Objective-First Nanophotonic Design Plan

Jesse Lu

December 31, 2012

Contents

1 Introduction **5**
- 1.1 Problem statement 5
- 1.2 Structure of our software 6

2 Theory overview **7**
- 2.1 Mathematically rigorous statement of the problem 7
- 2.2 Definition of physics residual 7
- 2.3 Bi-affine property of the physics residual 8
- 2.4 Definition of the field design objective 8
- 2.5 Choice of the structure design objective 8
- 2.6 Convexity analysis 9

3 The optimization module **11**
- 3.1 Capabilities of the module 11
- 3.2 Structure of the module 11
- 3.3 The paradigm submodule 12
  - 3.3.1 The local optimization paradigm 12
  - 3.3.2 The global optimization paradigm 14
- 3.4 The structure submodule 16
  - 3.4.1 The continuous structure update 16
  - 3.4.2 The discrete structure update 17

Chapter 1

Introduction

1.1 Problem statement

Our goal is to create a software package to enable the design of nanophotonic devices. Specifically, our software produces designs for *linear* nanophotonics devices based on *desired coupling strengths* between various input and output fields.

To put things mathematically, we want to create software to solve the following problem,

$$\begin{aligned} \text{minimize} \quad & f(x) + g(z) && \text{(1.1a)} \\ \text{subject to} \quad & A(z)x - b(z) = 0 && \text{(1.1b)} \end{aligned}$$

where

- x and z are the variables representing the *field* our device produces and the *structure* of the device respectively,
- $f(x)$ and $g(z)$ are our *design objectives*, which tell us the desirable properties that we would like our device to achieve, and
- $A(z)x - b(z)$ is the *physics residual*, the underlying physical laws which must be met.

In general, we need to consider multiple fields produced by the device. Our problem statement is then

$$\begin{aligned} \text{minimize} \quad & \sum_{i}^{N} f_i(x_i) + g(z) && \text{(1.2a)} \\ \text{subject to} \quad & A_i(z)x_i - b_i(z) = 0, \quad \text{for } i = 1, \ldots, N. && \text{(1.2b)} \end{aligned}$$

1.2 Structure of our software

Our software package consists of various modules which are designed to be maximally orthogonal. The various modules are illustrated in figure 1.1.

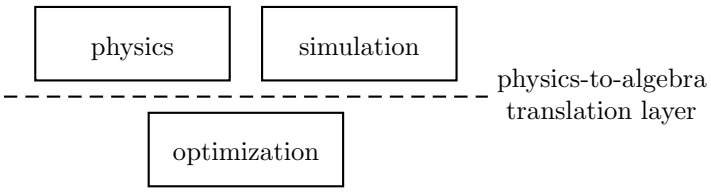

Figure 1.1: Structure of our software. The physics and simulation modules provide a complete description of the problem. The translation module takes this description and casts it in the language of linear algebra, which the optimization module can then use to produce various designs.

Although all modules are provided by our software, the majority of the innovation lies in the optimization module. Since the optimization module works entirely in the realm of linear algebra, a translation module is provided in order to cast concepts from electromagnetics into this form. Finally, physics and simulation modules are provided to allow the user to describe the design problem in physical, rather than mathematical terms.

Chapter 2

Theory overview

2.1 Mathematically rigorous statement of the problem

As previously stated, the problem we want to solve is

$$\text{minimize} \quad \sum_{i}^{N} f_i(x_i) + g(z) \tag{2.1a}$$

$$\text{subject to} \quad A_i(z)x_i - b_i(z) = 0, \quad \text{for } i = 1, \ldots, N. \tag{2.1b}$$

To now be more precise,

- $x_i \in \mathbf{C}^m$ are the field variables,
- $z \in \mathbf{C}^n$ is the structure variable,
- $f_i(x_i) \in \mathbf{C}^m \to \mathbf{R}$ are the field design objectives,
- $g(z) \in \mathbf{C}^n \to \mathbf{R}$ is the structure design objective,
- $A_i(z)x_i - b_i(z)$ are the physics residuals, with
- $A_i(z) \in \mathbf{C}^{n \times n}$ and
- $b_i(z) \in \mathbf{C}^n$.

2.2 Definition of physics residual

The physics residual, $A_i(z)x_i - b_i(z)$, corresponds to the electromagnetic wave equation

$$(\nabla \times \mu^{-1} \nabla \times -\omega^2 \epsilon) E = -i\omega J \tag{2.2}$$

via

- $\nabla \times \mu^{-1}\nabla \times -\omega^2\epsilon \to A_i(z)$,
- $\epsilon \to S_i z + \epsilon_{\text{const},i}$,
- $E \to x_i$, and
- $-i\omega J \to b_i(z)$, typically constant with respect to z.

2.3 Bi-affine property of the physics residual

Critically, (2.2) is not only linear in E, but is also affine in ϵ. This allows us to form the extremely useful relationship,

$$A_i(z)x_i - b_i(z) = B_i(x_i)z - d_i(x_i) = 0, \tag{2.3}$$

where $B(x_i) \in \mathbf{C}^{m\times n}$, $d(x_i) \in \mathbf{C}^n$ and

- $-\omega^2 E \to B_i(x_i)$,
- $\nabla \times \mu^{-1}\nabla \times E \to d_i(x_i)$.

2.4 Definition of the field design objective

Although the field design objective $f_i(x_i)$ can take on virtually any form, we choose to define it very specifically as

$$f_i(x_i) = \sum_j I_+(|c_{ij}^\dagger x_i| - \alpha_{ij}) + I_+(\beta_{ij} - |c_{ij}^\dagger x_i|), \tag{2.4}$$

where $c_{ij} \in \mathbf{C}^m$ and I_+ is the indicator function on nonnegative reals,

$$I_+(u) = \begin{cases} 0 & u \geq 0, \\ \infty & u < 0. \end{cases} \tag{2.5}$$

Such a design objective implements the constraints $\alpha_{ij} \leq |c_{ij}^\dagger x_i| \leq \beta_{ij}$ which can be interpreted physically as constraining the power emitted into the optical modes represented by c_{ij}.

2.5 Choice of the structure design objective

The form of the structure design objective consists of two parts,

- a structure parameterization function $m(p) \in \mathbf{R}^l \to \mathbf{C}^n$, and
- a parameter weighting function $w(p) \in \mathbf{R}^l \to \mathbf{R}$.

Both use the variable p which allows us to cast z into arbitrary parameters of our choice. Together, the general form of the structure design objective is

$$g(z) = I_0(z - m(p)) + w(p), \tag{2.6}$$

where I_0 is the indicator function on the set containing only 0,

$$I_0(u) = \begin{cases} 0 & u = 0, \\ \infty & \text{otherwise.} \end{cases} \tag{2.7}$$

An alternative interpretation of (2.6) is that it implements a hard constraint on z via the parameterization function $m(z)$, as well as a soft constraint on z via the weighting function $w(z)$.

Additionally, constraints on the elements of p include either a bounded continuous range of allowable values, or a finite set of discrete allowable values.

2.6 Convexity analysis

First we note that, as presented, (2.1) is non-convex in the variables x_i and z. Not only are the physics residuals $A_i(z)x_i - b_i(z)$ non-convex, but the field design objective, in that it implements the $\alpha_{ij} \leq |c_{ij}^{\dagger} x_i|$ constraint, is non-convex as well.

That our problem is non-convex means that it is fundamentally hard to solve because of the existence of multiple local minima. Additionally, even if we were to arrive at the global maxima, we would not have a straightforward way to verify global optimality. Lastly, fast convergence even to local minima may be difficult because methods such as Newton's method can not be directly applied to non-convex problems.

That said, (2.1) is *separably convex* in x_i and z if we ignore the non-convexity in the field design objectives, $f_i(x_i)$, and assume that the structure design objective, $g(z)$, is convex as well. This is given because of the bi-affine property of the physics residual, as shown in (2.3).

The separably convex, or *bi-convex*, properties of our problem open the door for the use of alternating direction algorithms, and specifically the alternating directions methods of multipliers (ADMM), for the implementation of a "global" optimization paradigm, Of course, in this context we do not use the term "global" rigorously, but only to differentiate it from strategies that rely purely on local information.

Lastly, since our problem is not bi-convex in the case of non-convex field or structure design objectives, simple extensions to alternating direction algorithms are employed.

Chapter 3

The optimization module

3.1 Capabilities of the module

We have implemented various *optimization paradigms* and *structure updates* which a user can arbitrarily combine in order to to solve (1.2).

Specifically, the user can choose to work in either a *local* or *global* optimization paradigm:

- the local paradigm uses the adjoint method to find small changes in the structure which will decrease the design objective;
- the global paradigm uses the objective-first method to arrive at a structure by forcing the design objective to be met from the start.

In addition, the user can choose between *continuous* or *discrete* structure updates:

- a continuous update constrains the values of p to be within a continous, bounded range; on the other hand,
- a discrete update constrains the values of p within a finite set of allowable values.

3.2 Structure of the module

The module is naturally separated into two submodules, the optimization paradigm (or simply, paradigm) submodule and the structure parameterization (or simply, structure) submodule. The interaction between these two submodules is illustrated in figure 3.1.

In essense, a design is achieved via repeated calls to the structure submodule by the paradigm submodule, in which carefully chosen quadratic functions $Q(z)$ are minimized.

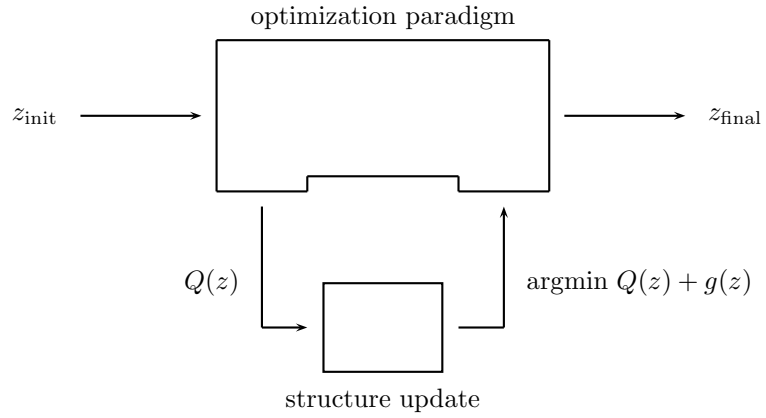

Figure 3.1: Basic layout of the module. The optimization paradigm submodule accepts an initial structure z_{init} and returns a final, optimized structure z_{final}. This is accomplished by repeatedly passing quadratic functions $Q(z)$ to the structure update submodule, which returns an updated structure z which minimizes $Q(z) + g(z)$.

3.3 The paradigm submodule

3.3.1 The local optimization paradigm

The local optimization paradigm applies the adjoint method to solve (2.1) in a manner analogous to the steepest-descent strategy. Specifically, we compute the derivative of the total field design objective with respect to z,

$$\nabla_z F = \sum_i^N \nabla_z f_i(x_i), \tag{3.1}$$

and then pass

$$Q(z) = \frac{1}{2}\|z - z_0\|^2 + \kappa \nabla_z F^\dagger (z - z_0) \tag{3.2}$$

to the structure submodule, for some κ.

The analogy with a steepest-descent strategy is apparent since the global minimum of $Q(z)$ can be found via

$$\nabla_z Q(z) = (z - z_0) + \kappa \nabla F = 0, \tag{3.3}$$

resulting in

$$z = z_0 - \kappa \nabla_z F, \tag{3.4}$$

from which we see that the role of κ is to determine the step-size in the direction of steepest-descent.

Computation of $\nabla_z F$

The derivatives of the individual field design objectives are found via

$$\frac{d}{dz} f_i(x_i) = \frac{\partial f_i}{\partial x_i} \frac{dx_i}{dz}. \tag{3.5}$$

To obtain dx_i/dz we first differentiate the corresponding physics residual, $A_i(z)x_i - b_i(z)$,

$$A_i(z)dx_i + dA_i(z) - db_i(z) = A_i(z)dx_i + B_i(x_i)dz, \tag{3.6}$$

where $dA_i(z) - db_i(z) = B_i(x_i)dz$ as a result of the bi-affine property of the physics residual (2.3).

Assuming that physics is already satisfied, $A_i(z)x_i - b_i(z) = 0$, and that we want to keep the physics residual at 0, the following condition must then be satisfied,

$$A_i(z)dx_i = -B_i(x_i)dz \tag{3.7}$$

from which we obtain

$$\frac{dx}{dz} = -A_i(z)^{-1} B_i(x_i). \tag{3.8}$$

Efficient calculation of dx_i/dz is almost always impossible since $B_i(x_i) \in \mathbf{C}^{n \times n}$. At the same time, since $\partial f_i/\partial x_i \in \mathbf{C}^{1 \times m}$, we instead compute $df_i(x_i)/dz$ via

$$\frac{d}{dz} f_i(x_i) = -\frac{\partial f_i}{\partial x_i} A_i(z)^{-1} B_i(x_i) = -\left(A_i(z)^{-\dagger} \frac{\partial f_i}{\partial x_i}^{\dagger} \right)^{\dagger} B_i(x_i) \tag{3.9}$$

which requires only one solve of $A_i^{\dagger}$ as opposed to n solves of A_i.

Computation of $\partial f_i(x_i)/\partial x_i$

The definition of $f_i(x_i)$ as given in (2.4) is not differentiable since any deviation away from the power constraints results in $f_i = \infty$. In order to make the constraints differentiable, then, we use the following relaxed indicator function I_+^{rel} in place of I_+,

$$I_+^{\text{rel}}(u) = \begin{cases} 0 & u \geq 0, \\ \frac{1}{a}|u|^p & u < 0, \end{cases} \tag{3.10}$$

where a is a normalization factor and $p \in (0, \infty]$. In order to guarantee a well-defined value of $\partial f_i(x_i)/\partial x_i$ even for $p = \infty$, we can let $a = \max_i f_i(x_i)$.

3.3.2 The global optimization paradigm

In contrast to the local paradigm, the global optimization paradigm utilizes an objective-first strategy in the design process. Although such a strategy is not technically global, it is differentiated from local strategies because it strictly enforces all design objectives, even at the expense of non-zero physics residuals.

The optimization paradigm utilizes the alternating directions method of multipliers[1] (ADMM) optimization technique, which casts (2.1) in terms of the following augmented Lagrangians,

$$\mathcal{L}(x,z,y) = \sum_i^N f_i(x_i) + \frac{\rho}{2}\|A_i(z)x_i - b_i(z)\|^2 + \mathbf{real}\left[y_i^\dagger(A_i(z)x_i - b_i(z))\right] + g(z), \tag{3.11}$$

where ρ is a constant scalar and y_i are dual variables. If we introduce $u_i = y_i/\rho$, then

$$\mathcal{L}(x,z,u) = \sum_i^N f_i(x_i) + \frac{\rho}{2}\|A_i(z)x_i - b_i(z) + u_i\|^2 - \frac{\rho}{2}\|u_i\|^2 + g(z). \tag{3.12}$$

ADMM obtains the solutions for x and z via

$$x_i \leftarrow \text{argmin}_{\ x_i}\mathcal{L}(x,z,u), \quad i = 1,\ldots,N \tag{3.13a}$$
$$z \leftarrow \text{argmin}_{\ z}\mathcal{L}(x,z,u), \tag{3.13b}$$
$$u_i \leftarrow u_i + (A_i(x_i)z - b(x_i)), \quad i = 1,\ldots,N. \tag{3.13c}$$

To obtain the correct $Q(z)$ which should be passed to the structure submodule, we use (2.3) to write

$$\text{argmin}_{\ z}\mathcal{L} = \text{argmin}_{\ z} g(z) + \sum_i^N \frac{\rho}{2}\|B_i(x_i)z - d(x_i) + u_i\|^2 \tag{3.14}$$

which means that

$$Q(z) = \sum_i^N \frac{\rho}{2}\|B_i(x_i)z - d(x_i) + u_i\|^2 \tag{3.15}$$

Computation of $\text{argmin}_{\ x_i}\mathcal{L}$

Updating x_i, which is left to the paradigm submodule, involves solving $\text{argmin}_{\ x_i}\mathcal{L}(x,z,u)$ via Newton's method for all $i = 1,\ldots,N$. This can be accomplished via the gradient with respect to x_i

$$\nabla_{x_i}\mathcal{L}(x,z,u) = \nabla_{x_i} f_i(x_i) + \rho A_i(z)^\dagger(A_i(z)x_i - b_i(z) + u_i), \tag{3.16}$$

as well as the Hessian with respect to x_i

$$\nabla^2_{x_i}\mathcal{L}(x,z,u) = \nabla^2_{x_i} f_{x_i} + \rho A_i(z)^\dagger A_i(z), \tag{3.17}$$

which can be combined to determine the direction of a step in Newton's method,

$$\Delta x_i = -\nabla^2_{x_i}\mathcal{L}^{-1}\nabla_{x_i}\mathcal{L}. \tag{3.18}$$

Solving $\operatorname{argmin}_{x_i}\mathcal{L}(x,z,u)$ from an initial x_i thus proceeds by repeatedly computing Δx_i and then performing a line search along that direction, until some convergence criterion is met, typically

$$\frac{1}{2}\nabla_{x_i}\mathcal{L}^\dagger\nabla^2_{x_i}\mathcal{L}^{-1}\nabla_{x_i}\mathcal{L} \leq \text{error threshold}. \tag{3.19}$$

Relaxation of $f_i(x_i)$

In order to compute Δx_i we relax the design objectives, $f_i(x_i)$, to a differentiable, and convex form. Specifically, the upper bound constraint is relaxed via

$$I_+(\beta_{ij} - |c_{ij}^\dagger x_i|) \to -\frac{1}{t}\ln(\beta_{ij}^2 - \|c_{ij}^\dagger x_i\|^2), \tag{3.20}$$

where t is a real positive scalar. The lower bound constraint is relaxed via

$$I_+(|c_{ij}^\dagger x_i| - \alpha_{ij}) \to -\frac{1}{t}\ln(\mathbf{real}[e^{-i\phi_{ij}}c_{ij}^\dagger x_i] - \alpha_{ij}), \tag{3.21}$$

where $e^{-i\phi_{ij}}$ is a phase factor on $c_{ij}^\dagger x_i$. The convention of $\phi_{ij} = \text{angle}(c_{ij}^\dagger x_i)$ is often used to determine ϕ_{ij} before optimizing for x_i.

The gradient can now be obtained as

$$\nabla_{x_i} f_i(x_i) = \sum_j \frac{r_{ij}}{t} c_{ij} \tag{3.22}$$

where

$$r_{ij} = \frac{2c_{ij}^\dagger x_i}{\beta_{ij}^2 - \|c_{ij}^\dagger x_i\|^2} - \frac{e^{i\phi_{ij}}}{\mathbf{real}[e^{-i\phi_{ij}}c_{ij}^\dagger x_i] - \alpha_{ij}}, \tag{3.23}$$

and the Hessian as

$$\nabla^2_{x_i} f_i(x_i) = \sum_j \frac{s_{ij}}{t} c_{ij}c_{ij}^\dagger \tag{3.24}$$

where

$$s_{ij} = \frac{2}{\beta_{ij}^2 - \|c_{ij}^\dagger x_i\|^2} + \frac{4\|c_{ij}^\dagger x_i\|^2}{(\beta_{ij}^2 - \|c_{ij}^\dagger x_i\|^2)^2} + \frac{1}{(\mathbf{real}[e^{-i\phi_{ij}}c_{ij}^\dagger x_i] - \alpha_{ij})^2}. \tag{3.25}$$

Computation of Δx_i

We now show how Δx_i may be efficiently computed. First we introduce

$$\tilde{c}_i = \frac{1}{\rho t} \sum_j r_j A^{-\dagger} c_{ij} \tag{3.26}$$

in order to write

$$\nabla_{x_i} \mathcal{L}(x, z, u) = \rho A_i(z)^\dagger (A_i(z) x_i - b_i(z) + u_i + \tilde{c}_i). \tag{3.27}$$

For its part, the Hessian can be simplified using the matrix inversion lemma[1]. Coupled with the introduction of

$$\tilde{C}_i = A_i(z)^{-\dagger} \left[\sqrt{\tfrac{s_1}{\rho t}} c_{i1} \quad \sqrt{\tfrac{s_2}{\rho t}} c_{i1} \quad \cdots \quad \sqrt{\tfrac{s_{m_i}}{\rho t}} c_{i m_i} \right] \tag{3.28}$$

we can write

$$\nabla^2_{x_i} \mathcal{L}(x, z, u) = \rho A_i(z)^\dagger (I + \tilde{C}\tilde{C}^\dagger) A_i(z), \tag{3.29}$$

and more importantly

$$(\nabla^2_{x_i} \mathcal{L}(x, z, u))^{-1} = \frac{1}{\rho} A_i(z)^{-1} M A_i(z)^{-\dagger}, \tag{3.30}$$

where

$$M = (I + \tilde{C}\tilde{C}^\dagger)^{-1} = I - \tilde{C}(I + \tilde{C}^\dagger \tilde{C})^{-1} \tilde{C}^\dagger. \tag{3.31}$$

Finally, the expression for Δx_i is obtained as

$$\Delta x_i = -\nabla^2_{x_i} \mathcal{L}^{-1} \nabla_{x_i} \mathcal{L} = A_i(z)^{-1} M (A_i(z) x_i - b_i(z) + u_i + \tilde{c}_i). \tag{3.32}$$

3.4 The structure submodule

3.4.1 The continuous structure update

A continuous structure update scheme solves

$$\text{argmin } Q(z) + g(z) \tag{3.33}$$

by limiting the allowable values of p in (2.6) to a continuous range of real values.

If both $m(p)$ and $w(p)$ are linear, then (3.33) is convex and can be solved using CVX.

On the other hand, if this is not the case, then (3.33) is solved using a gradient-descent method.

[1] $(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$

3.4.2 The discrete structure update

A discrete structure update scheme solves (3.33) while limiting the range of p to a finite set of discrete values.

In the special case where both $m(p)$ and $w(p)$, as well as $Q(z)$ are all linear and diagonal, then (3.33) may be solved trivially. Naturally, this is a very unique case and requires some amount of care to set up on the user's part.

In general, (3.33) is solved using a greedy algorithm where the most beneficial change in a single element of p is taken until no beneficial steps remain.

Bibliography

[1] Boyd group ADMM paper