Math 112C Project Outline

Kyle Spink

Topic:

An Introduction to Shape Optimization & Optimal Control of Systems Governed by Partial Differential Equations via the Adjoint Method

Part I—What is the Adjoint Method?/Necessary Background

I see this part as the primer for next one and want to build up a level of understanding starting with a relatively easy to grasp analogy such as this one.

Consider

$$\min_{S\in\Omega}\left\{C^T x\right\} \text{ s.t. } Ax = b(S)$$

Then, assuming A is invertible,

$$x = A^{-1}b(S) \Rightarrow C^T x = C^T A^{-1} b(S)$$

Rather than solving $Ax = b(S)$ after each change of S and updating $C^T x$ accordingly, one can pre-compute $C^T A^{-1}$ and optimize the original function by only varying S.

I think the point I want to drive home in this part is that the the order of operations plays a significant role in a problem's computational complexity.

Part II—The Adjoint Method for PDEs and a Steady-State Heat Equation Example:

This would be the bulk of my presentation/project. I see it serving the purpose of a) deriving the adjoint method for PDEs and bridging that to its linear algebra form, b) going through the problem setup, specific adjoint equation, and numerical implementation for optimal control of heat through a two dimensional body, and lastly c) acknowledging its generality to prime how the adjoint method is used for the airfoil and other shape optimization problems.

First, I plan to go through the following derivation: Consider an objective function $F(b_i)$, $i = 1, \ldots, N$ and constraint/state function $R(U_k(b_i)) \equiv 0$, $k = 1, \ldots, M$. (One example could be to extremize the lift to drag ratio of an airfoil subject to steady, inviscid flow). Then the total derivatives of both functions with respect to the design variables are given by

$$\frac{dF}{db_i} = \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{db_i} \qquad (1) \qquad\qquad \frac{dR}{db_i} = \frac{\partial R}{\partial b_i} + \frac{\partial R}{\partial U_k}\frac{dU_k}{db_i} = 0 \qquad (2)$$

The motivation is to isolate terms indexed by i from terms indexed by k. To do this, one appropriately address $\frac{dU_k}{db_i}$. As a result, the total derivatives of the object function with respect to the design variables can be equivalently expressed as

$$\frac{dF}{db_i} = \frac{\partial F}{\partial b_i} + \psi_k\frac{\partial R}{\partial b_i}$$

with

$$\frac{\partial F}{\partial U_k} + \psi_k\frac{\partial R}{\partial U_k} = 0 \qquad (3)$$

The value ψ_k is referred to as the adjoint, or costate variable. To see this equivalence, clearly from (2), $\frac{\partial R}{\partial b_i} = -\frac{\partial R}{\partial U_k}\frac{dU_k}{db_i}$, and from (3),

$$\begin{aligned}
\psi_k = -\frac{\left(\frac{\partial F}{\partial U_k}\right)}{\left(\frac{\partial R}{\partial U_k}\right)} \Rightarrow \frac{\partial F}{\partial b_i} + \psi_k\frac{\partial R}{\partial b_i}& \\
= \frac{\partial F}{\partial b_i} + \frac{\left(\frac{\partial F}{\partial U_k}\right)}{\left(\frac{\partial R}{\partial U_k}\right)}\cdot\frac{\partial R}{\partial U_k}\frac{dU_k}{db_i}& \\
= \frac{\partial F}{\partial b_i} + \frac{\partial F}{\partial U_k}\frac{dU_k}{db_i}&
\end{aligned}$$

as in (1).

The benefit of this derivation (which I initially had trouble grasping) is that while although both $\frac{\partial F}{\partial U_k}$, $\frac{\partial R}{\partial U_k}$ (and, as a result, the adjoint variable) change with each new iteration of the *set* of design variables b_i, the adjoint variable is only computed once per iteration (for any given b_i) and used to compute the total derivatives with respect to *all* design variables b_i, therefore significantly reducing the computational complexity.

Informally, from the Stanford lecture series titled A Crash-Course on the Adjoint Method for Aerodynamic Shape Optimization there are "efficient methods to obtain sensitivities of *many* functions with respect to a *few* design variables" as well as "efficient methods to obtain sensitivities of a *few* functions with respect to *many* design variables"—which the adjoint method falls under.

I like the outline in Adjoint Based Optimization Methods For Flow Problems where for $i = 1 : N$,

- Step 1: Solve objective equation system
- Step 2: Solve the adjoint equation system
- Step 3: Compute the sensitivity information
- Step 4: Update the geometry based on Step 3
- Step 5: Evaluate the cost function I_i
- Step 6: Proceed if $|I_i - I_{i-1}| < \epsilon$

The setup and numerical implementation for the specific example of optimal control of heat through a two-dimensional body is something I'm still in the process of researching. The general idea is for given boundary conditions (ambient temperature with a source), determining the optimal geometry of a heat sink to minimize the steady-state temperature.

If its too complicated to explain in my project (or for me to understand), I may resort to omitting the optimization method that I'll use although I do think the concept of the Hessian Matrix is something interesting in its own right so I'm open to suggestions on whether or not I should learn more about it and allocate time to explain its use in my presentation.

I plan on programming in Matlab. I might take advantage of the access I have to the school-wide computing cluster when running the program depending on its runtime.

Part III—Adjoint Method for Shape Optimization of an Airfoil:

Aimed to be roughly half in length compared to Parts I or II, as suggested in my project proposal, its better to go deep into a "simple" example (the steady-state heat equation) to fully understand it than superficially across a whole bunch of topics. The most I would plan on touching would be how the scheme is set up—not derived, and maybe some results of the optimized airfoil. On my own time/outside of the presentation, it might be worth compiling a side-by-side comparison of the adjoint method versus the direct method of optimization via shape derivatives for a given number of epochs.