COUPLING BOUNDARY ELEMENT METHODS WITH SHAPE OPTIMIZATION

DARIN PEETZ *

1. Introduction. Topology optimization is a rapidly growing topic in the field of mechanics. At its core, topology optimization seeks to optimize the form of a structure for a given purpose. The most common technique, solid isotropic material with penalization (SIMP) works on top of a finite element mesh to assign varying material densities within a domain. Coupled with an efficient filtering technique to give the structure a finite length scale (to ensure mesh independence) as well as remove numerical artifacts, it is a powerful and versatile method to freely design a structure. However, it is not without its drawbacks. In particular, the filtering technique leaves the structure with blurred and ill-defined edges between material and void. While some more complicated filters can be used to partially alleviate this blurring, it will always be a core flaw of the SIMP method.

Another vein of topology optimization forgoes the idea of fully designing the topology of the structure, and instead seeks to modify the boundaries of the structure directly. While not quite as versatile as SIMP and other true topology optimization methods, shape optimization nonetheless guarantees the resulting structure will be well-defined and therefore manufacturable. In many cases, simply optimizing the shape of a structure is enough to greatly improve efficiency, especially in the case where holes in the structure are added a priori and modified as part of the optimization.

Shape optimization offers another advantage in that the structure can be defined with much higher fidelity than a method built on standard finite elements, such as SIMP. Rather than forcing the structure to follow the inter-element boundaries in a mesh, shape optimization method gives full freedom to the optimizer in modifying the boundaries of a structure. Using NURBS basis functions to define the boundary provides even more freedom in that true curves and non-polynomial shapes can be represented accurately.

Given that these shape optimization techniques only need information on the boundaries, it makes sense to evaluate the governing differential equation with a method that operates on the boundary as well. Thus, boundary integral methods are well suited to work with shape optimization. Again coupled with NURBS basis functions, they provide an unparalleled capability to design smooth, well-defined shapes that simply cannot be constructed with other methods.

2. Boundary Integral Methods for Elasticity. For problems in elasticity, the response of a structure is governed by Navier's equations:

$$\sigma_{ij,j} + \rho b_i = \rho \ddot{u}_i \tag{1}$$

For the scope of this project, we will assume that no body forces are present and that the structure is in static equilibrium. As a result, the equation simplifies to:

$$\sigma_{ij,j} = \mathbf{0} \tag{2}$$

*peetz2@illinois.edu

In boundary integral form, this equation is usually written like so:

$$C_{ij}(\mathbf{x})u_j(\mathbf{x}) + \int_\Gamma T^*_{ij}(\mathbf{x},\mathbf{y})u_j(\mathbf{y})d\Gamma(\mathbf{y}) = \int_\Gamma U^*_{ij}(\mathbf{x},\mathbf{y})t_j(\mathbf{y})d\Gamma(\mathbf{y}) \tag{3}$$

where C_ij is δ_{ij} where the boundary is smooth (a closed form expression exists for non-smooth boundaries in 2D, but not in 3D), $\mathbf{x}$ corresponds to the target point, $\mathbf{y}$ corresponds to the source point, $u_j(\mathbf{y})$ is the displacement vector at $\mathbf{y}$, and $t_j(\mathbf{y})$ is the traction vector at $\mathbf{y}$. U_{ij} and T_{ij} are the fundamental displacement and fundamental traction kernels, respectively. In matrix form, these terms can be represented by the system

$$\mathbf{Hu} = \mathbf{Gt} \tag{4}$$

where $\mathbf{H} = \mathbf{C} + \hat{\mathbf{H}}$. For 2D plane strain problems the kernels are as follows[1]:

$$U^*_{ij}(\mathbf{x},\mathbf{y}) = \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\,\delta_{ij}\ln\frac{1}{r} + r_{,i}r_{,j}\right] \tag{5}$$

$$T^*_{ij}(\mathbf{x},\mathbf{y}) = -\frac{1}{4\pi(1-\nu)r}\left\{\frac{\partial r}{\partial \mathbf{n}}\left[(1-2\nu)\,\delta_{ij} + 2r_{,i}r_{,j}\right] + (1-2\nu)(n_i r_{,j} - n_j r_{,i})\right\} \tag{6}$$

Here $\mathbf{r} = \mathbf{y} - \mathbf{x}$, $r = \|\mathbf{r}\|$, $r_{,i} = \frac{y_i - x_i}{r}$, $\frac{\partial r}{\partial \mathbf{n}} = r_{,i}n_i$, ν is Poisson's ratio, and μ is the shear modulus.

By simple observation, both of these kernels exhibit singularities, the term $\ln\frac{1}{r}$ in the displacement kernel is a log singularity, and the coefficient on the front of the traction kernel has a $1/r$ singularity. The weak log singularity can be handled by transforming the equation and using logarithmic Gaussian quadrature. First note that $\ln(\frac{1}{r}) = -\ln(r)$. The singularity could be removed in the limit by dividing r by a term that goes to zero at the same rate. For example, if the target point is at the left end of an element, choose the divisor as $\hat{\xi} = \frac{\xi - \xi_L}{\xi_R - \xi_L}$, the parametric distance from the source point to the target point. For nodes in the middle of the element, the integration can be split in half, and each integration can use as the divisor the parametric distance from the target point to the source point. Without making any changes to the value of the integral, we can now integrate the function $-\ln(\frac{r}{\hat{xi}\hat{xi}}) = -\ln(\frac{r}{\hat{xi}}) - \ln(\hat{xi})$. The first term is now nonsingular and can be integrated with standard Gaussian quadrature, and the second can be integrated with a logarithmic Gaussian quadrature.

The strong singularity in the traction kernel is more difficult to handle in theory, but can actually be accommodated quite easily in practice. One common approach makes use of the rigid body modes of a structure. Assume that the tractions are all set to 0, in this case only rigid body modes can be present in the deflections. Since the right hand side of (4) is all 0, each row of $\mathbf{H}$ must sum to 0. The singularities are only present on the diagonal, so by setting each component of the block diagonal submatrices in $\mathbf{H}$ to the negative of the sums of the corresponding values in the other columns of $\mathbf{H}$, the singularities (and the C_{ij}) term never have to be calculated.

An alternative approach suggested by [2] makes use of an identity with the C_{ij} term to create a regularized traction kernel:

$$\int_\Gamma T^*_{ij}(\mathbf{x},\mathbf{y})(u_j(\mathbf{y}) - u_j(\mathbf{x}))d\Gamma(\mathbf{y}) = \int_\Gamma U^*_{ij}(\mathbf{x},\mathbf{y})t_j(\mathbf{y})d\Gamma(\mathbf{y}) \tag{7}$$

as a result, the first integral contains terms that are $\mathcal{O}(1/r)$ and $\mathcal{O}(r)$, and the singularity cancels out. This approach was used for this project because it is more amenable to NURBS basis functions where the shape functions do not take values of unity at the individual nodes.

3. Shape Optimization. The optimization of the structure can be done in a variety of ways, depending on the type of optimizer used (gradient-based or not). The prevailing approach in the topology optimization literature is to use gradient-based methods because they reduce the number of solutions of the pde by using information from the gradients. While any optimizer would likely suffice, the Method of Moving Asymptotes (MMA)[3] is commonly chosen as it was developed specifically for structural optimization problems and constrained optimization problems where a gradient can readily be obtained but Hessian information is unavailable or too expensive to compute.

The sensitivity of an objective or constraint function with respect to design variables takes the general form:

$$\frac{df}{d\alpha} = \frac{\partial f}{\partial \alpha} + \frac{\partial f}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\alpha} + \frac{\partial f}{\partial \mathbf{t}}\frac{d\mathbf{t}}{d\alpha} \tag{8}$$

Generally the term $\frac{\partial f}{\partial \alpha}$ is 0, and expressions for the terms $\frac{\partial f}{\partial \mathbf{u}}$ and $\frac{\partial f}{\partial \mathbf{t}}$ can be readily obtained. That leaves the terms $\frac{d\mathbf{u}}{d\alpha}$ and $\frac{d\mathbf{t}}{d\alpha}$ to be determined (usually by numerics). By differentiating (4) with respect to the design variable, we can obtain an adjoint equation

$$\mathbf{H}\mathbf{u}' + \mathbf{H}'\mathbf{u} = \mathbf{G}\mathbf{t}' + \mathbf{G}'\mathbf{t} \tag{9}$$

Assuming that the original system has already been solved (as it generally has to be to evaluate the objective/constraint function anyway), the only unknowns in this equation are $\mathbf{u}'$ and $\mathbf{t}'$, the sensitivities of the displacement and traction fields. Conveniently, this system is nearly identical to the one previously solved, just with an additional vector on each side. Thus solving the adjoint equation constitutes solving the same system as before, just with a different right-hand-side vector.

To construct that equation, it is needed obtain the sensitivities of the kernels. This is a relatively straightforward approach as the sensitivities are described by:

$$\begin{aligned}\mathbf{G}' = \int_\Gamma \frac{1}{8\pi\mu(1-\nu)} &\left[(3-4\nu)\,\delta_{ij}\left(\ln\dot{\frac{1}{r}}\right) + \dot{r_{,i}}r_{,j} + r_{,i}\dot{r_{,j}}\right] d\Gamma(\mathbf{y}) \\ &+ \frac{1}{8\pi\mu(1-\nu)}\left[(3-4\nu)\,\delta_{ij}\ln\frac{1}{r} + r_{,i}r_{,j}\right] d\dot{\Gamma}(\mathbf{y})\end{aligned} \tag{10}$$

(11)

$$
\begin{aligned}
T^*_{ij}(\mathbf{x},\mathbf{y}) = &-\frac{1}{4\pi(1-\nu)}\frac{\dot{1}}{r}\left\{\frac{\partial r}{\partial \mathbf{n}}\left[(1-2\nu)\,\delta_{ij}+2r_{,i}r_{,j}\right]+(1-2\nu)(n_i r_{,j}-n_j r_{,i})\right\}d\Gamma(\mathbf{y})\\
&-\frac{1}{4\pi(1-\nu)r}\left\{\frac{\dot{\partial} r}{\partial \mathbf{n}}\left[(1-2\nu)\,\delta_{ij}+2r_{,i}r_{,j}\right]\right\}d\Gamma(\mathbf{y})\\
&-\frac{1}{4\pi(1-\nu)r}\left\{\frac{\partial r}{\partial \mathbf{n}}(2\dot{r}_{,i}r_{,j})+(1-2\nu)(n_i\dot{r}_{,j}-n_j r_{,i})\right\}d\Gamma(\mathbf{y})\\
&-\frac{1}{4\pi(1-\nu)r}\left\{\frac{\partial r}{\partial \mathbf{n}}\left[(1-2\nu)\,\delta_{ij}+2r_{,i}r_{,j}\right]+(1-2\nu)(n_i r_{,j}-n_j r_{,i})\right\}d\dot{\Gamma}(\mathbf{y})
\end{aligned}
$$

where a $\dot{()}$ denotes a derivative with respect to the design variable. An analytical expression should exist for each of these sensitivities.

As described in [4], these sensitivity expressions share the same singularity behavior as the original kernels. What's more the logarithmic singularity in the displacement kernel disappears when obtaining its sensitivity. It still must be taken into account when constructing the term containing $d\dot{\Gamma}(\mathbf{y})$, but the kernel of that integral is the same as the original kernel and the singularity can be handled in the same method.

4. Example. In this section, we demonstrate the use of boundary integral methods and shape optimization to optimize a beam in tension. The objective function in this problem will be the compliance

$$\int_{\Gamma_t} t_i(\mathbf{y})u_i(\mathbf{y})d\Gamma_t(\mathbf{y}) \tag{12}$$

where Γ_t denotes the portion of the boundary with Neumann boundary conditions. The original domain of the structure is shown in Figure 1. Note that the left edge has a prescribed zero displacement, and the right edge has a uniform horizontal distributed load. The remainder of the boundary is unloaded. The material properties were chosen as those of aluminum, $\mu = 27GPa$ and $\nu = 0.33$. The design variables are the vertical component of the control points specifying the nodes along the top and bottom boundary. The first four nodes on either side of the top and bottom boundary are specified to create a smooth transition between the side boundaries and the top or bottom boundaries.

A volume constraint will be applied to the optimization, limiting the volume of the resulting structure to a volume of 75% of the original volume of the structure. The integral for the volume of the domain can be transferred to the boundary by making use of properties of the kronecker delta, δ, and the divergence theorem. Specifically in 2D:

$$\int_\Omega d\Omega(\mathbf{y}) = \frac{1}{2}\int_\Omega \delta_{ii}d\Omega(\mathbf{y}) = \frac{1}{2}\int_\Omega x_{i,i}(\mathbf{y})d\Omega(\mathbf{y}) = \frac{1}{2}\int_\Gamma x_i(\mathbf{y})n_i(\mathbf{y})d\Gamma(\mathbf{y}) \tag{13}$$

The sensitivity of that function is then

$$\frac{1}{2}\int_\Gamma \dot{x_i}(\mathbf{y})n_i(\mathbf{y})d\Gamma(\mathbf{y}) + \frac{1}{2}\int_\Gamma x_i(\mathbf{y})\dot{n_i}(\mathbf{y})d\Gamma(\mathbf{y}) + \frac{1}{2}\int_\Gamma x_i(\mathbf{y})n_i(\mathbf{y})d\dot{\Gamma}(\mathbf{y}) \tag{14}$$

The sensitivity of the compliance function is:

$$\int_\Gamma t_i(\mathbf{y})u_i\dot{(}\mathbf{y})d\Gamma(\mathbf{y}) + \int_\Gamma t_i(\mathbf{y})u_i(\mathbf{y})d\Gamma\dot{(}\mathbf{y}) \tag{15}$$

The resulting structure from the compliance optimization is shown in Figure 2. The result is sensible, a somewhat uniform cross section throughout the length of the beam (at least where the optimizer has control over the profile). However, the fact that any roughness appears in the structure is surprising. Rough edges along the profile generally mean that some material is being used inefficiently. In particular the spikes near the edge of the beam are also unexpected and don't appear to have a physical purpose It is possible that this is a result of some numerical error in the system, although the analytical sensitivities in the code were verified with finite difference approximations. Another possible culprit is erratic behavior in the MMA code. It should be noted here that this code was copied from an existing Matlab version for use in this project, and hasn't been throughly analyzed to ensure no bugs are present. Note in Figure 3 that the objective function at times jumped erratically from iteration to iteration. Also of note, the constraint value for volume was approximately 30, and the majority of the iterations violated this constraint. Nonetheless, when used to optimize the same problem, but with the compliance objective swapped for a minimization of perimeter, the result is smooth as expected (Figure 4), although a number if intermediate iterations displayed similarly nonsmooth boundaries. The shape optimized for least perimeter was then used as an initial condition for the compliance optimization in an attempt to provide a good initial condition for the compliance minimiziation (i.e. one that satisfies the volume constraint).

Fig. 1: Initial structure and boundary conditions. Nodes on the left edge have both degrees of freedom fixed, and nodes on the right edge have a uniform load applied horizontally.

5. Discussion. While the boundary integral method was used successfully here as part of a shape optimization scheme, it did exhibit some drawbacks. While the use of NURBS allows for the optimizer to capture shapes that aren't obtainable with e.g. Lagrange basis functions, they are more expensive to compute. In addition, per degree of freedom, the boundary integral equations are much more costly than similar methods for standard finite elements (not shown in this report). The fully dense nature of the operators also means that sensitivity calculations are much more

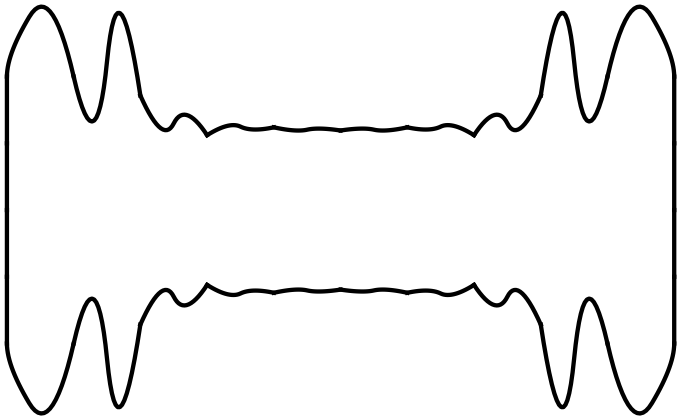

Fig. 2: Result of optimizing the structure for smallest compliance with a fixed maximum volume constraint.

expensive. In standard finite elements, the sensitivity of the system matrix with respect to a design variable is generally restricted to a small mxm matrix, where m is the number of degrees of freedom attached to any element. In the boundary element method however, the sensitivity is at least m dense columns in the matrix, with some additional dense rows as well. Thus, not only are the sensitivities more expensive to obtain, they are more expensive to use in constructing the adjoint equation.

In addition to the disadvantage of increased density in the boundary integral method, the resulting operators are also unsymmetric. Whereas the compliance problem is self-adjoint in standard finite elements due to symmetry of the operator, this is not the case for boundary integral methods. The standard finite element thus saves by not having to solve the adjoint problem (the solution can be substituted with the with the displacement field), whereas the boundary integral method requires an adjoint problem to be solved for every design variable. This may be the reason that many problems in the literature use only a handful of design variables (eg. radii of corners and thickness of members) which are all in turn functions of the control points/nodes, instead of treating each control point as a design variable as was done here.

Another interesting note for this problem is that the conditioning the system to be solved is in general very bad. This is due to the fact that the displacement kernel has the shear modulus term in the denominator, whereas the traction kernel does not. This leads to the operator $\mathbf{H}$ having a much larger norm than the operator $\mathbf{G}$. Thus, for pure Neumann or pure Dirichlet problems, the conditioning may be reasonable. However, for mixed problems there will often be issues (the example problem here had condition numbers on the order of $1e-13$). This is troublesome given that efficient performance of boundary methods often relies on the use of iterative methods like GMRES where the operator is implemented in an optimized fashion rather than as a true dense matrix. With condition numbers so large though, iterative solvers will often fail (as was the case in this project) and direct solvers must be used. Alternatively, one of the operators could be scaled by a constant to make the operators have similar

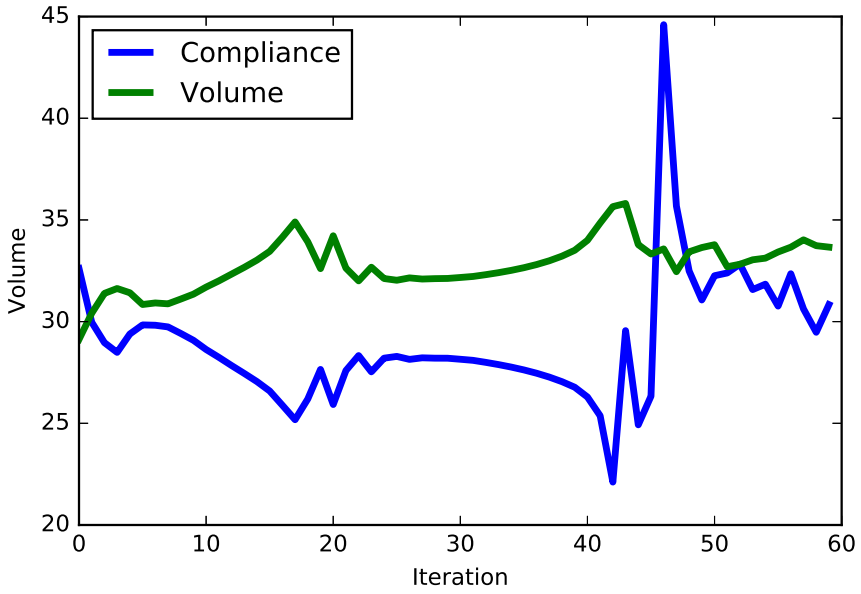

Fig. 3: Compliance value and volume of structure at each optimization iteration.

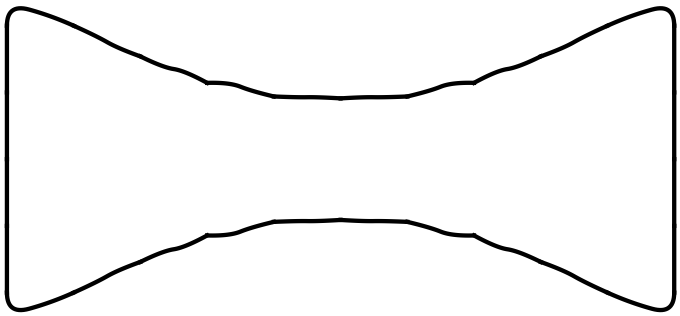

Fig. 4: Result of optimizing structure for smallest perimeter with a fixed maximum volume constraint.

norms. Then the solutions just need to be scaled appropriately after solving the system.

6. Conclusions. This project demonstrated how boundary integral methods and shape optimization can be coupled together, albeit on a very simple test problem. A gradient-based optimizer was used, relying on analytical expressions for the kernel sensitivities to construct an adjoint problem. This adjoint problem provided displacement and traction sensitivities that were used to determine the sensitivity of the compliance objective function. While the optimization produce some reasonable

results, there was some unexpected roughness in the resulting structure. Additionally, the boundary integral forumlation lacks some of the conveniences that are found in similar problems making use of standard finite elements. However, the low-rank behavior present in many of the operators used here could enable the use of some more efficient algorithms that were not leveraged for this project.

REFERENCES

[1] Lothar Gaul, Martin Kögl, and Marcus Wagner. *Boundary Element Methods for Engineers and Scientists*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[2] Yijun Liu and T.J. Rudolphi. Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations. *Engineering Analysis with Boundary Elements*, 8(6):301 – 311, 1991.

[3] Krister Svanberg. The method of moving asymptotes-a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.

[4] K TAI and R T FENNER. OPTIMUM SHAPE DESIGN AND POSITIONING OF FEATURES USING THE BOUNDARY INTEGRAL EQUATION METHOD. *International Journal for Numerical Methods in Engineering*, 39(12):1985–2003, 1996.