Optimal Sizing of Industrial Structural Mechanics Problems Using Automatic Differentiation

Gundolf Haase, Ulrich Langer, Ewald Lindner and Wolfram Mühlhuber

ABSTRACT We consider minimizing the mass of the frame of an injection moulding machine as an example of optimal sizing. The deformation of the frame is described by a generalized plane stress state with an elasticity modulus scaled by the thickness. The resulting constrained nonlinear optimization problem is solved by sequential quadratic programming (SQP), which requires gradients of the objective and the constraints with respect to the design parameters. As long as the number of design parameters is small, finite differences may be used. For several hundreds of varying thickness parameters, we use the reverse mode of automatic differentiation (AD). AD works fine but requires huge memory and disk capabilities and limits the use of iterative solvers for the governing state equations. Therefore, we combine AD with the adjoint method to get a fast and flexible hybrid gradient evaluation procedure. Numerical results show the potential of this approach and imply that this method can also be used for finding an initial guess for a shape optimization.

1 Introduction

The design of mechanical structures in many industrial applications has to fulfill various constraints. In most cases, engineers designing a machine component desire an optimal design subject to several constraints, but deadlines force them to truncate the design process after a few iterations and accept a suboptimal design. Therefore, tools supporting such a design process must 1) be flexible enough to handle the various requirements, requiring only a little work when the requirements change, and 2) be fast.

As an example for a typical optimal sizing problem, we minimize the mass of the frame of an injection moulding machine. We focus on the efficient and flexible calculation of the gradients of the objective and the constraints for use in standard optimization procedures (such as SQP). Since implementing analytic derivatives is infeasible, we present a very flexible approach using automatic differentiation (AD), coupled with a well-known adjoint method

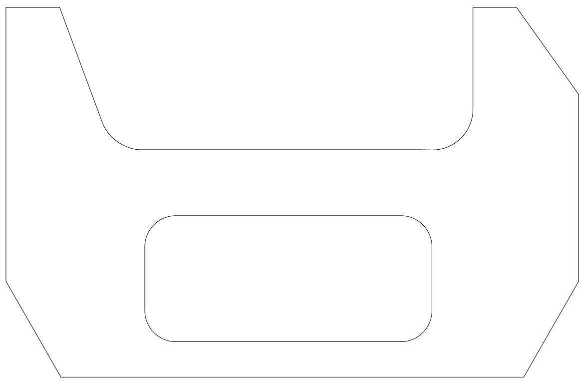

FIGURE 1. Cross section of the original shape

approach from shape optimization. Numerical results show the strength of this approach.

There are extensive reviews of the various methods for structural optimization using finite elements [7] or topology optimization [1]. Mahmoud [10] focuses on an optimal sizing approach similar to the one in this paper, but he uses approximate representations of the objective and the constraints to reduce the overall costs of the method. Stangl [15] presents a generalization of that approach to a class of nonlinearly elastic materials. For approaches using a topology optimization for getting an initial guess of the topology used in a shape optimization afterwards, see e.g. [11, 14]. In this volume, Forth and Evans [3] and Kim and Hovland [8] apply AD to similar problems in fluid dynamics inverse problems, respectively. Tadjouddine et al. [16] describe an efficient combination of AD and hand-coded parts of the derivative whose concept is similar to the approach in this paper.

2 Modeling of the problem

The frame of an injection moulding machine is shown by its 2D-cut Ω (Figure 1). For a frame of homogeneous thickness typical dimensions are:

thickness of one plate	180 mm
mass of one plate	3.8 tons
clumping force (surface force)	300 tons $\approx$ 16 N/mm^2
length	2.8 m
height	1.7 m
supporting areas	2

The primary goal of the design phase is to minimize the mass of the frame, subject to several other requirements.

Let $V_0 = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D, \text{meas}\, \Gamma_D > 0\}$ denote the set of admissible displacements, where $\partial\Omega = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. For a fixed thickness $\rho(x)$, the displacement field $u \in V_0$, satisfies

$$a(\rho; u, v) = F(v) \qquad \text{for all } v \in V_0 \tag{1.1}$$

with

$$a(\rho; u, v) = \int_\Omega \rho \, \frac{\partial u_i}{\partial x_j} \, E_{ijkl} \, \frac{\partial v_k}{\partial x_l} \, dx, \quad F(v) = \int_\Omega \langle f, v \rangle \, dx + \int_{\Gamma_N} g \, v \, ds \, ,$$

where E_{ijkl} denotes the elasticity tensor, f the volume force density and g the surface force density on a part Γ_N of the boundary. The design problem is

$$\begin{aligned}
& \int_\Omega \rho \, dx \to \min_{u,\rho} & & \\
\text{subject to} \quad & a(\rho; u, v) = F(v) & & \text{for all } v \in V_0 \\
& \underline{\rho} \le \rho \le \overline{\rho}, & & \text{a.e. in } \Omega \\
& \sigma^{\mathrm{vM}}(u) \le \sigma^{\mathrm{vM}}_{\max}, \quad \sigma^{\mathrm{ten}}(u) \le \sigma^{\mathrm{ten}}_{\max} & & \text{a.e. in } \Omega \\
& \alpha(u) \le \alpha_{\max} & &
\end{aligned} \tag{1.2}$$

where $\sigma^{\mathrm{vM}}(u)$ denotes the v. Mises stress, and $\sigma^{\mathrm{ten}}(u)$ denotes the tensile stress in the frame. The change in the shrinking angle of the clumping unit (vertical edges on top, called *wings*) is denoted by $\alpha(u)$.

For discretizing the problem, we use triangular finite elements with piecewise constant shape functions for approximating ρ and piecewise quadratic ones for approximating u. We denote the discrete approximation of ρ and u by ρ and u, respectively. In our application, the upper limits on the angle and the stresses are treated either as constraints or as soft limits, which can be violated to some extent, if the mass would be significantly smaller. Furthermore, the pointwise constraints on σ^{vM} and σ^{ten} are replaced using a higher order ℓ^p norm. Treating the upper limits as soft constraints leads to the reformulation:

$$\begin{aligned}
\mathrm{mass}(\rho) &+ \omega_1 \big(\max\left(\|\sigma^{\mathrm{vM}}\|_p - \sigma^{\mathrm{vM}}_{\max}, 0\right)\big)^2 + \omega_2 \big(\max\left(\|\sigma^{\mathrm{ten}}\|_p - \sigma^{\mathrm{ten}}_{\max}, 0\right)\big)^2 \\
&+ \omega_3 \big(\max\left(\alpha - \alpha_{\max}, 0\right)\big)^2 \quad \to \min_{u,\rho} \\
&K(\rho)\, u = F \quad \text{and} \quad \underline{\rho} \le \rho \le \overline{\rho}
\end{aligned} \tag{1.3}$$

3 Sketch of the Optimization Strategy

Problem (1.3) is a special case of

$$J(u, \rho) \to \min_{u,\rho} \quad \text{subject to } K(\rho)\, u \;= f(\rho) \text{ and } \underline{\rho} \le \rho \le \overline{\rho}, \tag{1.4}$$

where ρ denotes the vector of design parameters and u the solution of the governing finite element (FE) state equation. The splitting of the parameter vector into design parameters ρ and the solution of the FE state equation u is typical for problems in shape optimization. From the optimization's point of view, the discretized state equation can be interpreted as equality constraints. In our case it is linear with respect to u, and $K(\rho)$ is symmetric and positive definite for all admissible parameters ρ. Therefore, u can be eliminated:

$$\begin{aligned} &\tilde{J}(\rho) = J(K^{-1}(\rho)\, f(\rho),\, \rho) \to \min_{\rho} \\ \text{subject to} \qquad &\underline{\rho} \leq \rho \leq \overline{\rho} \end{aligned} \tag{1.5}$$

For a standard SQP optimization method, (1.5) is preferred to (1.4) because it has many fewer parameters (see [4, 12]). The optimizer used in our code is based on a Quasi-Newton approximation of the Hessian using a modified BFGS update formula of Powell [13] to avoid the need for Hessian information of the objective.

4 Calculating Gradients

Quasi-Newton strategy and update formulas need gradients for the objective and the constraints. Hand coded analytic derivatives are usually too complicated, error prone, and time consuming. Furthermore, hand coding is not well suited to the design process, as we would loose completely flexibility. Hence, we consider black box methods such as finite differences or automatic differentiation (AD) [5] or methods exploiting the special structure of the state equation such as the direct or the adjoint methods [7]. None of these methods is well suited for our problem, so we developed a hybrid method combining AD and the adjoint method. Since our finite element code is completely written in C++ and heavily uses virtual inheritance, we use ADOL-C [6]. We applied the reverse mode within our calculations, as we have at most a few nonlinear constraints (c.f. (1.2) and (1.3)), while the dimension of the design space is usually large.

4.1 Direct and adjoint methods

Both methods are well-known in shape optimization and take into account the special structure of the FE state equation. Differentiating the discretized state equation with respect to a design parameter ρ_i leads to

$$K \frac{\partial u}{\partial \rho_i} = \frac{\partial f}{\partial \rho_i} - \frac{\partial K}{\partial \rho_i}\, u. \tag{1.6}$$

For the direct method, (1.6) is solved numerically. Then the gradient of the objective can be calculated by (c.f. (1.5))

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\rho_i} = \frac{\partial J}{\partial \rho_i} + \left\langle \frac{\partial J}{\partial u}, \frac{\partial u}{\partial \rho_i} \right\rangle . \tag{1.7}$$

The adjoint method solves (1.6) formally and inserts the result in (1.7), which leads to

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\rho_i} = \frac{\partial J}{\partial \rho_i} + \left\langle K^{-T} \frac{\partial J}{\partial u}, \frac{\partial f}{\partial \rho_i} - \frac{\partial K}{\partial \rho_i} u \right\rangle . \tag{1.8}$$

4.2 Comparison

ADOL-C needs a file containing the evaluation graph of the function in a symbolic form which is generated at run time. Structural optimization problems require huge memory and disk capabilities for that purpose, e.g. about 1 GB is needed to store the evaluation graph for a problem with about 450 design parameters and about 7500 degrees of freedom. The flexibility of ADOL-C with respect to changes in the objective is similar to finite differences. For the reverse mode, the calculation time of the gradient is independent of the number of design parameters and takes the time of about 15 - 20 native C++ function evaluations as long as the evaluation graph can be stored in the main memory of the computer. Compared to the use of finite differences, this is a tremendous speedup, even for problems with only 10 - 20 design parameters. The coupling of AD with iterative solvers is a topic of current research (see e.g. [5] and references therein) since the use of iterative methods (e.g. multilevel methods) is important for solving fine discretisations of the state equation efficiently, the applicability is limited to problems where direct solvers can be used.

The direct method needs the solution of one state equation per design parameter, whereas the adjoint method needs the solution of one adjoint problem for the objective and in principle for each constraint. Depending on the number of design parameters and constraints, the better suited method can be chosen. As analytic partial derivatives of J with respect to ρ and u are needed, both methods can only be applied to simple objectives, where this can be done easily. Furthermore, the flexibility of the method suffers from the need of hand-coded gradient routines. Compared to finite differences or the use of AD for the whole function, this approach is much faster. Finite differences need many more solutions of the FE state problem, while avoiding the huge evaluation graph from the solution of the state equation required by AD. For both methods, any solver can be used for solving the state problem, including iterative solvers such as conjugate gradient methods with multilevel preconditioning.

		Finite Diff.	Pure AD	Hybrid M.
Problem dim.	Nr. of design par.	24	24	24
	Nr. of elements	3981	3981	3981
	DOFs of state equ.	16690	16690	16690
Optimizer statistics	Iterations	83	100	100
	gradient eval.	12.40 h	6.00 h	0.18 h
	function eval.	0.23 h	2.36 h	0.20 h
Runtime	Total CPU time	12.4 h	4.88 h	0.39 h
	Total elapsed time	12.6 h	8.42 h	0.40 h

TABLE 1.1. Comparison of the run time for various differentiation strategies

4.3 Hybrid Method

The strengths of the direct and the adjoint method and of AD lie in completely different areas. AD provides very high flexibility with respect to the objective, but has severe space and time requirements and cannot easily be used with iterative solvers for the state equation. In contrast, the direct and adjoint methods can be combined easily with iterative solvers and provide a fast way for calculating the needed gradients, but they lack the needed flexibility. Hence, we combine both approaches into a new hybrid method.

The main drawback of the direct or the adjoint method is the need for analytic partial derivatives of the objective and the constraints with respect to ρ and u, which can be provided easily by AD. Then $\frac{\partial K}{\partial p_i}$ and $\frac{\partial f}{\partial p_i}$ can be hand-coded or produced by AD. For optimal sizing problems, routines for $\frac{\partial K}{\partial p_i}$ and $\frac{\partial f}{\partial p_i}$ can be hand-coded easily. Furthermore, they can be quite general, justifying the additional effort of coding even for more complex problems.

5 Numerical Results

The following numerical results for the problem stated in §2 were calculated on an SGI Origin 2000 at 300 MHz. At the beginning, we tried to use only few design parameters. Therefore, we divided our domain into a number of sub-domains and approximated the thickness with a constant function per sub-domain. For evaluating the gradient, either finite differences, an AD approach for (1.5), or the hybrid method were used. For a better comparison (see Table 1.1), the calculation was terminated after a fixed number of steps. The run using finite differences terminated earlier because the search direction was not a descent direction.

All three methods lead to a similar design with about 5% reduction of the mass compared to the starting configuration (which is the current design of the frame). Compared to finite differences and the pure AD approach, the

		Pure AD	Hybrid M.	Hybrid M.
Problem dim.	Nr. of design par.	449	449	449
	Nr. of elements	1796	1796	7184
	DOFs of state equ.	7518	7518	29402
Evaluation graph	Operations	45521797	1399910	5578270
	Total file size	953 MB	32.4 MB	129.2 MB
Optimizer statistics	Iterations	800	800	800
	gradient eval.	18.0 h	0.54 h	3.35 h
	function eval.	16.5 h	1.29 h	8.13 h
Runtime	Total CPU time	32.3 h	3.73 h	14.01 h
	Total elapsed time	38.5 h	3.76 h	14.12 h

TABLE 1.2. Comparison of the run time for many design parameters

hybrid method is significantly faster, combining a fast function evaluation and a fast gradient evaluation. The gradient evaluation is the main drawback for finite differences. For the pure AD approach we implemented additional safeguards. To detect when a regeneration of the evaluation graph was necessary, we compared the value of the objective using the evaluation graph and the value using a native C++ implementation, which explains the longer run time of the function evaluation.

In the following we used the coarsest grid of our FE triangulation for discretizing the thickness. For solving the state problem, each coarse grid element was subdivided into 16 elements using 2 levels of uniform refinement. The state equation was discretized on this refined triangulation . Table 1.2 contains results for the pure AD approach and the hybrid method and results for an even finer discretization of the state equation.

Analyzing the run time behaviour we see that the pure AD approach is no longer competitive due to the large file containing the evaluation graph. Furthermore, it can be seen that for the hybrid approach the optimizer represents a considerable portion of the total run time, a portion that grows when using more design parameters as the complexity of one optimization step is proportional to $(\dim \rho)^3$ (due to the use of dense matrix linear algebra), whereas the complexity of solving one state equation is proportional to $\dim u$ (if solvers with optimal complexity e.g. conjugate gradients with multigrid or multilevel preconditioning are used).

6 Remarks and Conclusions

in this chapter, we presented differentiation strategies needed to solve a real optimal sizing problem. We focused our attention on the flexibility of the gradient routine and on the possibility of combining the gradient module with iterative solvers for the state equation. A hybrid method combines the strengths of AD and the adjoint method. This method preserves the flexibility of the pure AD approach at the cost of a completely hand-coded gradient routine. Furthermore, the huge memory and disk requirements of the pure AD approach are reduced significantly.

Our current implementation of the optimizer is based on dense matrix linear algebra. Therefore, it is only well suited for small to medium size optimization problems. To close the gap with topology optimization, which is of high practical importance, new optimization methods for large scale problems have to be developed such as an approach using multigrid methods proposed by Maar and Schulz [9].

Acknowledgement

This work was supported by the Austrian Science Fund - 'Fonds zur Fö rderung der wissenschaftlichen Forschung (FWF)' - SFB F013 'Numerical and Symbolic Scientific Computing'

7 References

[1] M. P. Bendsøe. *Optimization of Structural Topology, Shape and Material*. Springer, Berlin, 1995.

[2] George Corliss, Andreas Griewank, Christèle Faure, Laurent Hascoët, and Uwe Naumann, editors. *Automatic Differentiation 2000: From Simulation to Optimization*. Number THISNUMBER in THISSERIES. Springer, THISADDR, 2001.

[3] Shaun A. Forth and Trevor P. Evans. Aerofoil optimisation via AD of a multigrid cell-vertex Euler flow solver. In Corliss et al. [2], chapter 13.

[4] P. E. Gill, W. Murray, and Margret H. Wright. *Practical Optimization*. Academic Press, New York, 1981.

[5] Andreas Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, 2000.

[6] Andreas Griewank, David Juedes, and Jean Utke. ADOL–C, a package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. Math. Software*, 22(2):131–167, 1996. See `ftp://info.mcs.anl.gov/pub/tech_reports/reports/TM162.ps`.

[7] J. Haslinger and P. Neittaanmäki. *Finite Element Approximation for Optimal Shape Design: Theory and Applications*. John Wiley & Sons Ltd., Chichester, 1988.

[8] Jong G. Kim and Paul D. Hovland. Sensitivity analysis and parameter tuning of a sea ice model. In Corliss et al. [2], chapter 5.

[9] B. Maar and V. Schulz. Interior point multigrid methods for topology optimization. Technical Report 98–57, IWR, University of Heidelberg, 1998.

[10] K. G. Mahmoud. Approximations in optimum structural design. In B. H. V. Topping and M. Papadrakakis, editors, *Advanced in Structural Optimization*, pages 57–67. Civil-Comp Press, Edinburgh, 1994.

[11] K. Maute and E. Ramm. Adaptive topology optimization. *STROPT*, 10:100–112, 1995.

[12] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[13] M. J. D. Powell. A fast algorithm for nonlinear constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis*, number 630 in Lecture Notes in Mathematics. Springer, Berlin, 1978.

[14] E. Ramm, K.-U. Bletzinger, R. Reitinger, and K. Maute. The challenge of structural optimization. In B. H. V. Topping and M. Papadrakakis, editors, *Advanced in Structural Optimization*, pages 27–52. Civil-Comp Press, Edinburgh, 1994.

[15] C. Stangl. Optimal sizing for a class of nonlinearly elastic materials. *SIAJO*, 9(2):414–443, 1999.

[16] Mohamed Tadjouddine, Shaun A. Forth, and John D. Pryce. AD tools and prospects for optimal AD in CFD flux Jacobian calculations. In Corliss et al. [2], chapter 27.