-05	181	114	242	2
7	1.607653e+00	1.884085e-06	2.785854e-06	2.497436e-04	1.00e+02	7.94e-03	1.00e-05	184	118	246	1

64×32 **Mesh.**

```
Total optimization time = 31.8461 seconds.
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	7.350310e+00	2.312965e-17	8.432039e-01		1.00e+01	1.26e-01	8.43e-03				
1	5.896663e-01	5.583371e-01	5.890546e-03	7.669962e-01	1.00e+02	1.26e-01	1.00e-01	26	26	38	12
2	1.265732e+00	2.060445e-02	6.029734e-02	2.846216e+00	1.00e+02	7.94e-02	1.00e-03	64	48	87	11
3	1.320557e+00	6.655961e-04	7.679560e-03	2.719648e-01	1.00e+02	5.01e-02	1.00e-05	125	82	168	20
4	1.322635e+00	9.465609e-05	2.259543e-04	1.054762e-01	1.00e+02	3.16e-02	1.00e-05	166	112	229	20
5	1.322374e+00	1.706988e-05	1.582025e-04	2.409165e-02	1.00e+02	2.00e-02	1.00e-05	206	142	288	20
6	1.322343e+00	7.862833e-06	9.226552e-06	1.308840e-02	1.00e+02	1.26e-02	1.00e-05	236	166	336	18

128×64 **Mesh.**

```
Total optimization time = 106.897 seconds.
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	7.103340e+00	2.312965e-17	8.428157e-01		1.00e+01	1.26e-01	8.43e-03				
1	5.718634e-01	5.514543e-01	6.374464e-03	7.632149e-01	1.00e+02	1.26e-01	1.00e-01	26	26	38	12
2	1.208526e+00	1.367780e-02	4.172568e-02	2.156943e+00	1.00e+02	7.94e-02	1.00e-03	70	52	97	15
3	1.231317e+00	4.153054e-03	8.186762e-04	6.104486e-01	1.00e+02	5.01e-02	1.00e-05	115	88	161	19
4	1.242531e+00	4.216716e-04	2.593033e-04	9.070282e-02	1.00e+02	3.16e-02	1.00e-05	155	120	221	20
5	1.243795e+00	5.409758e-07	4.839722e-06	3.799744e-02	1.00e+02	2.00e-02	1.00e-05	178	140	257	13

256×128 **Mesh.**

```
Total optimization time = 466.188 seconds.
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	7.069760e+00	2.312965e-17	8.427338e-01		1.00e+01	1.26e-01	8.43e-03				
1	5.684030e-01	5.530874e-01	8.406265e-03	7.575763e-01	1.00e+02	1.26e-01	1.00e-01	24	24	35	11
2	1.204637e+00	9.988045e-03	9.390741e-02	1.787331e+00	1.00e+02	7.94e-02	1.00e-03	76	52	103	16
3	1.212107e+00	5.974959e-03	1.000092e-03	8.040423e-01	1.00e+02	5.01e-02	1.00e-05	115	90	162	20
4	1.226482e+00	8.648857e-04	6.979474e-04	1.864555e-01	1.00e+02	3.16e-02	1.00e-05	154	122	220	20
5	1.228790e+00	6.240197e-05	4.387441e-07	3.126531e-02	1.00e+02	2.00e-02	1.00e-05	175	142	252	11
6	1.228938e+00	1.106293e-05	2.362168e-07	2.246611e-03	1.00e+02	1.26e-02	1.00e-05	181	148	261	3
7	1.228964e+00	1.972295e-06	4.233083e-07	3.929887e-04	1.00e+02	7.94e-03	1.00e-05	184	152	265	1

512×256 **Mesh.**

```
Total optimization time = 3276.86 seconds.
```

0	7.041291e+00	2.312965e-17	8.427030e-01		1.00e+01	1.26e-01	8.43e-03				
1	5.673678e-01	5.495446e-01	7.122910e-03	7.622124e-01	1.00e+02	1.26e-01	1.00e-01	26	26	38	12
2	1.185058e+00	1.555199e-02	3.787994e-02	2.341128e+00	1.00e+02	7.94e-02	1.00e-03	67	50	92	13
3	1.219717e+00	7.951031e-04	2.161753e-03	3.179712e-01	1.00e+02	5.01e-02	1.00e-05	115	90	160	20
4	1.221806e+00	9.060603e-07	3.634393e-04	9.348139e-02	1.00e+02	3.16e-02	1.00e-07	154	128	219	20
5	1.221776e+00	7.153583e-06	4.751420e-05	5.085859e-02	1.00e+02	2.00e-02	1.00e-07	192	164	277	20
6	1.221768e+00	9.636387e-06	1.061434e-05	2.220382e-02	1.00e+02	1.26e-02	1.00e-07	229	198	334	20
7	1.221789e+00	2.467628e-06	8.197599e-08	3.953958e-03	1.00e+02	7.94e-03	1.00e-07	263	226	388	20

1024×512 **Mesh.**

```
Total optimization time = 15296.5 seconds.
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	7.033946e+00	2.312965e-17	8.426949e-01		1.00e+01	1.26e-01	8.43e-03				
1	5.667705e-01	5.494992e-01	7.468271e-03	7.622279e-01	1.00e+02	1.26e-01	1.00e-01	26	26	38	12
2	1.188033e+00	1.305901e-02	3.058507e-02	2.095147e+00	1.00e+02	7.94e-02	1.00e-03	70	52	97	15
3	1.211831e+00	2.931495e-03	1.032600e-03	5.144561e-01	1.00e+02	5.01e-02	1.00e-05	111	90	158	20
4	1.219164e+00	2.939813e-04	4.240317e-04	1.116103e-01	1.00e+02	3.16e-02	1.00e-05	152	128	219	20
5	1.219645e+00	1.184069e-04	7.650312e-05	7.072162e-02	1.00e+02	2.00e-02	1.00e-05	192	166	278	20
6	1.220011e+00	1.086020e-05	2.665818e-05	2.457984e-02	1.00e+02	1.26e-02	1.00e-05	230	202	336	20
7	1.219975e+00	1.726753e-06	6.263598e-06	2.868126e-03	1.00e+02	7.94e-03	1.00e-05	241	212	353	6

3.3.2 3D Results

We consider (3.3) constrained by (3.2) with $D = (0,2) \times (0,1) \times (0,1)$ and $\Gamma_d = \{0\} \times [0,1] \times [0,1]$. The load ℓ in this example is a Gaussian load of width 0.125 applied at $x = (2, 0.5, 0.5)$. We further specify that a feasible design cannot have compliance larger than $c_0 = 3$ times the compliance of the solid beam. Figure 3.7 depicts the optimal designs for a $32 \times 16 \times 16$ mesh (top) and a $64 \times 32 \times 32$ mesh (bottom). For this problem setup, the resulting volume fraction was approximation 0.3. In the following table, we provide the iteration history from ROL's Augmented Lagrangian solver for the $32 \times 16 \times 16$ problem. To ensure mesh independence, we scaled the objective function and constraints in a similar fashion to that described in the Section 3.3.1. As with the 2D examples, the algorithm shows only mild mesh dependence.

$32 \times 16 \times 16$ **Mesh.**

```
Total optimization time = 2752.38 seconds.

Augmented Lagrangian solver
```

iter	fval	cnorm	gLnorm	snorm	penalty	feasTol	optTol	#fval	#grad	#cval	subIter
0	2.000000e+00	6.768891e-14	1.414214e+00		1.00e+01	1.26e-01	1.41e-02				
1	2.149712e-01	3.132267e-01	1.349219e-02	1.335167e+00	1.00e+02	1.26e-01	1.00e-01	66	37	88	22
2	4.912828e-01	2.666639e-02	4.400842e-02	3.175645e+00	1.00e+02	7.94e-02	1.00e-03	96	58	129	11
3	5.309233e-01	1.490753e-02	7.806634e-04	1.695502e+00	1.00e+02	5.01e-02	1.00e-05	125	87	172	14
4	5.671782e-01	6.728358e-03	4.590311e-06	8.611898e-01	1.00e+02	3.16e-02	1.00e-05	167	126	236	22
5	5.826644e-01	3.551886e-03	2.613437e-06	4.778663e-01	1.00e+02	2.00e-02	1.00e-05	185	143	263	9
6	5.925839e-01	1.632762e-03	8.096817e-06	2.609762e-01	1.00e+02	1.26e-02	1.00e-05	208	166	297	11
7	5.953950e-01	1.103146e-03	2.449657e-06	1.623867e-01	1.00e+02	7.94e-03	1.00e-05	223	181	319	7

8	5.977847e-01	6.641208e-04	2.005720e-07	1.146461e-01	1.00e+02	5.01e-03	1.00e-05	248	204	357	13
9	5.987113e-01	4.967627e-04	1.099315e-07	7.496046e-02	1.00e+02	3.16e-03	1.00e-05	264	219	381	8
10	6.006522e-01	1.482794e-04	4.273121e-08	5.671474e-02	1.00e+02	2.00e-03	1.00e-05	282	236	408	9
11	6.007579e-01	1.294053e-04	6.649835e-06	1.549910e-02	1.00e+02	1.26e-03	1.00e-05	285	239	412	1
12	6.008365e-01	1.153843e-04	2.403128e-06	1.332930e-02	1.00e+02	7.94e-04	1.00e-05	288	242	416	1
13	6.009054e-01	1.031171e-04	1.742653e-06	1.188013e-02	1.00e+02	5.01e-04	1.00e-05	291	245	420	1
14	6.009650e-01	9.254027e-05	1.166701e-06	1.059768e-02	1.00e+02	3.16e-04	1.00e-05	294	248	424	1
15	6.010171e-01	8.329581e-05	8.260544e-07	9.499960e-03	1.00e+02	2.00e-04	1.00e-05	297	251	428	1
16	6.014745e-01	1.049919e-06	5.521129e-06	2.245776e-02	1.00e+02	1.26e-04	1.00e-05	307	260	443	5

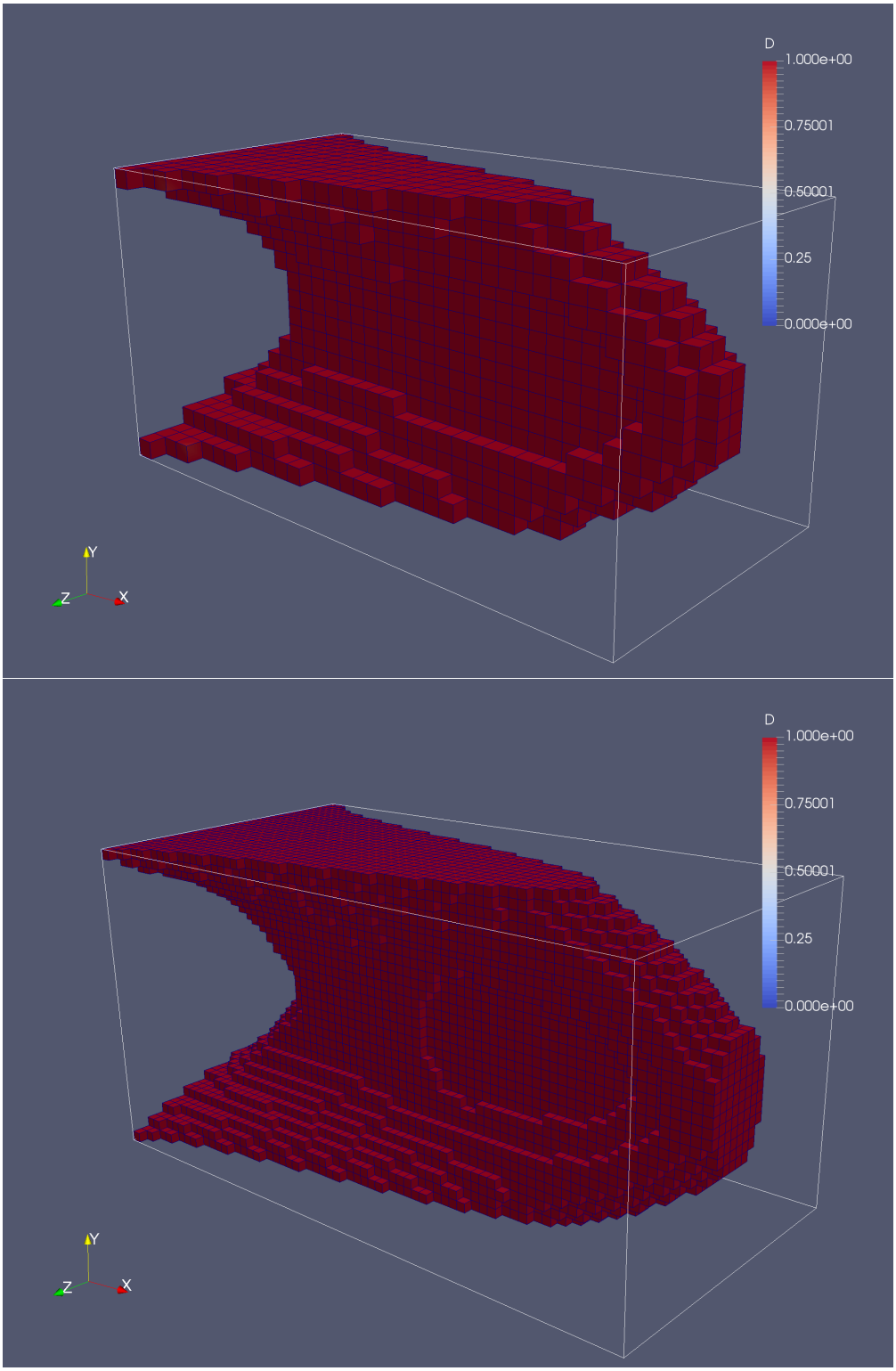

Figure 3.7. Averaged "unfiltered" density optimization variables z for a $32 \times 16 \times 16$ mesh (top) and a $64 \times 32 \times 32$ mesh (bottom).

Chapter 4

Design of Conducting Networks

This chapter develops topology optimization formulation for use in design of electromagnetic conducting networks. Section 4.1 presents a review of Maxwell's equations and describes our discretization and solution approach. Section 4.2 presents a four contact electrostatic design formulation. The goal of this formulation is to design two wires that can carry current between pairs of conductors. Finally, Section 4.3 develops a time dependent formulation for design of the inner MITL or, more generally, coaxial cable topologies.

4.1 Maxwell's Equations

Maxwell's equations collectively describe the evolution of electromagnetic fields

$$\varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J} = \frac{1}{s_B \mu_0} \nabla \times \mathbf{B} \tag{4.1}$$

$$\frac{\partial \mathbf{B}}{\partial t} + s_B \nabla \times \mathbf{E} = 0 \tag{4.2}$$

$$\nabla \cdot \mathbf{E} = \varepsilon_0^{-1} \rho \tag{4.3}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{4.4}$$

where Eq. 4.1 is Ampere's law, Eq. 4.2 is Faraday's law, Eq. 4.3 is Gauss' law, and Eq. 4.4 is Gauss' law for magnetism or the solenoidal condition [3, 14]. Here $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, $\mathbf{J}$ is the current and ρ is the charge density. The constants ε_0 and μ_0 are the permittivity and permeability. We introduced the constant s_B as a scaling of the magnetic field used to control the relative magnitudes of the electric and magnetic fields to ease the numerical solution of Maxwell's equation. Note that with the addition of a continuity relation on the charge density (e.g. $\partial_t \rho + \nabla \cdot \mathbf{J} = 0$) Eq. 4.3 is automatically satisfied if the initial condition also satisfies Eq. 4.3. Similarly with Eq. 4.4 for $\mathbf{B}$. Below we will assume that current is proportional to the electric field, this is Ohm's law

$$\mathbf{J} = \sigma \mathbf{E} \tag{4.5}$$

where σ is a potential spatially varying conductivity.

The electromagnetic energy is a function of the fields

$$W = \frac{1}{2\mu_0}\mathbf{B}\cdot\mathbf{B} + \frac{\varepsilon_0}{2}\mathbf{E}\cdot\mathbf{E} \tag{4.6}$$

whose evolution satisfies the conservation equation

$$\frac{\partial W}{\partial t} + \mu_0^{-1}\nabla\cdot(\mathbf{E}\times\mathbf{B}) = -\mathbf{J}\cdot\mathbf{E}. \tag{4.7}$$

This can be easily derived by talking the time derivative of W and substituting Maxwell's equations. The term on the right is a dissipative term that accounts for energy loss due to Ohmic heating. The flux term $\mathbf{P} = \mathbf{E}\times\mathbf{B}$ is the Poynting vector that describes the flow of electromagnetic energy.

4.1.1 Coaxial Cables

An important application of Maxwell's equations for this project is a coaxial cable. These transmission lines are designed to carry a signal over long distances. For the Z-machine, the magnetically insulated transmission lines can be (grossly) simplified as a coaxial cable that needs to transmit a large fast electromagnetic pulse with relatively little loss. Figure 4.1 shows a schematic of a coaxial cable with red being the conductor, and blue a void region. The general structure, is an inner cylindrical conductor nested inside an outer cylindrical conductor. In the left image, terms from the energy evolution equation give a rough estimate of the behavior of electromagnetic energy as it flows through the cable. On the bottom, the Poynting flux indicates an inflow of power, while on the top the energy flows out. On the outer walls, a perfect conductor is used in this model that restricts the flow of energy in or out through the sides. Finally, in the volume energy is lost through the Ohmic heating term. This term is larger with increasing conductivity. The right image gives a top down view of the direction of the fields in the coaxial cable used in this report. The electric field, which drives a potential difference, is radially pointing outward in the gap between the conductors. While the magnetic field circles the current flowing in the inner conductor. The current flows in opposite directions in each conductor.

4.1.2 Weak Form and Compatible Finite Element Discretization

A variational form for Maxwell's equation is to find $\mathbf{E}\in H_{\nabla\times}(\Omega)$ and $\mathbf{B}\in H_{\nabla\cdot}(\Omega)$ such that

$$\int_\Omega (\varepsilon_0\partial_t\mathbf{E}+\mathbf{J})\cdot\psi - \mu_0^{-1}\mathbf{B}\cdot\nabla\times\psi\,dx + \int_{\partial\Omega}\mu_0^{-1}(\mathbf{n}\times\mathbf{B})\cdot\psi\,dx + \int_\Omega(\partial_t\mathbf{B}+\nabla\times\mathbf{E})\cdot\phi = 0 \quad \forall\psi\in H_{\nabla\times}(\Omega),\forall\phi\in H_{\nabla\cdot}(\Omega). \tag{4.8}$$

Note that the two Gauss' law constraints are automatically satisfied through the choice of space and by the assumption the constraints are satisfied at the initial time. Eq. 4.4 is satisfied because

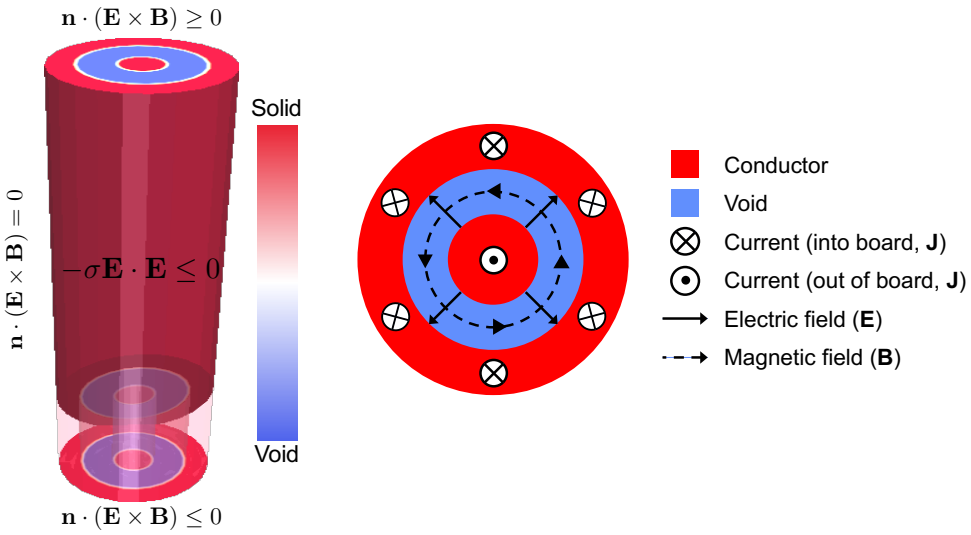

Figure 4.1. (Left) A schematic view of an idealized coaxial cable. Power flows in from the bottom, and out the top, while no loss occurs through the sides, while energy is lost in the volume due to Ohmic heating. Solid conducting material is depicted in red, while the void region is depicted in blue. (Right) A top down view of the fields in the coaxial cable.

$\nabla \times \mathbf{E} \in H_{\nabla\cdot}$ and $\nabla \cdot \nabla \times = 0$, thus $\partial_t(\nabla \cdot \mathbf{B}) = 0$. For Eq. 4.3, select $\psi = \nabla\omega \in H_{\nabla\times}$ for any $\omega \in H_\nabla$ and $\phi = 0$, then Eq. 4.8 reduces to (for simplicity ignoring boundary conditions)

$$\int_\Omega (\varepsilon_0 \partial_t \mathbf{E} + \mathbf{J}) \cdot \nabla\omega \, dx = 0 \quad \forall \omega \in H_\nabla(\Omega) \tag{4.9}$$

$$\Rightarrow \int_\Omega \varepsilon_0 \partial_t \mathbf{E} \cdot \nabla\omega - \nabla \cdot \mathbf{J}\omega \, dx = 0 \quad \forall \omega \in H_\nabla(\Omega). \tag{4.10}$$

This last statement, combined with a continuity condition on the charge density $\partial_t \rho = -\nabla \cdot \mathbf{J}$ gives a weak expression for Eq. 4.3 showing that Gauss's law is satisfied given that the initial conditions are satisfied.

Spatial discretization of Eq. 4.8 uses the compatible approaches described and implemented in Intrepid [5]. These discretizations mirror the properties of the continuous spaces. For instance, the compatible spaces $V_\nabla \subset H_\nabla$, $V_{\nabla\times} \subset H_{\nabla\times}$, and $V_{\nabla\cdot} \subset H_{\nabla\cdot}$ satisfy (with abuse of notation)

$$\nabla V_\nabla \in V_{\nabla\times} \text{ and } \nabla \times V_{\nabla\times} \in V_{\nabla\cdot}. \tag{4.11}$$

These properties, allow the trivial extension of the analysis for the continuous problem to the discrete system using the weak form in Eq. 4.8 with proper substitution of the discrete spaces. In all cases below, piecewise linear *edge* ($H_{\nabla\times}$) and *face* ($H_{\nabla\cdot}$) discretizations are used.

Applying implicit Euler to the weak form in Eq. 4.8 implies the variational form to find $\mathbf{E}^{n+1} \in V_{\nabla\times}$ and $\mathbf{B}^{n+1} \in V_{\nabla\cdot}$ given $\mathbf{E}^n$ and $\mathbf{B}^n$:

$$\begin{aligned}\int_\Omega (\Delta t^{-1}\varepsilon_0 \mathbf{E}^{n+1} + \mathbf{J}^{n+1}) \cdot \psi - \mu_0^{-1}\mathbf{B}^{n+1} \cdot \nabla \times \psi \, dx + \int_{\partial\Omega} \mu_0^{-1}(\mathbf{n} \times \mathbf{B}^{n+1}) \cdot \psi \, dx \\ + \int_\Omega (\Delta t^{-1}\mathbf{B}^{n+1} + \nabla \times \mathbf{E}^{n+1}) \cdot \phi \, dx = \int_\Omega \Delta t^{-1}\varepsilon_0 \mathbf{E}^n \cdot \psi \, dx + \int_\Omega \Delta t^{-1}\mathbf{B}^n \cdot \phi \, dx\end{aligned} \tag{4.12}$$

for all $\psi \in V_{\nabla\times}, \forall \phi \in V_{\nabla\cdot}$. The implicit solution of this system allows us to step over the speed of light stability limit in a stable fashion. However, efficient solution requires an effective preconditioner. We have developed a class of block preconditioner specifically designed for this formulation of Maxwell's equations (see [11]). This preconditioner is effective for linear systems derived from Eq. 4.12 with an Ohm's law model for the current in the form

$$\begin{bmatrix} (\Delta t^{-1}\varepsilon_0 + \sigma)M_E & \mu_0^{-1}C \\ C^T & \Delta t^{-1}M_B \end{bmatrix} \begin{bmatrix} E^{n+1} \\ B^{n+1} \end{bmatrix} = \begin{bmatrix} f_E^n \\ f_B^n \end{bmatrix} \tag{4.13}$$

where M_B, and M_E are the mass matrices for B and E, C is the discrete curl (whose transpose is also a discrete curl), and the right hand side is defined by Eq. 4.12.

4.2 Electrostatics

4.2.1 Minimum Energy Dissipation Formulation

The problem of minimizing the energy dissipation in a conductor between two or more contacts was considered in [18]. For this 2D problem the domain Ω, has a boundary $\partial\Omega$ subdivided into

N_c contacts labeled Γ_c. Each contact had a prescribed current I_c that satisfies the compatibility condition $\sum_c^{N_c} I_c = 0$. The PDE constraint is defined using the Eddy current form of Maxwell's equations that is simplified to the electrostatic limit. Finally, they impose a volume constraint, v_f that is a fraction of the volume of Ω. This results in the optimization formulation:

$$\min_z \quad \sum_{c=1}^{N_c} \int_{\Gamma_c} (-\sigma \nabla \phi) \cdot n\phi \, ds \tag{4.14}$$

subject to

$$-\nabla \cdot \sigma(z) \nabla \phi = 0, \tag{4.15}$$

$$\nabla \sigma \phi \cdot n|_{\Gamma_c} = I_c \quad \text{for all } c, \tag{4.16}$$

$$\phi|_{\partial\Omega/\cap_c \Gamma_c} = 0, \tag{4.17}$$

$$\int_\Omega z \, dx - v_f |\Omega| = 0, \tag{4.18}$$

$$0 \le z \le 1, \tag{4.19}$$

where the conductivity $\sigma(z) = \sigma_{min} + (\sigma_{max} - \sigma_{min})z$, and ϕ is the electrostatic potential,

The approach summarized here was demonstrated to connect one inflow contact to multiple outflow contacts. For multiple inflow contacts, the conducting material merges and is then distributed to each outflow contact. For the coaxial problem described above these designs would not connect distinct pairs of contacts with conducting material.

4.2.2 Design of a Two Wire System

This section considers the design of a two wire conducting network using an electrostatic formulation similar to the one presented above. The image on the left in Fig. 4.2 shows a four contact box where current will be driven by a potential difference on the left contacts and grounded contacts on the right. The right half of the image labels each of the contacts. The design goal, is to develop two straight wires connecting contacts Γ_A to Γ_D and Γ_B to Γ_C. To do this we have developed two optimization formulations motivated by the minimum energy dissipation formulation from [18] discussed above. The first formulation successfully designs the two wires; however, the wires are not straight. The second formulation adds a regularization parameter based on a minimal interface criteria used in phase transitions to straighten the wires.

Formulation 1

Considering Fig. 4.2, the flux through the domain is driven by the potential difference between the contacts. The contacts on the right represent a ground, and the contacts on the left represent a source with a voltage difference between them. Ideally the large potential gradient between the two left contacts would be resolved by delivering power in the form of current to the ground contacts

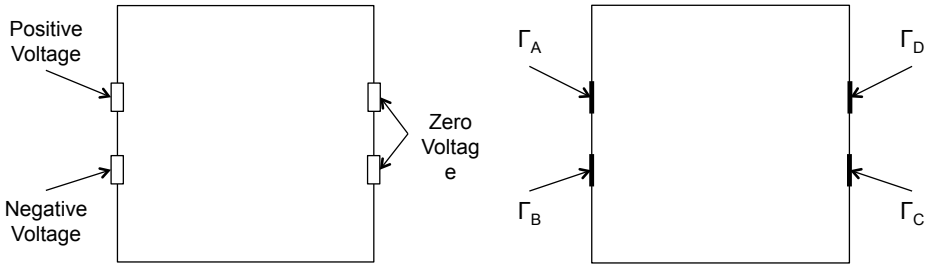

Figure 4.2. On the left a four contact device is pictured, with potential values specified. On the right is the computational domain Ω and its boundaries defining the location of the contacts is pictured.

on the right. To do this we will maximize the current flowing over the right contacts. Note, without strategic placement of the conductors, relying instead on a uniform conductor, power would flow between the two left contacts with relatively little current reaching the contacts on the right.

Define the approximation of the current on a contact denoted by $\Gamma \subset \partial\Omega$ as

$$I(\Gamma, z) \approx \int_{\Gamma} \sigma(z)\nabla\phi \cdot n \, ds. \tag{4.20}$$

Then the optimization problem for design of a two wire system is

$$\min_{z \in V} \quad I(\Gamma_C, z) - I(\Gamma_D, z) \tag{4.21}$$

subject to

$$-\nabla \cdot \sigma(z)\nabla\phi = 0 \tag{4.22}$$
$$\phi|_{\Gamma_A} = \phi_0 \tag{4.23}$$
$$\phi|_{\Gamma_B} = -\phi_0 \tag{4.24}$$
$$\phi|_{\Gamma_C} = 0 \tag{4.25}$$
$$\phi|_{\Gamma_D} = 0 \tag{4.26}$$
$$\int_{\Omega} z \, dx - v_f|\Omega| \leq 0 \tag{4.27}$$
$$0 \leq z \leq 1. \tag{4.28}$$

where

$$V = \{v \in H^1(\Omega) : v|_{\Gamma} = 1 \text{ where } \Gamma = \Gamma_A \cup \Gamma_B \cup \Gamma_C \cup \Gamma_D \text{ and } v|_{\partial\Omega/\Gamma} = 0\}. \tag{4.29}$$

Here the space V defines boundary conditions on the control z, boundary conditions and spaces for the state variable follow from Eqs. 4.22-4.26 using a standard finite element discretization. The

boundary conditions specified for the PDE correspond to those shown in Fig. 4.2. Note that the objective is minimizing the difference between the currents over the contacts on the right side of the domain. This minimization will force the current over contact Γ_C to be as small (negative) as possible, while Γ_D will have a current as large (positive) as possible. This corresponds to maximizing the magnitude of the current over these two contacts. The parameter v_f is the fraction of the volume that the conductive material is allowed to fill. To handle the volume constraint (Eq. (4.27)) we will reformulation the objective function using Moreau-Yosida regularization and include the inequality in the objective (see Section 2.3.2). Thus the new optimization system is

$$\min_{z\in H^1(\Omega)} \quad I(\Gamma_C,z) - I(\Gamma_D,z) + \frac{w_z}{3}\max\left(\int_\Omega z\,dx - v_f|\Omega|, 0\right)^3 \tag{4.30}$$

subject to the PDE strong form in Eqs. (4.22)-(4.26). Here w_z is a user specified weighting parameter. Finally the conductivity σ is modeled by the SIMP method [35]. Where the relationship to the control z is

$$\sigma(z) = \sigma_{min} + (\sigma_{max} - \sigma_{min})z^3. \tag{4.31}$$

Here σ_{min} is the conductivity of the void material and σ_{max} is the conductivity of the solid material.

The model parameters are

$$v_f = 0.25,\ w_z = 10^6,\ \sigma_{min} = 10^{-8} \text{ and } \sigma_{max} = 1.45\times 10^6. \tag{4.32}$$

The mesh was 100×100 run on 12 processors. This was solved using the ROL Truncated CG Trust Region method. The Poisson operator was discretized using the finite element method implemented in **Drekar** . The initial condition for the control was a uniform field of $z = 0.25$. Finally, the definition of the numerical current follows from direct substitution of the numerical potential ϕ_h into Eq. 4.20, e.g.:

$$I_{low}(\Gamma,z) \approx \int_\Gamma \sigma(z)\nabla\phi_h\cdot n\,ds. \tag{4.33}$$

This form of the current is accurate to only first order. At the final step the currents and volume values were

$$\int_{\Gamma_C}\sigma(z)\nabla\phi\cdot\vec{n}\,ds = 60.6692597,\ \int_{\Gamma_D}\sigma(z)\nabla\phi\cdot\vec{n}\,ds = -56.8403947 \text{ and } \int_\Omega s\,ds = 0.270162055 \tag{4.34}$$

The left image in Figure 4.3 shows the placement of conductive material (red). From this image its clear that the optimization problem finds two wires. However, there are a number of interesting issues. First the wires clearly bend. In addition, wires are disconnected near the C and D contacts. The gap appears regardless of mesh resolution and is always one element wide. This is blamed on the low order form of the numerical current calculation, thus we will develop a second order form.

A second order form of the current can be derived for use with the finite element method by carefully applying integration by parts. For a contact on Γ, define a subdomain $\Omega_s\subset\Omega$ such that $\Gamma\subset\partial\Omega_s$. Let v be a function in $H^1(\Omega)$ such that $v|_\Gamma = 1$ and $v|_{\Omega/\Omega_s} = 0$. The last equality implies that v has support only on Ω_s. Assume additionally that the length (or measure) of $\partial\Omega_s/\Gamma$ is *small*.

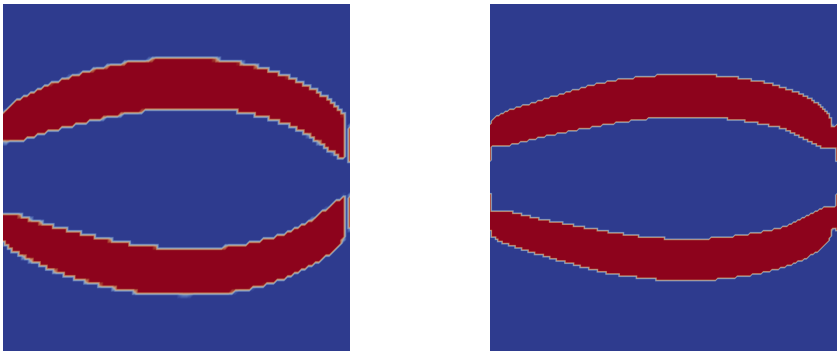

Figure 4.3. The result of a gradient descent trust region algorithm applied to the PDE constrained optimization formulation in Eqs. 4.21-4.28. The optimal solution on the left was computed using a numerical current defined in Eq. 4.33 , while the solution on the right was computed with the second-order numerical current in Eq. 4.37. Note that the gap in front of the right contacts is closed by the second-order current definition.

Then

$$\int_\Gamma \sigma\nabla\phi\cdot\vec{n}\,ds \approx \int_{\partial\Omega_s} v\sigma\nabla\phi\cdot\vec{n}\,ds \tag{4.35}$$

$$= \int_{\Omega_s} \sigma\nabla\phi\cdot\nabla v + v\nabla\cdot\sigma\nabla\phi\,dx. \tag{4.36}$$

The last term in this expression is simply the weak residual of the Laplician equation (for the PDE-constraint $\nabla\cdot\sigma\nabla\phi = 0$). If Ω_s is the set of elements that share at least a node with Γ, then v is defined to be 1 on Γ and 0 on the remaining nodes. Thus the current can be approximated as

$$I_2(\Gamma,z) = \int_{\Omega_s(\Gamma)} \sigma(z)\nabla\phi_h\cdot\nabla v_\Gamma\,ds \tag{4.37}$$

where $\Omega_s(\Gamma)$ and v_Γ are defined as describe above.

Using the second order definition to compute the current, and applying the above optimization formualtion with the model parameters:

$$v_f = 0.22,\ w_z = 10^6,\ \sigma_{min} = 10^{-8} \text{ and } \sigma_{max} = 1.45\times 10^6. \tag{4.38}$$

The mesh was 160×160 run on 16 processors. This was solved using ROL (Truncated CG). The initial condition was a uniform field of $z = 0.2$.

The right image in Fig. 4.3 shows the results of optimizing with the second order current. In particular the gaps from the first order formulation have now closed. The far left image in Figure 4.4 shows the currents and the volume as a function of the number of iterations. First notice

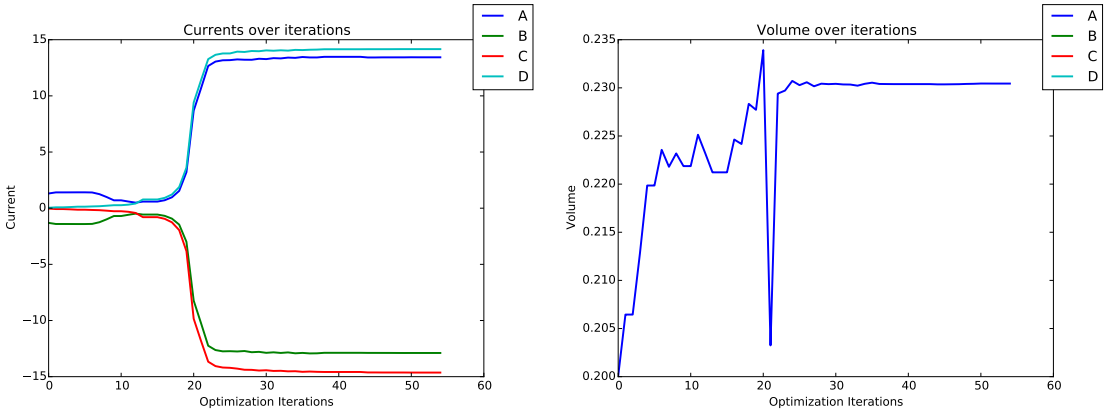

Figure 4.4. From left to right these figures shows the currents and the volumes plotted for each iteration of the optimization algorithm.

that the A current is postive while the opposite contact C has negative current. This is as expected. Similarly D has positive current and B has negative current. Finally, notice that the conservation properties are not ideal and that the currents don't exactly balance. In fact a closer examination shows that A and B currents are closer to equal and opposite than (for instance A and C). This is also reflected in C and D. This is a result of leakage between the wires.

A brief mesh study is presented in Figures 4.5 and 4.6. These show the behavior of the optimization algorithm over three different mesh sizes.

Two theories we considered for the bending of the wires were 1) leakage between the wires resulted in a greater loss of current when the wires were straight and closer together, 2) the objective surface has many local minima and bent wires is the first reached. To investigate the source of the bending wires we ran the same optimization formulation with straight wires set as the initial condition. The result was the wires stay straight. The wires were not forced to a bent position where any leakage between the two would be reduced. Our preliminary conclusion is that the bent wires are a result of non-uniqueness in the objective function and the added length of the wires is not a substantial loss mechanism for this model.

Formulation 2

The change to using the high order current calculation in Formulation B removed the gaps at the boundary. However, the bending of the wires remained. To address this, we developed a topology optimization formulation that minimizes the surface area between the conductor and void space (red and blue in the images above). This is based on theory from the phase field community developed in [29]. This theory is used to define an energy functional whose constrained minima

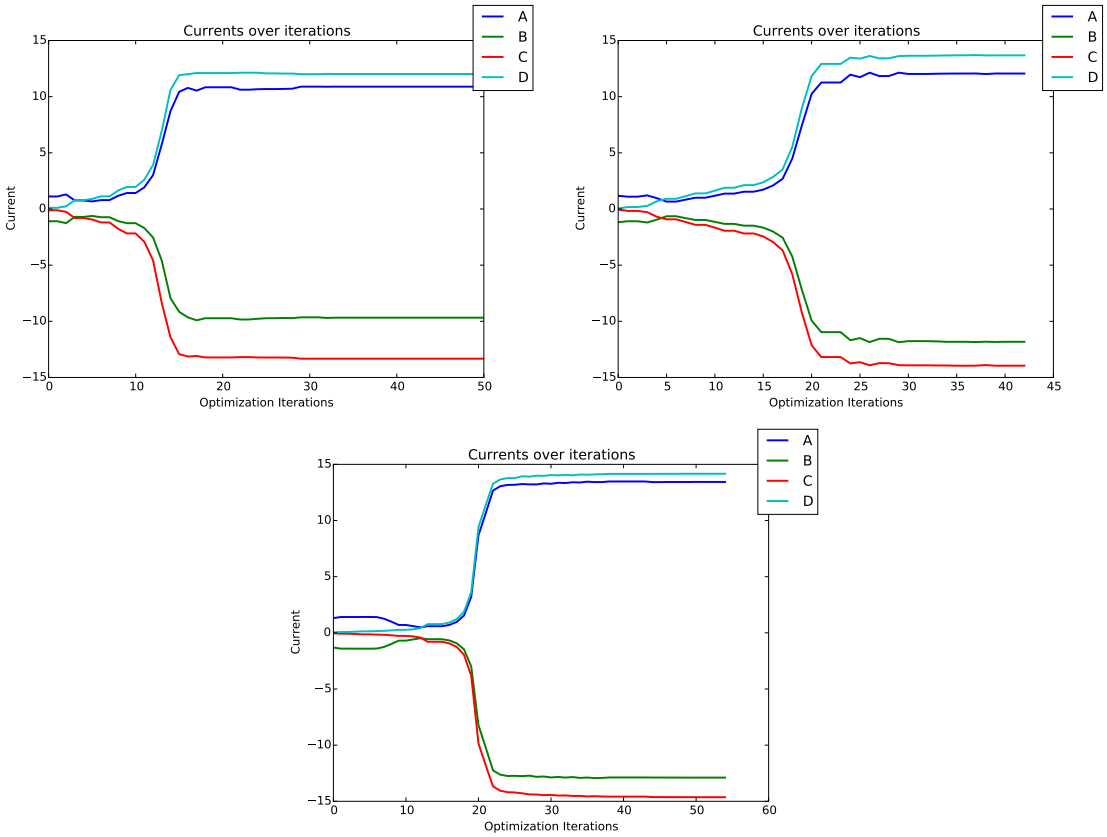

Figure 4.5. These figures shows the current over a range of mesh sizes, from left to right and top to bottom are 40×40, 80×80 and 160×160.

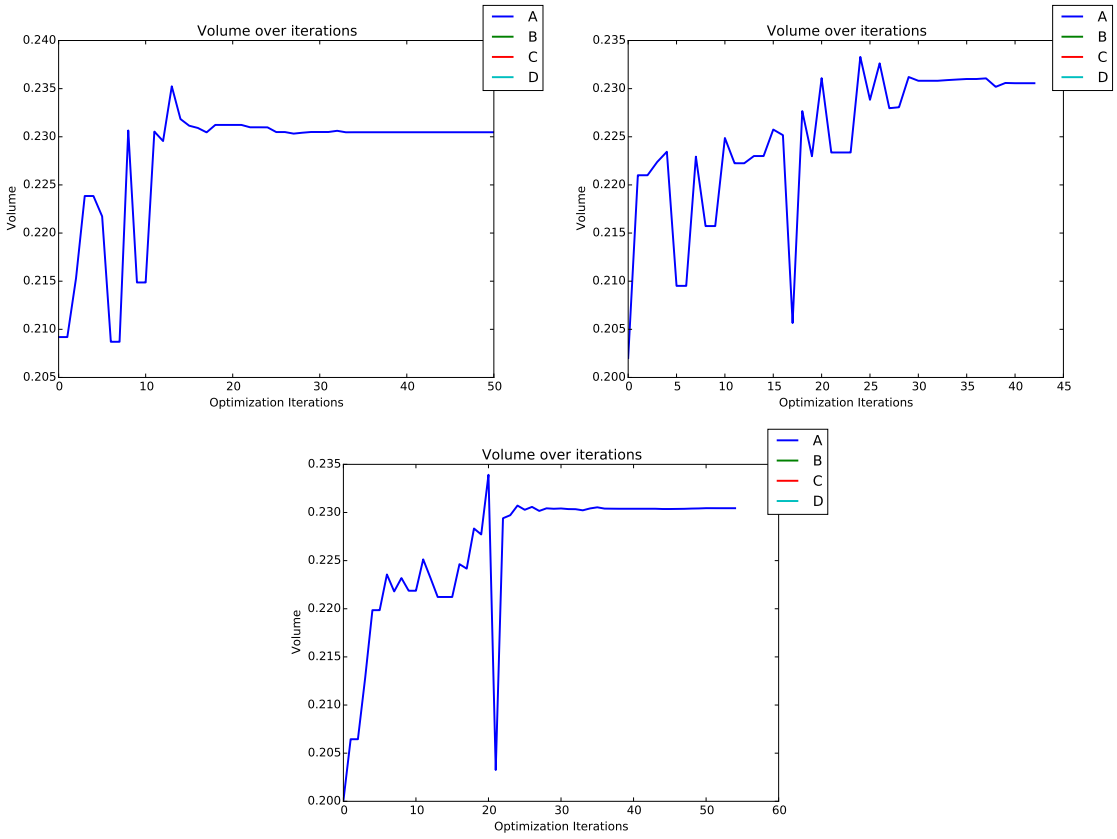

Figure 4.6. These figures shows the volume over a range of mesh sizes, from left to right and top to bottom 40×40, 80×80 and 160×160.

defines the stable configuration of a fluid composed of multiple density distributions with minimal surface area interfaces between the densities. We paraphrase the main result from this paper here:

Theorem 4.1. *Define $W : \mathbb{R} \to \mathbb{R}$ to be a continuous, positive function with two minima at $\alpha < \beta$ satisfying $W(\alpha) = W(\beta) = 0$. Further, let $m \in \mathbb{R}$ satisfy $\alpha|\Omega| \leq m \leq \beta|\Omega|$. Now define a function $u_\varepsilon \in L^1(\Omega)$ that minimizes*

$$\begin{aligned} \underset{u_\varepsilon}{\text{minimize}} \quad & \int_\Omega \varepsilon \nabla u_\varepsilon(x) \cdot \nabla u_\varepsilon(x) + W(u_\varepsilon(x))\, dx \\ \text{subject to} \quad & \int_\Omega u_\varepsilon(x)\, dx = m,\ u(x) \geq 0. \end{aligned}$$

Then as $\varepsilon \to 0^+$, the sequence u_ε converges to $u_0 \in L^1(\Omega)$ satisfying:

1. *$u_0(x) = \alpha$ or $u_0(x) = \beta$ for almost all $x \in \Omega$,*
2. *the interface between the sets $x : u_0(x) = \alpha$ and $x : u_0(x) = \beta$ has minimal area.*

Proof. See [29]. □

Motivated by this result we will add a regularization term to the objective function used in the previous formulation. Because, our topology optimization formulation (and algorithm) enforces that the control is between $[0,1]$, and SIMP enforces the digital nature of the solution, we drop the penalization term from Theorem 4.1 (e.g. the $W(x)$) and focus on the regularization to produce the minimal interface. Thus the second two wire formulation is:

$$\min_{z \in V} \quad I(\Gamma_C, z) - I(\Gamma_D, z) + \frac{w_z}{3} \max\left(\int_\Omega z\, dx - v_f|\Omega|, 0 \right)^3 + \int_\Omega \varepsilon \nabla z \cdot \nabla z \tag{4.39}$$

subject to Eqs. 4.22-4.28. The regularization term $\int_\Omega \varepsilon \nabla z \cdot \nabla z$ is referred to below as the Ginzburg-Landau energy (this is slightly incorrect as this typically includes the 0-1 penalization term implied by the W we have excluded).

Figure 4.7 shows a sequence of images for the optimization problem run with a fixed ε. Early in the optimization process the wires are bent, then straighten as the Ginzburg-Landau energy term begins to dominate the objective function. This minimizes the surface area between the conductor (red) and the void regions (blue). For this fixed value of ε the interface is more diffused then the original unregularized problem. Figures 4.8, 4.9, 4.10 show the convergence over several iterations of various figures of merit associated with this formulation. Notice that the currents (Fig. 4.8) behave similar to formulation 1. The conservation properties of this system appear to be much better with closer matching between input and output currents, and less overall loss. The volume (Fig. 4.9) converges to the target volume. Finally the images in Fig. 4.10 show the convergence of the Ginzburg-Landau energy. Clearly minimizing this values has an effect on the solution.

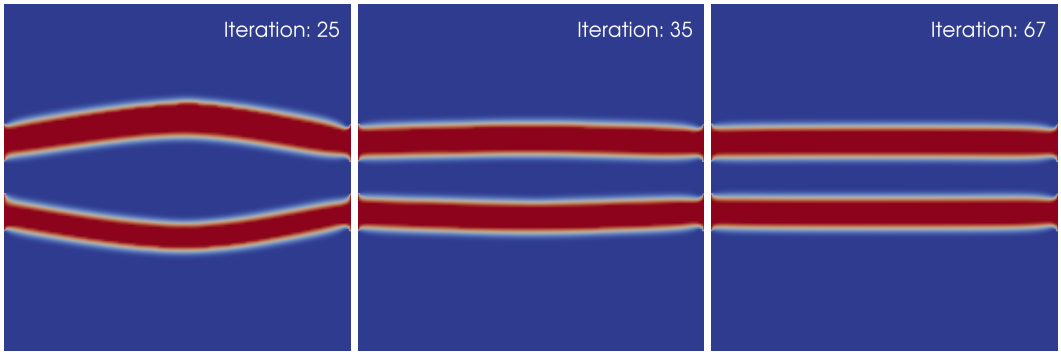

Figure 4.7. This figure plots the conductivity computed using formulation 2. Notice that the gaps that existed in formulation 1 are closed and the wires are no longer bent. However, the interface is more diffused. Each image is computed from a different optimization iteration count.

4.3 Electromagnetics

This section develops an approach for design of coaxial conducting networks. This builds on the brief discussion of the electromagnetic properties of the coaxial cable in Section 4.1.1. The spatial discretization used for all results developed here was discussed in detail in Section 4.1.2. The formulation developed is based on a transient backward Euler implementation of Maxwell's equation. This is used to design an inner MITL and compare to the theoretical results developed in [19].

4.3.1 Design Objective and Constraints

Maxwells equations discretized using compatible basis functions, as discussed in Section 4.1, will be used as the PDE constraint. Ohm's law will be used to defined the current such that

$$\mathbf{J} = \sigma(z)\mathbf{E} \tag{4.40}$$

where the conductivity σ is dependent on the control parameter z. This will use a SIMP model

$$\sigma(z) = \sigma_{min} + (\sigma_{max} - \sigma_{min})z^3 \tag{4.41}$$

where $z \in [0,1]$ and $\sigma_{max}/\sigma_{min}$ is the maximum/minimum conductivity. For boundary conditions the outer wall of the coaxial cable uses a natural boundary condition (e.g. $\mathbf{n} \times \mathbf{B} = \mathbf{0}$). The boundary conditions on the outflowing surfaces on the top of the coaxial cable usie a low order absorbing boundary conditions (ABC) that enforces

(4.42)

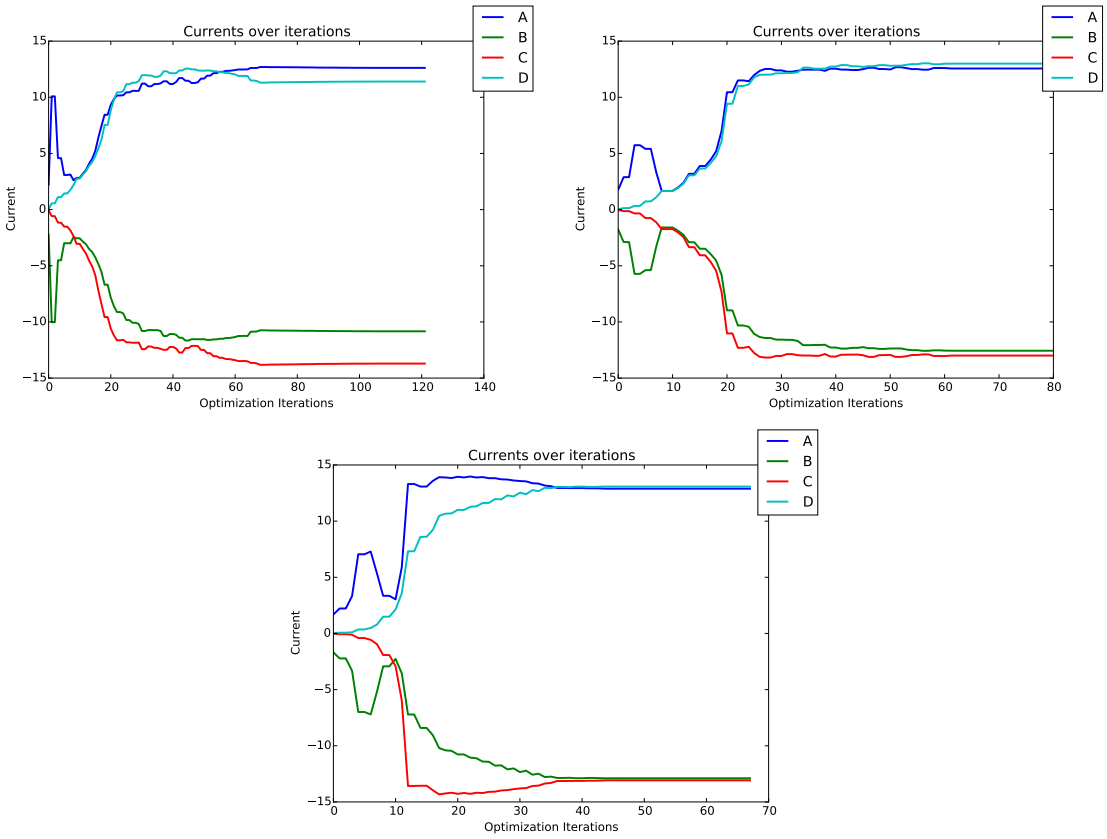

Figure 4.8. These figures show the current over a range of mesh sizes, from left to right and top to bottom 40×40, 80×80 and 160×160. (Formulation 2)

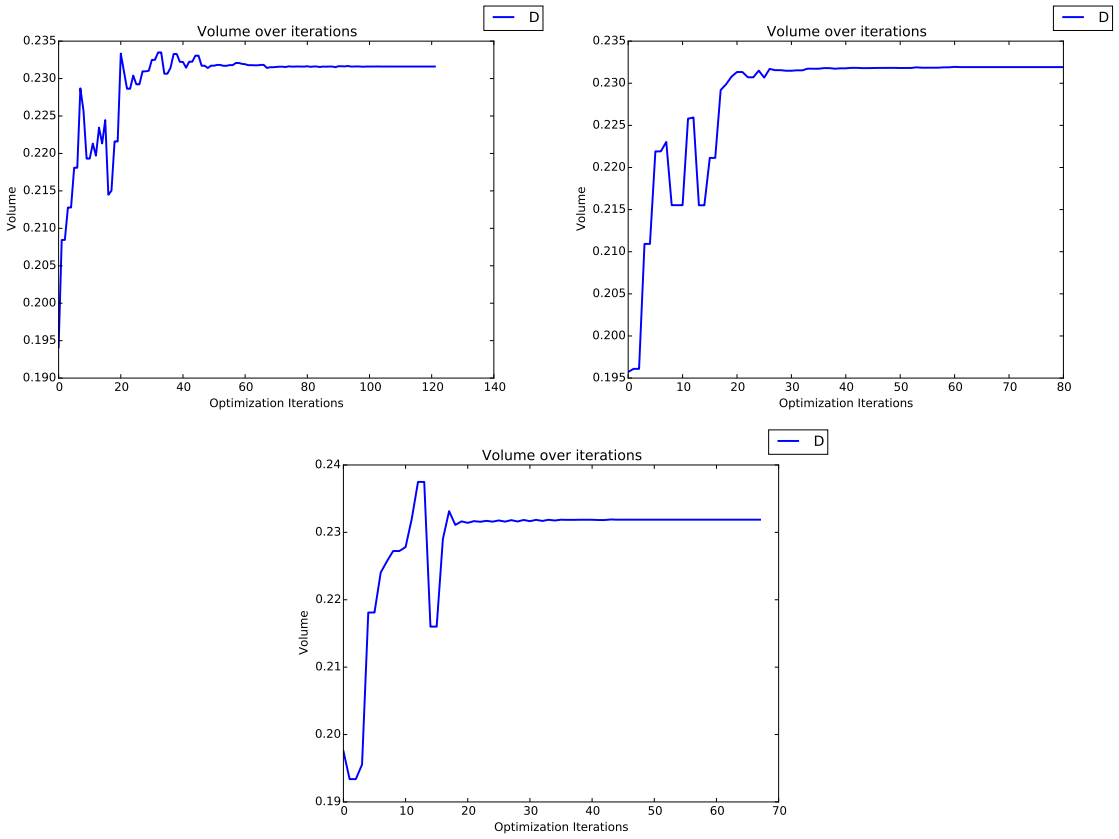

Figure 4.9. These figures show the volume over a range of mesh sizes, from left to right and top to bottom 40×40, 80×80 and 160×160. (Formulation 2)

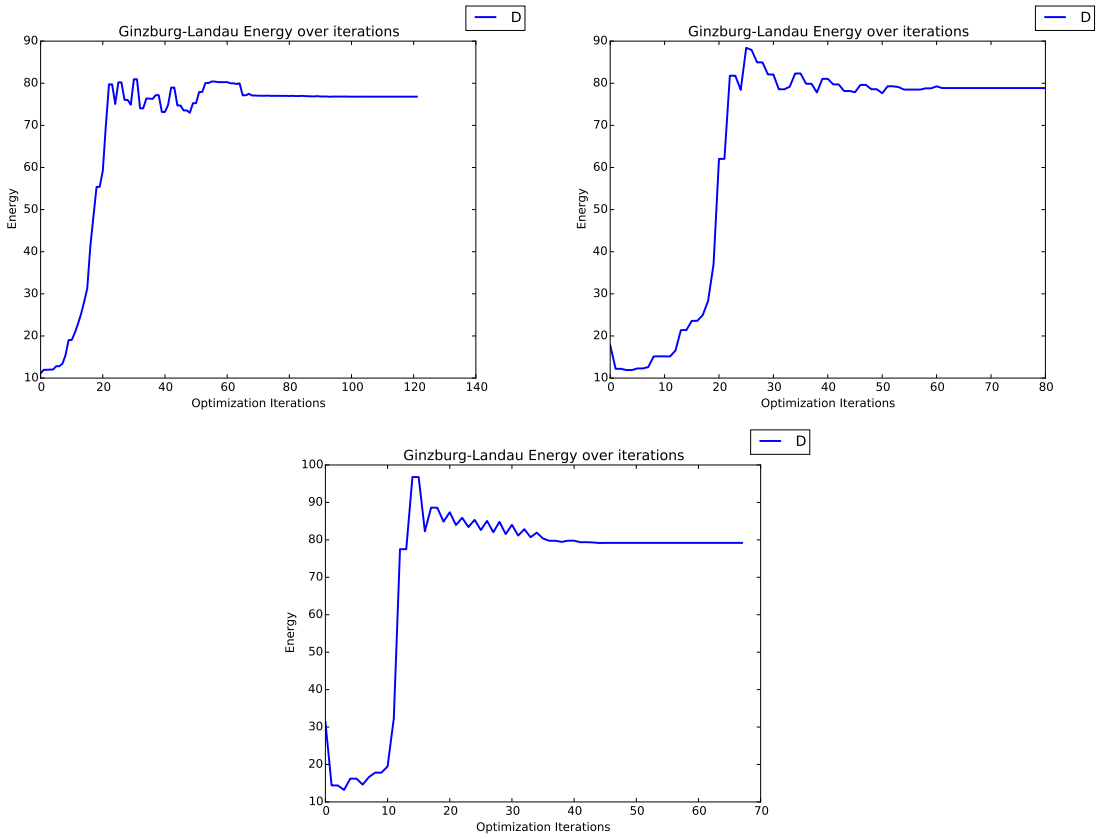

Figure 4.10. These figures show the Ginzburg-Landau energy over a range of mesh sizes, from left to right and top to bottom 40×40, 80×80 and 160×160. (Formulation 2)

See [7] for more details.

The base, or input port of the cable, is divided into three regions based on the distance from the center. Let $\mathbf{r}$ be the vector from the current point to the center of the coaxial cable. The distribution of conducting material on the base is defined as

$$\sigma(\mathbf{r}) = \begin{cases} \sigma_{max} & 0 \leq |\mathbf{r}| \leq r_0 \\ \sigma_{min} & r_0 < |\mathbf{r}| < r_1 \\ \sigma_{max} & r_1 \leq |\mathbf{r}| \leq r_2 \end{cases}. \tag{4.43}$$

Here $0 < r_0 < r_1 < r_2$ are the radii of each of three concentric circles on the base of the cable. In an ideal coaxial cable, this would define the nested cylinders. The boundary conditions on the input boundary are set to be

$$\mathbf{E} \cdot \mathbf{r} = \begin{cases} 0 & 0 \leq |\mathbf{r}| \leq r_0 \\ E_0/|\mathbf{r}| & r_0 < |\mathbf{r}| < r_1 \\ 0 & r_1 \leq |\mathbf{r}| \leq r_2 \end{cases}. \tag{4.44}$$

These boundary conditions are easily enforced strongly as Dirichlet conditions given the compatible $H_{\nabla\times}$ basis. Notice that this boundary condition implies that their is no electric field in the conductor. While in the gap, the electric field is radial. This enforces a fixed potential drop between the inner and outer conductors (see the Appendix A).

For optimization the objective function measures the power flow through the upper port of the coaxial cable. The layout of the conducting material is defined in the same way as the lower port. Define the 2D domain $\Gamma_{out} = (r, L_{top}) : r_1 \leq r \leq r_2$ where L_{top} is the vertical location of the top port. Then the objective that is used for optimization is the power flow through the gap out of the domain:

$$\int_{\Gamma_{out}} \mu_0^{-1} (\mathbf{E} \times \mathbf{B}) \cdot \mathbf{n} \, ds. \tag{4.45}$$

The goal of the optimization is to maximize this quantity, attempting to ensure that the power flowing through the cable is maximized and minimal energy is lost to flow through the boundaries or Ohmic heating.

In the text that follows many images will use slices through the coaxial cable to display the placement of conducting material. See Fig. 4.11 for a visual depiction of the process on an idealized coaxial cable. The conducting material is clearly placed in nested cylinders with a void between the two cylinders. In the 2D colormap, this is shown as a two vertical void regions (blue) and three vertical conducting regions (red).

4.3.2 Transient Design

The transient version of the design problem uses the backwards Euler method to evolve Maxwell's equations. As discussed above, the enforcement of the charge conservation (Gauss' laws) involutions is implied by the discretization. The optimization problem formulation maximizes the total

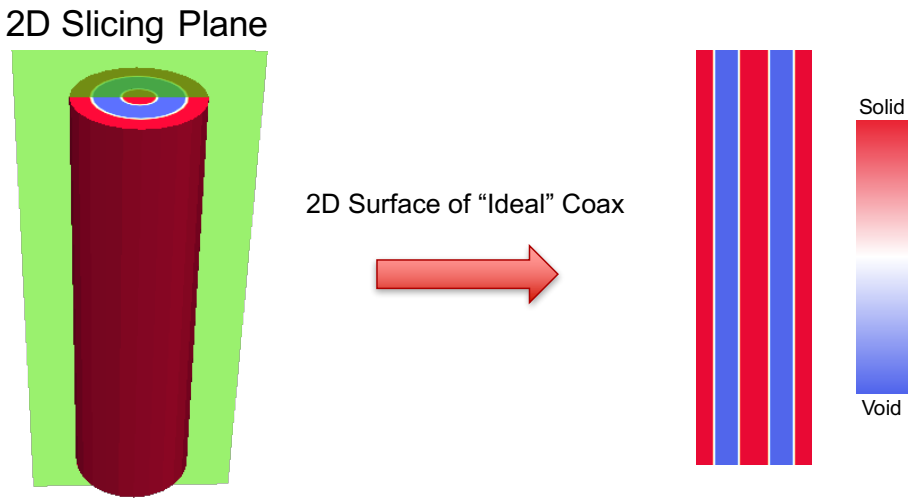

Figure 4.11. A visualization guide for the topology optimization results below. A 3D coaxial cable, with the colormap denoting conducting regions (red) and void regions (blue), is sliced through the middle. On the right is a 2D colormap displaying the placement of conducting material in the interior (along the slice) of the cable. This images is for an idealized coaxial cable.

power flow through the top port subject to satisfying Maxwell's equation.

$$\min_{z} \quad -\alpha_{power}\int_0^{t_f}\int_{\Gamma_{top}} \mathbf{n}\cdot(\mathbf{E}\times\mathbf{B})\,dxdt \tag{4.46}$$

subject to

$$\varepsilon_0\frac{\partial \mathbf{E}}{\partial t}+\sigma(z)\mathbf{E}=\frac{1}{s_B\mu_0}\nabla\times\mathbf{B} \tag{4.47}$$

$$\frac{\partial \mathbf{B}}{\partial t}+s_B\nabla\times\mathbf{E}=0 \tag{4.48}$$

$$\mathbf{n}\times\mathbf{B}|_{\Gamma_{side}}=0 \tag{4.49}$$

$$\mathbf{n}\times\mathbf{E}|_{\Gamma_{bottom}}=\mathbf{E}_{radial} \tag{4.50}$$

$$\mathbf{n}\times\mathbf{B}|_{\Gamma_{top}}=-\mathbf{E}_{tan} \tag{4.51}$$

$$\sigma(z)=\sigma_{min}+(\sigma_{max}-\sigma_{min})z^3 \tag{4.52}$$

Note the final time for the simulation t_f, and a scaling parameter for the objective, α_{power}. The radial boundary condition on the bottom port is specified to be zero in the conductor and radially outward in the gap between the conductors. Also, as of publication, we have not found that the minimal interface condition discussed in Sec. 4.2.2 to be required for these simulations. We consider three different modifications of this with respect to the volume of conducting material. The three cases are

Vol1 Volume is explicitly constrained to satisfy some fraction of the total volume

$$\int_\Omega z\,dx - V_f|\Omega| = 0, \tag{4.53}$$

Vol2 Volume contribution is ignored,

Vol3 Volume contribution is maximized by adding it to the objective,

$$-\alpha_{vol}\int_0^{t_f}\int_\Omega z\,dxdt \tag{4.54}$$

When presented below, the numerical optimization approach used is different for each case. The **Vol1** case is solved using an Augmented-Lagrangian approach. While the **Vol2** and **Vol3** cases are solved using a Kelley-Sachs trust region method (as there is no explicit volume constraint).

Regarding the total simulation time t_f. Some effort was spent to play with this parameter. Often if this was set too small, not enough electromagnetic energy would pass through the domain and poor unreasonable designs would result. In particular, Figure 4.12 shows a pseudo transient (one speed of light pass through) simulation where a single large time step is taken to design a straight coaxial cable. The designs used the **Vol1** enforcement of the volume constraint, thus an augmented Lagrangian approach is applied. Two different starting guesses are chosen, one a uniform conductivity, and the second an oscillating conductivity. The figure shows the two

resulting designs. The design on the left gives a reasonable coaxial cable design. The inner/outer conductor on the bottom connecting to the inner/outer conductor on the top. For the second design (with a uniform initial guess) the inner/outer conductor on the bottom connects to the outer/inner conductor on the top. A reversal of the conductors. This reversal is made apparent in the images showing the direction of the electric field on the top port. For the oscillating conductor the electric field on the top gap matches the electric field direction in the bottom gap, as expected for a coaxial cable. For the uniform initial guess, the electric field on the top of the gap is reversed from that which was set on the bottom! Thus in this case there are two local minima that are accessible that are both power maximizing (the magnetic field is reversed in the uniform case). This problem goes away if a longer transient optimization simulation is run.

If to large a value of t_f is chosen, the simulation takes an extraordinarily large amount of time to complete. This was due to a need to increase the number of time steps taken to reduce the preconditioning costs. In these simulations, it was often the case the preconditioners developed in [11] were not as effective due to the highly variable conductivity. The choice made for the design of the feed below were chosen to allow multiple $(3-4)$ speed of light pass throughs, while still maintaining a reasonable[1] optimization time.

The coaxial design problem considered here is to solve the two port (upper and lower) problem with two distinct radii. This is the case considered in the [19]. The goal is to try to replicate the structural properties of their analytic shape optimization using the topology optimization technology developed here. Two versions of the design are considered. The radii and heights defining the ports for the two problems are

Tall

$$r_0^{top} = 4\times 10^{-4},\ r_1^{top} = 8\times 10^{-4},\ r_2^{top} = 2\times 10^{-3} \tag{4.55}$$
$$r_0^{bottom} = 1.2\times 10^{-3},\ r_1^{bottom} = 1.6\times 10^{-3},\ r_2^{bottom} = 2\times 10^{-3} \tag{4.56}$$
$$z^{bottom} = 0.0,\ z^{top} = 2.5\times 10^{-3} \tag{4.57}$$
$$\text{Hurricane coordinates}: (R_{out}, Z_{out}) = ((0.0014, 0.0),\ (R_{in}, Z_{in}) = (0.0006, 0.0025) \tag{4.58}$$

Short

$$r_0^{top} = 4\times 10^{-4},\ r_1^{top} = 8\times 10^{-4},\ r_2^{top} = 3\times 10^{-3} \tag{4.59}$$
$$r_0^{bottom} = 2.2\times 10^{-3},\ r_1^{bottom} = 2.6\times 10^{-3},\ r_2^{bottom} = 3\times 10^{-3} \tag{4.60}$$
$$z^{bottom} = 0.0,\ z^{top} = 1.32\times 10^{-3} \tag{4.61}$$
$$\text{Hurricane coordinates}: (R_{out}, Z_{out}) = (0.0024, 0.0),\ (R_{in}, Z_{in}) = (0.0006, 0.00132) \tag{4.62}$$

In all cases the gap width of the incoming and outgoing ports are defined to be 4×10^{-4}. Below the

[1] For the smallest grids this was often on the order of $5-6$ hours on sixteen cores depending on the optimization algorithm

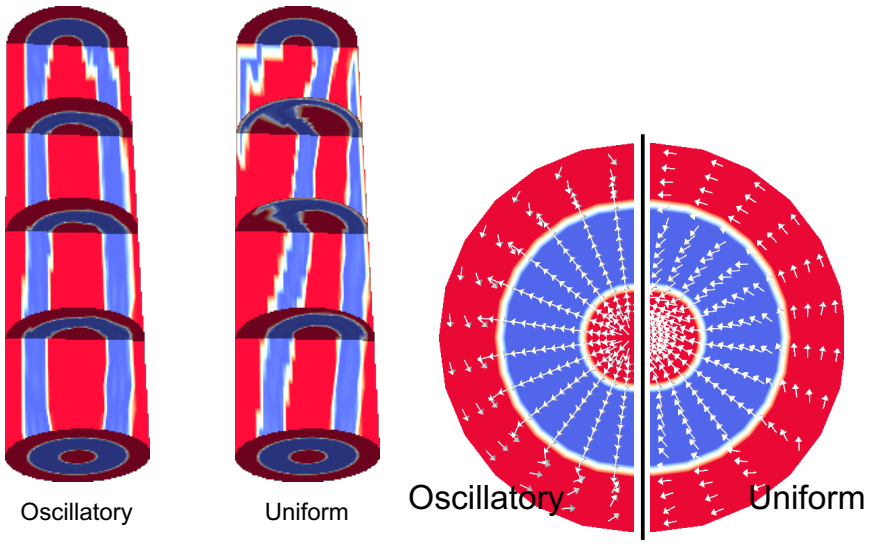

Figure 4.12. Two pseudo-transient designs of a straight coaxial cable. The designs differ only in the initial condition of the conductor, where one is uniform and the other is oscillatory. The time step is slightly longer then would be dictated by a single speed of light pass through. The image on the right shows the electric field on the top port of the conductor. Notice that the electric field for the uniform initial condition is reversed from the oscillatory case. This suggests that both designs maximize power, but in distinct local minima.

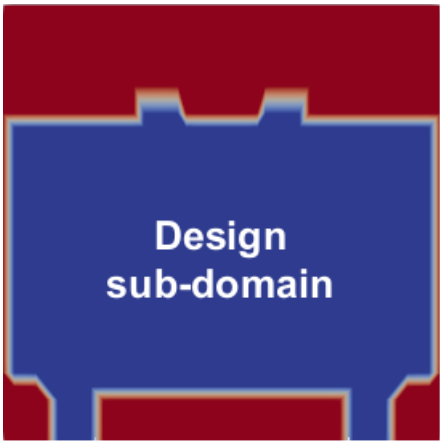

Figure 4.13. A picture of the domain used for the design of the varying radii feed. The lower portion represents a coaxial cable with radii $r_0^{bottom}, r_1^{bottom}, r_2^{bottom}$ and the input port. The upper portion represents a load, in this case a solid block of conductor with the port ending in $r_0^{top}, r_1^{top}, r_2^{top}$. The overall height of the design regions is specified as $z^{top} - z^{bottom}$.

design images and results will be labeled with the shape parameters (**Tall**, **Short**), and the volume constraint approach taken (**Vol1**,**Vol2**, **Vol3**).

The design problem is relegated to a subset of the domain, while the top and bottom provide for the incoming port and the outgoing load respectively. See Figure 4.13 for a description of the domain. On the lower portion, the electromagnetic fields are allowed to evolve in the coaxial geometry before the design portion of the domain. The upper portion of the domain is designed to absorb any energy and dissipate it in the conductor. In general, this could lead to reflections. Though for the choices made for physical simulation time t_f, this has not been problematic.

For both the **Tall** and **Short** inner MITL problems, the initial condition is a uniform guess for the conductivity corresponding to $z = 0.2$. Figure 4.14 presents 3D images of the initial condition and the final design. The value for the conductivity ranges from $\sigma_{min} = 10^{-3}$ to $\sigma_{max} = 10^3$. These values, like other terms, were chosen to minimize the stress on the preconditioner, while simultaneously providing enough separation between conductor and void for the optimization problem to progress. The simulation time is $t_f = 4.67128190104799 \times 10^{-11}$ which was chosen to allow the initial electromagnetic field to fully propogate and pass through the domain if it were unimpeded. A total of 200 backward Euler steps are taken to evolve the state equation. The magnetic field is scaled as $s_B = 3 \times 10^6$ (this was chosen so that the electric field's and magnetic field's magnitude is roughly the same order). Physical values of the permittivity and permeability are used; $\mu_0 = 1.25663706^{-6}$ and $8.85418782 \times 10^{-12}$. The input boundary condition for the bottom port is defined on the electric field. For regions where the conductivity is solid, there is no electric field. In the gap the electric field is ramped and then maintained as a constant throughout the remainder

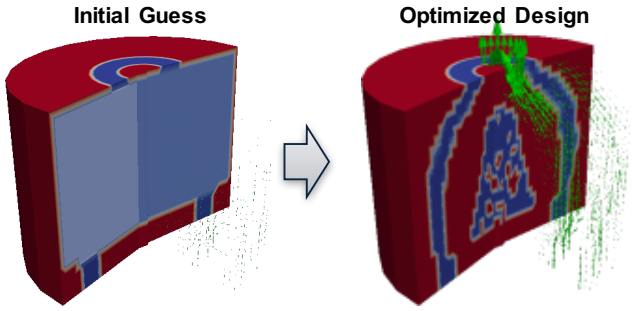

Figure 4.14. This image demonstrates the initial condition and end design computed for the inner MITL **Tall** problem. Red represents solid conductor, and blue represents void material. The current, depicted by the green arrows is evident in the final design. While the current is not flowing through the domain in the initial condition.

of the simulation. Based on a radial electric field the magnitude is specified as

$$E_r(t) = \min(1.0, (t/t_r)^p)E_0. \tag{4.63}$$

For the results described below $p = 2$, $t_r = 3.50346142578599 \times 10^{-12}$, and $E_0 = -1.091356267179026 \times 10^2$. The parameter scaling the objective is $\alpha_{power} = 10^{-16}$.

The design results after 40 iterations of the Kelley-Sachs algorithm using no volume contributions (**Vol2**) are presented in Figure 4.15. The ideal curve for the inner MITL derived in [19] overlays the colormap slice through the domain. These curves, for this coarse simulation, show qualitative agreement with the theory for both the **Tall** and **Short** problem. In the **Tall** case the topology optimization solution captures the initial increase in the radius of the inner feed.

One aspect of this result that is suboptimal is the placement of "void" in the center (this area is a bell shaped blue region in Figure 4.15). This is the result of using the **Vol2** formulation that places no restraint on the volume of conducting material. Note that this isn't particularly important for this electromagnetic design problem as little current or magnetic field is present in this part of the domain. So it neither harms nor helps the total power outflow. To demonstrate this, Figure 4.16 shows the same design problem run with the **Vol3** formulation that adds the volume to the objective function. The volume contribution is scaled by $\alpha_{vol} = 2.14073999639297 \times 10^{20}$. This value is chosen to ensure the two terms of the objective are roughly balanced. Figure 4.16 shows the resulting design for this optimization problem. The design is qualitatively the same as the design in Figure 4.15. The primary difference is there is less of the void region in the center of the domain. Quantitatively the power output of **Tall**-**Vol2** is 2.378×10^{-9}, versus 2.309×10^{-9} for **Tall**-**Vol3**. This deviation is attributed to the magnitude of the volume term in the objective overtaking the power contribution as the design is refined.

Figure— 4.17 shows a 2D slice of the **Tall** design problem optimized using **Vol1**. In this case

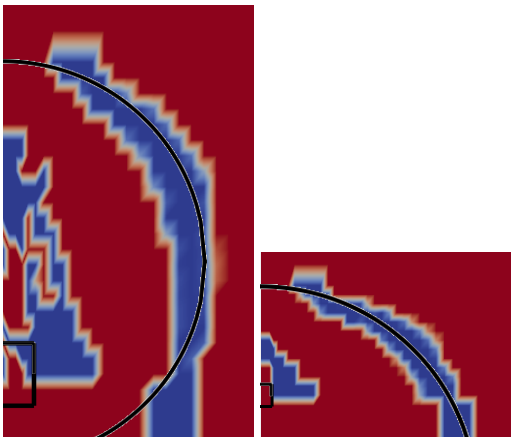

Figure 4.15. Two dimensional slices through the 3D inner MITL domain, run with the **Vol2** formulation. The theoretical result for the ideal shape of the inner MITL is overlaid on the images. Notice there is good qualitative agreement for both the **Tall** (left) and **Short** (right) problem.

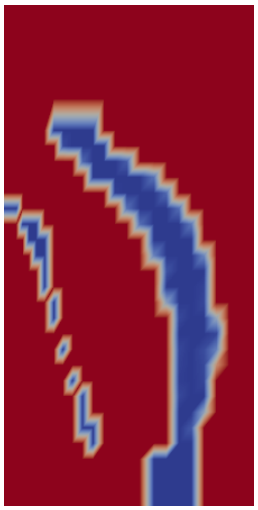

Figure 4.16. This image shows the result of the **Tall** problem run using **Vol3** volume enforcement (compare to the left image in Figure 4.15). Notice a good inner MITL design is developed, though the placement of arbitrary void in the center is reduced as compared to the **Tall**-**Vol2** case.

Figure 4.17. This image shows the result of the **Tall** problem run using **Vol1** volume enforcement with coaxial output port used instead of conducting block of conducting material (compare to the left image in Figure 4.15). In this case the optimization algorithm was not able to find a good inner MITL design.

instead of a solid block of conductor in the upper domain, a coaxial output on the upper port is used. This leads to a different impedance (as the load is only defined by the absorbing boundary condition on the upper boundary) then the solid conductor which dissipates the electromagnetic energy. This design is particularly poor and does not even create a complete circuit. The effect of the load on the design is an interesting avenue for future exploration.

This page intentionally left blank.

Chapter 5

Conclusion

Computational design optimization is an increasingly important tool for many aspects of the Sandia mission space. This LDRD proposed to extend these ideas into the challenging area associated with design of components of the Z-machine. As a result of this ambitious goal a number of critical capabilities were implemented, methods developed, and formulations pursued. These techniques will lead to improvements in performance, robustness and extend the applicability of existing approaches. In addition, this effort applied these approaches to a range of physical models including linear elasticity, electrostatics, and electromagnetics. The key results of this project include:

- A number of important software advances were made through the efforts of this project. A sequence of new optimization methods were incorporated into the **ROL** library available in the **Trilinos** framework. This capability is being applied and used by other Sandia programs including the ASC program, CRADA partners, and DARPA projects. In addition, we used the **Sacado** automatic differentiation library to implement first and second derivatives in the **Drekar** multi-physics application. Finally, we developed a runtime calculator that uses **Sacado** to automatically differentiate the objective function.
- We developed a pair one dimensional discrete beam models to explore the effect of topology optimization formulations on the performance of various optimization algorithms. We were interested in understanding how the performance (measured in terms of iterations) is effected as the number of cells increases. A critical outcome of this effort was to determine that Newton-based optimization methods can achieve mesh independence, while the gradient based approaches are not algorithmically scalable.
- A new scalable approach to structural design using topology optimization was developed. This involved formulating the optimization problem using volume minimization with an inequality constraint on the compliance. To ensure the solution is mesh independent, a filtering algorithm is applied to the design variable. To solve this problem an augmented Lagrangian trust region approach is applied. With this formulation, the iterations required by the algorithm are independent of mesh size.
- A topology optimization formulation constrained by an electrostatic model was developed to design conducting networks. This formulation was used to design a four contact conducting network.
- A transient electromagnetic design capability was implemented. This used a compatible basis discretization of Maxwell's equations and novel preconditioning technology to allow

efficient implicit simulation. A SIMP based topology optimization formulation was implemented and used to design an inner MITL. This result compared favorably to a theoretical development to design the same device.

5.1 Next Steps

This project provided a topology optimization foundation for design of electromagnetic conducting networks. However, a number of activities should be pursued in extension of this work. For instance, the H_{∇} discretization of the control used in the electromagnetics problem seems suboptimal. In particular, the current typically lives on the surface of the conductor. The SIMP formulation does not do a good job providing the sharp interfaces required. In general other approaches beyond SIMP and a density based method should be considered. To increase the value of the modeling, improving the electromagnetic model and working to justify or extend the Ohm's law approach considered above should be pursued. Finally, considering the scalability and applicability of the optimization approaches there are two paths forward. First new algorithms could be developed that take into account the structure used by the SIMP method. Second, developing density based or other topology formulations that are more amenable to existing optimization algorithms.

References

[1] M. P. Bendsoe and O. Sigmund. *Topology optimization: theory, methods and applications*. Springer, 2003.

[2] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York,. London, Paris, San Diego, San Francisco, 1982.

[3] José A Bittencourt. *Fundamentals of plasma physics*. Springer Science & Business Media, 2013.

[4] P. Bochev, H. C. Edwards, R. C. Kirby, K. Peterson, and D. Ridzal. Solving pdes with intrepid. *Sci. Program.*, 20(2):151–180, April 2012.

[5] Pavel Bochev, H Carter Edwards, Robert C Kirby, Kara Peterson, and Denis Ridzal. Solving pdes with intrepid. *Scientific Programming*, 20(2):151–180, 2012.

[6] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. *Knitro: An Integrated Package for Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006.

[7] Gary C Cohen and Sébastien Pernet. *Finite element and discontinuous Galerkin methods for transient wave equations*. Springer, 2017.

[8] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[9] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *LANCELOT: A FORTRAN package for large scale nonlinear optimization with simple bounds*. Springer Series in Computational Mathematics, Vol. 17. Springer Verlag, Berlin, Heidelberg, New York, 1992.

[10] E.C. Cyr, J.N. Shadid, and R.S. Tuminaro. Teko: A block preconditioning capability with concrete example applications in navier–stokes and mhd. *SIAM Journal on Scientific Computing*, 38(5):S307–S331, 2016.

[11] E. G. Phillips, J. N. Shadid, and E. C. Cyr. Scalable Preconditioners for Structure Preserving DIscretizations of Maxwell Equations in First Order Form. *submitted to SISC*, 2017.

[12] E.C. Cyr, M. Hoemmen, R.P. Pawlowski, B. Seefeldt. Parallel Unknown Numbering for the Finite-Element Method. Technical report, in preparation, 2017.

[13] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns*. Addison-Wesley Professional, 1995.

[14] Johan Peter Goedbloed and Stefaan Poedts. *Principles of magnetohydrodynamics: with applications to laboratory and astrophysical plasmas*. Cambridge university press, 2004.

[15] Matthew Robert Gomez, Michael Edward Cuneo, Ryan D McBride, Gregory Alan Rochau, DJ Ampleford, JE Bailey, AD Edens, B Jones, M Jones, MR Lopez, et al. Spectroscopic measurements in the post-hole convolute on sandia's z-machine. In *Pulsed Power Conference (PPC), 2011 IEEE*, pages 688–695. IEEE, 2011.

[16] MR Gomez, RM Gilgenbach, ME Cuneo, CA Jennings, RD McBride, EM Waisman, BT Hutsel, WA Stygar, DV Rose, and Y Maron. Experimental study of current loss and plasma formation in the z machine post-hole convolute. *Physical Review Accelerators and Beams*, 20(1):010401, 2017.

[17] W. Hock and K. Schittkowski. Test examples for nonlinear programming codes. *Journal of Optimization Theory and Applications*, 30(1):127–129, Jan 1980.

[18] Ronald HW Hoppe, Svetozara I Petrova, and V Schulz. Primal-dual newton-type interior-point method for topology optimization. *Journal of Optimization Theory and Applications*, 114(3):545–571, 2002.

[19] OA Hurricane. Optimized minimal inductance transmission line configuration for z-pinch experiments. *Journal of applied physics*, 95(8):4503–4505, 2004.

[20] K. Ito and K. Kunisch. Augmented lagrangian methods for nonsmooth, convex optimization in hilbert spaces. *Nonlinear Analysis: Theory, Methods & Applications*, 41(5):591 – 616, 2000.

[21] C.T. Kelley and E.W. Sachs. Truncated newton methods for optimization with inaccurate functions and gradients. *Journal of Optimization Theory and Applications*, 116(1):83–98, Jan 2003.

[22] G. Kiziltas, D. Psychoudakis, J. L. Volakis, and N. Kikuchi. Topology design optimization of dielectric substrates for bandwidth improvement of a patch antenna. *Antennas and Propagation, IEEE Transactions on*, 51(10):2732–2743, 2003.

[23] Michal Kočvara and Sudaba Mohammed. Primal-dual interior point multigrid method for topology optimization. *SIAM Journal on Scientific Computing*, 38(5):B685–B709, 2016.

[24] Sandia National Laboratories. Z Pulsed Power Facility: How Does the Z Machine Work? `http://www.sandia.gov/z-machine/about_z/how-z-works.html`, 2017. Accessed: 2017-08-29.

[25] B. S. Lazarov, F. Wang, and O. Sigmund. Length scale and manufacturability in density-based topology optimization. *Archive of Applied Mechanics*, 86(1):189–218, 2016.

[26] KR LeChien, WA Stygar, ME Savage, PE Wakeland, V Anaya, MJ Baremore, DE Bliss, R Chavez, GD Coombs, JP Corley, et al. 6.1-mv, 0.79-ma laser-triggered gas switch for multimodule, multiterawatt pulsed-power accelerators. *Physical Review Special Topics-Accelerators and Beams*, 13(3):030401, 2010.

[27] M. Hajghassem and E.C. Cyr and D. Ridzal. A time-parallel method for the solution of PDE-constrained optimization problems. In *CSRI Summer Proceedings 2015*, 2015.

[28] E. A. Madrid, D. V. Rose, D. R. Welch, R. E. Clark, C. B. Mostrom, W. A. Stygar, M. E. Cuneo, M. R. Gomez, T. P. Hughes, T. D. Pointon, et al. Steady-state modeling of current loss in a post-hole convolute driven by high power magnetically insulated transmission lines. *Physical Review Special Topics-Accelerators and Beams*, 16(12):120401, 2013.

[29] L. Modica. The gradient theory of phase transitions and the minimal interface criterion. *Archive for Rational Mechanics and Analysis*, 98(2):123–142, 1987.

[30] N.L. Fisher and E.C. Cyr. Orthogonal spline collocation for pde constrained optimization. In *CSRI Summer Proceedings 2016*, 2016.

[31] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, Berlin, Heidelberg, New York, second edition, 2006.

[32] Roger P Pawlowski, Eric T Phipps, and Andrew G Salinger. Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, part i: Template-based generic programming. *Scientific Programming*, 20(2):197–219, 2012.

[33] Roger P Pawlowski, Eric T Phipps, Andrew G Salinger, Steven J Owen, Christopher M Siefert, and Matthew L Staten. Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, part ii: Application to partial differential equations. *Scientific Programming*, 20(3):327–345, 2012.

[34] Susana Rojas-Labanda and Mathias Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization*, 52(3):527–547, Sep 2015.

[35] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[36] Robert J. Vanderbei. Loqo:an interior point code for quadratic programming. *Optimization Methods and Software*, 11(1-4):451–484, 1999.

[37] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.

[38] Eduardo M Waisman and ME Cuneo. Minimal inductance for axisymmetric transmission lines with radially dependent anode-cathode gap. *Physical Review Special Topics-Accelerators and Beams*, 12(9):090401, 2009.

[39] W.J. Rider, et al. ASC ATDM FY16 Level 2 Milestone. Technical Report SAND2016-9754, Sandia National Laboratories, 2016.

This page intentionally left blank.

Appendix A

Radial solutions to Gauss' law

A frequent activity for this project was to solve Gauss's law

$$\nabla \cdot \mathbf{E} = \frac{1}{\varepsilon_0}\rho \tag{A.1}$$

where ρ is the charge density and $\mathbf{E}$ is the electric field) in cylindrical coordinates. Introducing the potential ($\mathbf{E} = -\nabla\phi$) gives In this case the potential satisfies

$$-\nabla \cdot \nabla\phi = \frac{1}{\varepsilon_0}\rho. \tag{A.2}$$

Assuming symmetry in the z and θ directions, then we can rewrite Gauss' law as

$$-\frac{1}{r}\partial_r(r\partial_r\phi(r)) = \frac{1}{\varepsilon_0}\rho \tag{A.3}$$

where r is the radius. If we have that $\rho = 0$ then the the potential is written as

$$\phi_0(r) = C_0 + C_1\log(r). \tag{A.4}$$

These can be verified by substituting the expression for ϕ into Eq. A.3. Thus the electric field is proportional to r^{-1}.

DISTRIBUTION:

1 MS 0899 Technical Library, 9536 (electronic copy)


Sandia National Laboratories