XXI ICTAM, 15–21 August 2004, Warsaw, Poland

COMPUTATIONAL TRICKS FOR EFFICIENT DESIGN SENSITIVITY ANALYSIS

Henrik T. Møller and Erik Lund
Institute of Mechanical Engineering, Aalborg University,
Pontoppidanstræde 101, DK-9220 Aalborg East, Denmark

Summary When applying gradient based optimization tools on complex coupled problems it is of vital importance that the design sensitivity analysis phase does not take up to much of the time required in the optimization run. However, in the analysis of coupled problems the exact Jacobian is often not available and one is faced with the problem of solving a linear system with an unknown coefficient matrix. In the paper we present two iterative methods to accomplish this task efficiently. The key advantages of the two methods compared to an overall finite difference approach are better efficiency because information obtained during analysis can be reused, and a better or at least a known accuracy, because the sensitivity equations are solved to a specified tolerance. These key points are illustrated with an example.

Keywords: Sensitivity analysis; Implementation; Nonlinear problems; Fluid-structure interaction

SENSITIVITY ANALYSIS

When applying the direct differentiation method to perform sensitivity analysis, one ultimately ends up with the task of solving the following sequence of linear systems for each of the I design variables a_i in the computation of the state variable sensitivities $\frac{d\mathbf{u}}{da_i}$.

$$\frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} \frac{d\mathbf{u}}{da_i} = -\frac{\partial \mathbf{R}(\mathbf{u}, a_i)}{\partial a_i} \tag{1}$$

Here $\mathbf{R}(\mathbf{u})$ denotes the equations of state for the problem under consideration. The right hand side can be computed analytically or by using finite differences. The Jacobian $\frac{\partial \mathbf{R}}{\partial \mathbf{u}}$ is the same as the one used for the analysis if the analysis problem is based on some variant of Newton's method. Therefore the computation of the state sensitivities can be performed very efficiently when the solution for state is based on Newton's method and a direct solver is used together with an exact Jacobian. If an exact Jacobian is not used in the analysis, then the methodology can not be directly used, and instead an iterative method must be used. In the paper we focus on what kind of efficiency and accuracy that can be obtained in the sensitivity analysis with two types of modified methods which avoids the use of an exact Jacobian.

The base ingredient in both types of methods is the realization that the product of the system Jacobian and an arbitrary vector $\mathbf{z}$ can be computed by using forward differences by means of the definition of the directional derivative of $\mathbf{R}$, i.e.

$$\frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} \mathbf{z} \approx \frac{\mathbf{R}(\mathbf{u} + \alpha \mathbf{z}) - \mathbf{R}(\mathbf{u})}{\alpha}, \quad \text{with} \quad \alpha \sim \frac{\|\mathbf{u}\|_\infty}{\|\mathbf{z}\|_\infty} \sqrt{10\varepsilon_m} \tag{2}$$

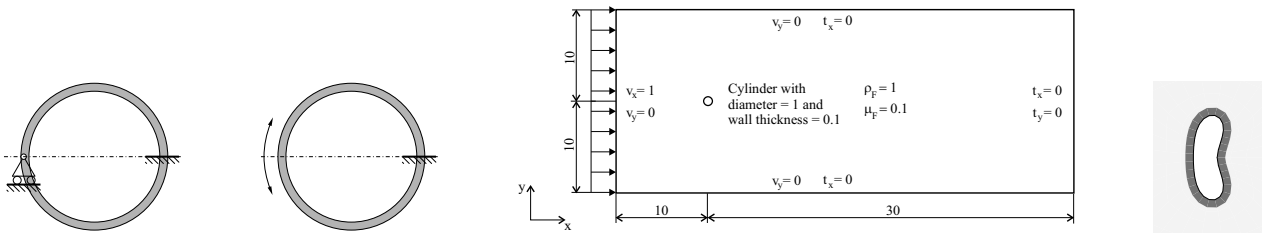
This way to evaluate the matrix-vector product is relatively cost efficient, provided that the computation of the residuals for the solution of the underlying nonlinear equations is efficient. Implementing the matrix product requires little effort. What is needed is to find a good perturbation α and a routine which can update the state variables according to a specified increment. The updating routine is the same as one would use in a Newton method for the analysis problem. The reason for putting 10 in front of the machine precision as indicated in (2) is an assumption that the residual computation is not accurate in the last significant digit. With this selection of the perturbation the implied accuracy in the matrix-vector product is such that approximately half the significant digits contained in the result are accurate provided that the problem is reasonably well scaled. Knowing a way to compute the product of the exact Jacobian with a vector the sensitivity equation can be rewritten in a form which makes use of the finite difference matrix-vector product as

$$\tilde{\mathbf{J}} \left(\Delta \frac{d\mathbf{u}}{da_i} \right)^k = -\frac{\partial \mathbf{R}}{\partial a_i} - \mathbf{J} \frac{d\mathbf{u}}{da_i}^k, \quad \text{with} \quad \frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{J} = \tilde{\mathbf{J}} + \mathbf{J}^d \tag{3}$$

where the Jacobian is split into the part used in the analysis as iteration matrix $\tilde{\mathbf{J}}$ and the part discarded in the analysis $\mathbf{J}^d$. This iteration is a stationary iterative method, i.e. the iteration matrix is constant throughout iterations.

The iterative DSA method of (3) can be improved by using a non-stationary method. The benefits are improved robustness and faster convergence. Since many iterative methods for linear systems only require matrix-vector products and preconditioning operations, the sensitivity system can be solved by using the finite difference evaluation of the matrix-vector product within the iterative method. This is a well-known approach for solving nonlinear equations and is often referred to as a matrix-free implementation or as an inexact Newton-Krylov method. The process of obtaining the design sensitivities is efficient because the preconditioner used for the analysis can be reused for the sensitivity analysis. If available, the iteration matrix $\tilde{\mathbf{J}}$ from the analysis phase is often a good preconditioner.

The second method uses Quasi Newton methods to solve the linear system. The initial iteration matrix is chosen as $\tilde{\mathbf{J}}$, and the iteration matrix is then updated during iterations using secant information. When $\tilde{\mathbf{J}}$ is a good approximation to the system Jacobian the methods are effective. The non-stationary method converges at a faster rate to a solution than the stationary method described by (3). For some of the examples studied the non-stationary method can converge to a much tighter convergence tolerance on the sensitivity system.



(a) (b) (c) (d)

Figure 1. Flexible cylinder example. Note the different boundary conditions in the two different cases.

Table 1. Number of matrix-vector products required for convergence in DSA example on mesh 1. The leftmost column shows boundary conditions. BGM is "Broyden's Good Method", GMRES is Generalized Minimum RESidual, BiCGSTAB is Bi-Conjugate Gradient STABilized, and CGS is Conjugate Gradient Squared.

Load case	DVNo	Stationary	BGM	GMRES	BiCGSTAB	CGS
	1	9	5	3	4	4
	2	9	5	3	4	4
	3	8	5	4	6	4
	4	8	6	4	6	4
	1	NC	6	4	6	6
	2	NC	7	4	6	6
	3	NC	7	5	8	8
	4	NC	7	5	8	12

EXAMPLE

The example is a nonlinear steady state fluid-structure problem and is illustrated in Fig. 1. The analysis problem is to find the flow around the flexible cylinder, which is very flexible, such that it deforms significantly due to the flow induced forcing on the interface between the solid domain and the fluid domain. The design variables we are investigating are two shape design variables (thickness of cylinder dv1 and dv2), and two material parameters, namely the stiffness of the cylinder material (dv3) and the fluid viscosity (dv4). Note that the example has been run with two sets of boundary conditions denoted by "a" and "b" in the table. The difference is that the cylinder is fixed vertically upstream in "a" and it is free to move in "b", see Fig. 1. This does not influence the obtained results, but it changes the spectral radius of the Jacobian, since fixing the cylinder in the front changes the smallest eigenvalue of the solid.

The results of the experiment are shown in Table 1. The basis for comparison is the number of matrix-vector multiplications needed for convergence. The convergence criterion was 10^{-5} on both error and solution update. As preconditioner in the Krylov methods and as the initial iteration matrix in BGM a full factorization of the approximate Jacobian is used. From the results it can be concluded that the stationary method always performs the worst of the methods, and for the case "b" boundary condition it does not converge. Therefore this method is not recommended for general use. The additional work involved in implementing BGM seems well worth the effort. Both the iterative Krylov solvers perform as well as BGM, however the effort required to implement the Krylov methods are substantially larger than what is required for BGM, assuming that the Krylov solvers are not already implemented for other purposes. Also note that the convergence criterion used for BGM is slightly tougher to fulfil than the one used for the Krylov methods. For the same residual norm the number of iterations differs with only one iteration. Of the two Krylov solvers the GMRES method gives the fastest convergence which will often be the case when a small number of iterations is used. The reason is that GMRES does not need to restart for the small number of iterations thus maintaining its minimization property.

The amount of time spent for the complete sensitivity analysis is around 2% of the analysis time for each design variable when a central difference is used in the matrix-vector product. This includes calculating the pseudo load with central differences. If forward differences are used the time would be approximately halved. The time complexity of the sensitivity analysis scales better than the analysis in terms of number of unknowns, where the most time consuming task for the sensitivity analysis is back substitution/preconditioning operations compared to matrix factorizations for the analysis. This has been verified numerically by using a sequence of successively refined discretizations.

CONCLUDING REMARKS

The two improved methods for performing iterative DSA on complex systems both perform better than the simple iterative method, both in terms of efficiency and robustness. The methods have also been successfully used on problems involving turbulence models on top of the coupled analysis, and they are easily extendible to other types of problems, as long as the underlying method for analysis is residual based. The resulting efficiency of the methods is as good as the efficiency of the residual evaluation.