Nanophotonic Inverse Design

Jesse Lu

December 12, 2011

Contents

1 Introduction

Our goal is to produce a software package capable of designing virtually any nanophotonic device. The software must produce designs

- with exceptional device performance,
- which are easily manufacturable, and
- with computational efficiency.

Such a software package would be extremely useful in designing the components needed to guide light on a computer chip such as couplers, filters, absorbers, and multiplexors.

However, the scale of the problem as well as the difficulty of inverting the underlying wave equation have been major obstacles in producing successful design algorithms.

1.1 Problem statement

We formulate the design problem in the following way,

$$\text{minimize} \quad f(x) + g(p) \tag{1a}$$

$$\text{subject to} \quad r(x, p) = 0 \tag{1b}$$

where

- x is the field variable,
- p is the structure variable,
- $f(x)$ is the performance objective,
- $g(p)$ is the manufacturability objective,
- $f(x) + g(p)$ is generally referred to as the design objective, and
- $r(x, p)$ is the physics residual for which we use the time-harmonic electromagnetic wave equation,

$$r(x, p) = (\nabla \times \epsilon^{-1} \nabla \times -\mu\omega^2)H - \text{sources}. \tag{2}$$

1.2 Key insights

Generic nonlinear optimization routines are usually unable to solve (1), because there are an extremely large (millions) of variables, and because $r(x, p)$ often results in ill-conditioned matrices. For this reason, we need to take advantage of key features of the problem.

- $r(x, p)$ is separably affine (bi-affine) in x and p,

$$r(x, p) = A(p)x - b(p) = B(x)p - d(x), \tag{3}$$

  this allows us to form two simpler sub-problems.
- Simulators which compute $A(p)^{-1}z$ are available, where z is an arbitrary vector, even for very large systems (millions of variables).
- Solving $B(x)p - d(x) = 0$ is possible with generic software, because manufacturing processes severely limit the degrees of freedom of p (decreasing p to thousands of variables).

2 Adjoint method

The adjoint method is a steepest-descent method on the space $r(x, p) = 0$, and relies upon the following linear approximations of the design objective and the physics residual,

$$(f(x + \Delta x) + g(p + \Delta p)) - (f(x) + g(p)) \approx \nabla f^T \Delta x + \nabla g^T \Delta p \tag{4a}$$

$$r(x + \Delta x, p + \Delta p) - r(x, p) \approx \nabla_x r^T \Delta x + \nabla_p r^T \Delta p. \tag{4b}$$

Assuming a starting point already satisfying $r(x,p)=0$, we note that (4b) must equal zero, in order to keep the physics residual at zero. This allows us to form the following relationship between Δx and Δp,

$$A(p)\Delta x + B(x)\Delta p = A\Delta x + B\Delta p = 0 \tag{5a}$$

$$\Delta x = -A^{-1}B\Delta p, \tag{5b}$$

which allows us to write (4a) only in terms of Δp,

$$\nabla f^T \Delta x + \nabla g^T \Delta p = -(\nabla f^T A^{-1} B - \nabla g)\Delta p. \tag{6}$$

Thus, we see that the steepest-descent direction is

$$\Delta p = B^T A^{-T} \nabla f - \nabla g. \tag{7}$$

2.1 Computational cost

2.2 Multi-mode formulation

3 Alternating directions method of multipliers (ADMM)

3.1 Computational cost of ADMM

3.2 Multi-mode ADMM

A Constructing the relevant matrices and vectors

B Solving the matrices