Advanced solution methods in topology optimization and shape sensitivity analysis

ARTICLE *in* ENGINEERING COMPUTATIONS · JULY 1996

Impact Factor: 1.5 · DOI: 10.1108/02644409610120696

4 AUTHORS, INCLUDING:

Papadrakakis Manolis

National Technical University of Athens

249 PUBLICATIONS **2,792** CITATIONS

SEE PROFILE

Yiannis Tsompanakis

Technical University of Crete

65 PUBLICATIONS **357** CITATIONS

SEE PROFILE

J. Sienz

Swansea University

99 PUBLICATIONS **513** CITATIONS

SEE PROFILE

Advanced solution methods in topology optimization and shape sensitivity analysis

Manolis Papadrakakis and Yiannis Tsompanakis
Institute of Structural Analysis and Seismic Research, National Technical University of Athens, Greece
Ernest Hinton and Johann Sienz
University of Wales, Swansea, Wales, UK

Introduction

In recent years, the integration of topology and shape optimization methods with the modern CAD technology has made remarkable progress in achieving automatically optimum designs. Topology optimization is usually employed first, in order to avoid local optima due to a crude initial layout, followed by shape optimization in order to fine tune the optimum layout. Communication between the two optimization stages is carried out through an image processing tool which extracts the final topology and pass the model to the shape optimizer[1]. Integrated methodologies, where topology and shape optimization are dealt not in succession but simultaneously, using the same design model in every optimization step, have also evolved[2,3].

Topology optimization is a tool which assists the designer in the selection of suitable initial structural topologies by removing or redistributing material of the structural domain. The procedure starts with an initial layout of the structure followed by a gradual "removal" of a small portion of low stressed material which is being used inefficiently. This procedure is a typical case of structural reanalysis. which results in small variations of the stiffness matrix between two optimization steps.

In a shape optimization problem the aim is to improve a given topology by minimizing an objective function subjected to certain equality and inequality constraints, where all functions are related to the design variables. The shape optimization algorithm proceeds with the following steps:

(1) a finite element mesh is generated;

(2) displacements and stresses are evaluated;

(3) sensitivities are computed by perturbing each design variable by a small amount; and

This work has been supported by HC&M 9203390 project of the European Union. The Swansea authors wish to acknowledge funding provided by the EPSRC project GR/K22839.

(4) the optimization problem is solved and the new shape of the structure is defined.

These steps are repeated until convergence has occurred. The most time-consuming part of a gradient-based optimization process is devoted to the sensitivity analysis phase. For this reason several techniques have been developed for the efficient calculation of the sensitivities in an optimization problem. The semi-analytical and the finite difference techniques are the two most widely used types of sensitivity analysis techniques. In the first case only the right-hand sides change and the coefficient matrix remains the same, while in the second case the coefficient matrix is modified as well due to the perturbations of the design variables. The latter technique results in a similar problem as in the case of topology optimization.

Usually, a structural optimization problem, whether it is topology, shape, or an integrated structural optimization problem is a computationally intensive task, where 60 to 90 per cent of the computations are spent on the solution of the finite element equilibrium equations. Although it is widely recognized that hybrid solution methods, based on a combination of direct and iterative solvers, outperform their direct counterparts, both in terms of computing time and storage, little effort has been devoted until now to their implementation in the field of structural optimization. In the present study three innovative solution methods are implemented based on the preconditioned conjugate gradient (PCG) and Lanczos algorithms. The first method is a PCG algorithm with a preconditioner resulted from a complete or an incomplete Cholesky factorization, the second is a PCG algorithm in which a truncated Neumann series expansion is used as preconditioner, and the third is the preconditioned Lanczos algorithm properly modified to treat multiple right-hand sides. The numerical tests presented demonstrate the computational advantages of the proposed methods, which become more pronounced in large-scale and/or computationally-intensive optimization problems.

Topological design of stuctures

In most shape optimization problems the topology of the structure is given and only the shape of the boundaries varies (boundary variation method). This optimized design, however, is dependent on the given topology and thus this solution may correspond only to a local optimum. For this reason, topology optimization techniques have been developed which allow the designer to optimize the layout or the topology of a structure. Topological or layout optimization can be undertaken by employing one of the following main approaches which have evolved during the last few years[1]:

(1) the ground structure[4];

(2) the homogenization method[5];

(3) the bubble method[6]; and

(4) the evolutionary fully-stressed design technique[7].

The first three approaches have several things in common. They are optimization techniques with an objective function, design variables, constraints and they solve the optimization problem by using an algorithm based on sequential quadratic programming (approach (1)), or on an optimality criterion concept (approaches (2) and (3)). However, inherently linked with the solution of the optimization problem is the complexity of these approaches. The fully-stressed design technique on the other hand, although not an optimization algorithm in the conventional sense, proceeds by removing inefficient material, and therefore optimizes the use of the remaining material in the structure, in an evolutionary process. The work presented in this study related to topology optimization is based on an improved implementation of the evolutionary fully stressed design technique proposed by Hinton and Sienz[1], and investigates the influence of effective hybrid solution techniques on the overall performance of the topology optimization problem.

Basic algorithm for topology optimization

This algorithm for topology optimization is based on the simple principle that material which has small stress levels is used inefficiently and therefore it can be removed. Thus, by removing small amounts of material at each optimization step, the layout of the structure evolves gradually. In order to achieve convergence of the whole optimization procedure, it is important for the amount of material removed at each stage to be small and to maintain a smooth transition from one layout of the structure to the subsequent one.

The domain of the structure, which is called the reference domain, can be divided into the design domain and the non-design domain. The non-design domain covers regions with stress concentrations, such as supports and areas where loads are applied, and therefore it cannot be modified throughout the whole topology optimization process. After the generation of the finite element mesh, the evolutionary fully stressed design cycle is activated, where a linear elastic finite element stress analysis is carried out. The equivalent stress σ_{eq}, or the maximum principal stress σ_{pr}, for each element can be computed, which for convenience is called in both cases the stress level and is denoted as σ_{evo}. The maximum stress level $\sigma_{\max}$ Of the elements in the structure at the current optimization step is defined, and all elements that fulfil the condition

$$\sigma_{evo} < \mathrm{ratre} * \sigma_{\max} \tag{1}$$

are removed, or switched off, where ratre is the rejection rate[7]. The elements are removed by assigning them a relatively small elastic modulus which is typically

$$E_{\mathrm{off}} = 10^{-5} * E_{\mathrm{on}}. \tag{2}$$

In this way the elements switched-off virtually do not carry any load and their stress levels are accordingly small in subsequent analyses. This strategy is called “hard kill”, since the low stressed elements are immediately removed, in contrast with the “soft kill” method where the elastic modulus varies linearly and the elements are removed more gradually. The remaining elements are considered active and they are sorted in ascending order according to their stress levels before a subsequent analysis is performed.

The number of active elements to be switched off per optimization step depends on the following control parameters:

(1) the parameter switl related to the minimum percentage of active elements to be switched off providing a lower bound for the number of elements to be switched off;

(2) the parameter switu related to the maximum percentage of active elements providing an upper bound for the number of elements to be switched-off;

(3) the cut-off stress parameter strmx, which is used to switch off all the elements having stresses lower than the cut-off stress.

The first two parameters are expressed as a percentage of the elements to be switched off per optimization step and operate as a safeguard against a stagnation of the evolutionary process, or an abrupt change of the stress field in the remaining structure. When only a few elements are switched off the rejection rate is incremented by the evolution rate in order to accelerate the convergence of the whole process

$$\mathrm{ratre}_{i+1} = \mathrm{ratre}_i + \mathrm{ratev}\ (i = 1, 2, \ldots). \qquad (3)$$

Typical values for the rejection rate and the evolution rate are ratre = 1 per cent and ratev = 1 per cent.

In addition to the above-mentioned control parameters there are two more criteria which influence the switching off of elements:

(1) *Composite suppression.* During the evolutionary process there will sometimes be single elements which are switched on but which are surrounded by elements that are switched off. This may result in a chess board pattern-like topology of elements which are alternatively switched on and off, and it is not desired.

(2) *Element growth.* This option allows the designer to switch on elements adjacent to elements in which the stress level is larger than a specified stress limit. This is the only way in which elements which have been switched off can be switched on again.

The iterative process of element removal and addition is continued until one of several specified convergence criteria is met:

(1) All stress levels are larger than a certain percentage value, defined by rspst, of the maximum stress. This criterion assumes that a fully stressed design has been achieved and the material is used efficiently.

(2) The number of active elements is smaller than a specified percentage, defined by rspel, of the total number of elements.

(3) When element growth is allowed the evolutionary process is completed when more elements are switched on than they are switched off.

Shape optimization

The shape optimization method used in the present study is based on a previous work for treating two-dimensional problems by Hinton and Sienz[8]. It consists of the following essential ingredients: shape generation and control; mesh generation; adaptive finite element analysis; sensitivity analysis; and shape optimization.

The method proceeds with the following steps:

(1) At the outset of the optimization, the geometry of the structure under investigation has to be defined. The boundaries of the structure are modelled using cubic B-splines which, in turn, are defined by a set of key points. Some of the co-ordinates of these key points will be the design variables which may or may not be independent.

(2) An automatic mesh generator is used to create a valid and complete finite element model. A finite element analysis is then carried out and the displacements and stresses are evaluated. An *h*-type adaptivity analysis may be incorporated in this stage.

(3) The sensitivities of the constraints and the objective function to small changes in the design variables are computed either with the finite difference, or the semi-analytical method.

(4) A suitable mathematical programming method, like the sequential quadratic programming SQP algorithms, is used to optimize the design variables.

If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been found and the solution process is terminated, otherwise the new geometry is defined and the whole process is repeated from step (2). The whole procedure is depicted in Figure 1.

Sensitivity analysis

Sensitivity analysis is the most important and time-consuming part of shape optimization. Although sensitivity analysis is mostly mentioned in the context of structural optimization, it has evolved into a research topic of its own. Several techniques have been developed which can be mainly distinguished by their numerical efficiency and their implementation aspects. The methods for sensitivity analysis can be divided into discrete and variational methods[9]. In the variational approach the starting point is an idealized but continuous structure, such as beam or shell. Applying basic variational theorems to the functions and operators describing the structure and the optimization problem, the sensitivity coefficients can be determined. In this case, the sensitivities are given as boundary and surface integrals which are then solved after the structure has been discretized. In the discrete approach the derivatives or the

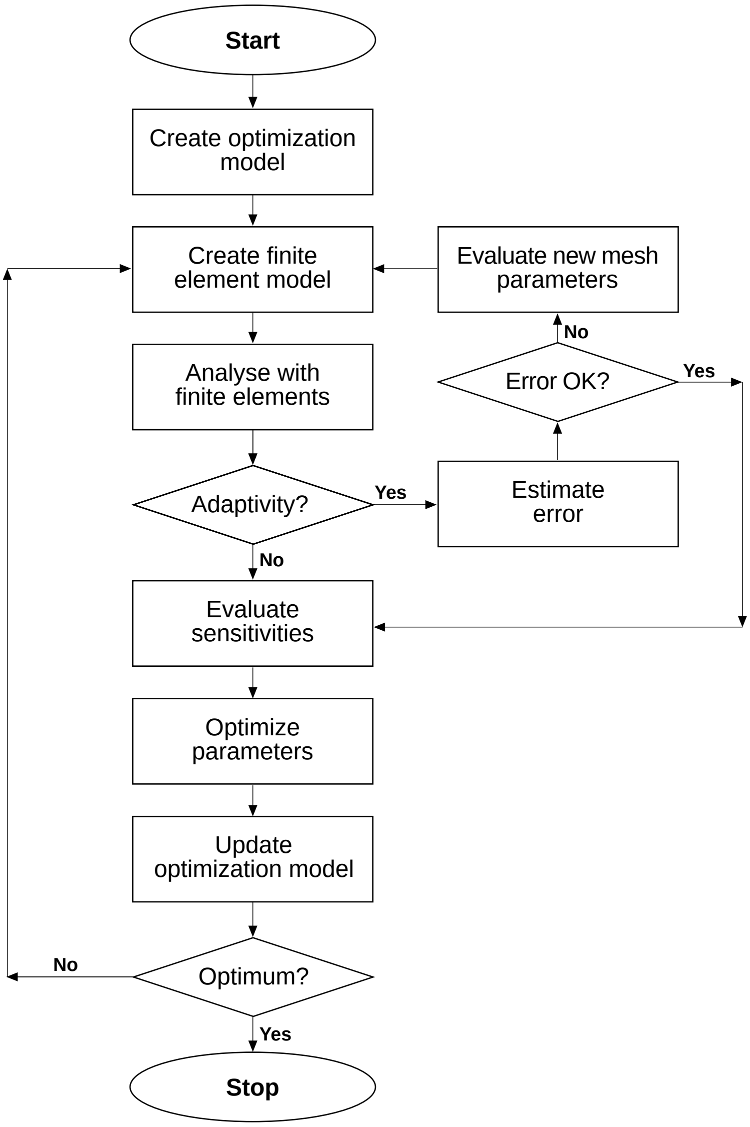

Start
Create optimization model
Create finite element model
Evaluate new mesh parameters
No
Analyse with finite elements
Error OK?
Yes
Adaptivity?
Yes
Estimate error
No
Evaluate sensitivities
Optimize parameters
Update optimization model
No
Optimum?
Yes
Stop

Figure 1.
Flowchart of the shape optimization algorithm

sensitivities of the characteristic parameters, i.e. displacements, stresses, volume, etc., which define the objective and constraint functions of the discretized structure, are evaluated using the finite element equations.

The implementation for discrete methods is simpler than that for variational techniques. A further classification of the discrete methods is as follows:

- *Global finite difference method.* A full finite element analysis has to be performed for each design variable and the accuracy of the method depends strongly on the value of the perturbation of the design variables.

- *Semi-analytical method.* The stiffness matrix of the initial finite element solution is retained during the computation of the sensitivities. This provides an improved efficiency over the finite difference method by a relatively small increase in the algorithmic complexity. The accuracy problem involved with the numerical differentiation can be overcome by using the exact semi-analytical method, which needs more programming effort than the simple method but, is computationally more efficient.
- *Analytical method.* The finite element equations, the objective and constraint functions, are differentiated analytically. The implementation is more complex than for the other two discrete methods and requires access to the source code.

The decision on which method to implement depends strongly on the type of the problem, the structure of the computer program and the access to the source code. The variational method is usually more reliable but requires a more complex computer implementation, whereas when a finite difference method is applied the formulation is much simpler and the sensitivity coefficients can be easily evaluated. In the present investigation the first two variations of discrete approach have been implemented. Irrespective of the type of approach used, the sensitivity analysis can take 50-80 per cent of the total computational effort required to solve the whole optimization problem. Therefore, an efficient and reliable sensitivity analysis module results in a considerable reduction of the overall computational effort.

The global finite difference (GFD) method. In shape optimization the stiffness matrix **K** and the loading vector **b** are functions of the design variables s_k (k = 1, …, n). The primary objective is to compute the derivative of the displacement field with respect to s_k. The GFD method provides a simple and straightforward way of computing the sensitivity coefficients. This method requires the solution of a linear system of equations **Kx** = **b**, where **x** is the displacement vector, for the original design variables s^0, and for each perturbed design variable $s_k^p = s_k^0 + \Delta s_k$, where Δs_k is the magnitude of the perturbation. The design sensitivities for the displacements $\partial_x/\partial s_k$ and the stresses $\partial\sigma/\partial s_k$ are computed using a forward difference scheme:

$$\partial\,\mathbf{x}\,/\,\partial s_k \approx \frac{\Delta\mathbf{x}}{\Delta s_k} = \frac{\mathbf{x}(s_k + \Delta s_k) - \mathbf{x}(s_k)}{\Delta s_k} \tag{4}$$

$$\partial\sigma\,/\,\partial s_k \approx \frac{\Delta\sigma}{\Delta s_k} = \frac{\sigma(s_k + \Delta s_k) - \sigma(s_k)}{\Delta s_k}. \tag{5}$$

The perturbed displacement vector $\mathbf{x}(s_k + \Delta s_k)$ is evaluated by

$$\mathbf{K}(s_k + \Delta s_k)\,x\,(s_k + \Delta s_k) = b(s_k + \Delta s_k) \tag{6}$$

and the perturbed stresses $\sigma(s_k + \Delta s_k)$ are computed from

$$\sigma(s_k + \Delta s_k) = \mathbf{DB}(s_k + \Delta s_k)\, x\, (s_k + \Delta s_k) \tag{7}$$

where **D** and **B** are the elasticity and the deformation matrices respectively.

The GFD scheme is usually sensitive to the accuracy of the computed perturbed displacement vectors, which is dependent on the magnitude of the perturbation. Too small values for the perturbation lead to insignificant changes of the response parameters making the whole procedure vulnerable to round-off errors, whereas too big values give rise to inaccuracies associated with any scheme based on truncated Taylor series expansion. The proper choice for the value of the perturbation is problem dependent and can be found by trial and error. It is usually taken between 10^{-3} and 10^{-5}.

The semi-analytical (SA) method. The SA method is based on the chain rule differentiation of the finite element equations $\mathbf{Kx} = \mathbf{b}$

$$\mathbf{K}\frac{\partial\, \mathbf{x}}{\partial s_k} + \frac{\partial\, \mathbf{K}}{\partial s_k}\,\mathbf{x} = \frac{\partial\, \mathbf{b}}{\partial s_k} \tag{8}$$

which when rearranged results in

$$\mathbf{K}\frac{\partial \mathbf{x}}{\partial s_k} = \mathbf{b}_k^* \tag{9}$$

where

$$\mathbf{b}_k^* = \frac{\partial\, \mathbf{b}}{\partial s_k} - \frac{\partial\, \mathbf{K}}{\partial s_k}\,\mathbf{x} \tag{10}$$

$\mathbf{b}_k^*$ represents a pseudo-load vector. The derivatives $\partial_b/\partial s_k$ and $\partial\mathbf{K}/\partial s_k$ are computed for each design variable by recalculating the new values of $\mathbf{K}(s_k + \Delta s_k)$ and $\mathbf{b}(s_k + \Delta s_k)$ for a small perturbation Δs_k of the design variable s_k, while the stiffness matrix remains unchanged throughout the whole sensitivity analysis.

The conventional semi-analytical (CSA) method. In the CSA sensitivity analysis, the values of the derivatives in equation (8) are calculated by applying the forward difference approximation

$$\partial\, \mathbf{K} / \partial s_k \approx \frac{\Delta\mathbf{K}}{\Delta s_k} = \frac{\mathbf{K}(s_k + \Delta s_k) - \mathbf{K}(s_k)}{\Delta s_k} \tag{11}$$

$$\partial\, \mathbf{b} / \partial s_k \approx \frac{\Delta\ \ \mathbf{b}}{\Delta s_k} = \frac{\mathbf{b}(s_k + \Delta s_k) - \mathbf{b}(s_k)}{\Delta s_k}. \tag{12}$$

For maximum efficiency of the semi-analytical method only those elements which are affected by the perturbation of a certain design variable are involved into the calculations of $\partial b/\partial s_k$ and $\partial\mathbf{K}/\partial s_k$.

Stress gradients can be calculated by differentiating $\sigma = \mathbf{DBx}$ as follows

$$\frac{\partial \sigma}{\partial s_k} = \frac{\partial \mathbf{D}}{\partial s_k}\mathbf{Bx} + \mathbf{D}\frac{\partial \mathbf{B}}{\partial s_k}\mathbf{x} + \mathbf{DB}\frac{\partial \mathbf{x}}{\partial s_k}. \tag{13}$$

Since the elasticity matrix, D, is not a function of the design variables then equation (13) reduces to

$$\frac{\partial \sigma}{\partial s_k} = \mathbf{D}\frac{\partial \mathbf{B}}{\partial s_k}x + \mathbf{DB}\frac{\partial x}{\partial s_k}. \tag{14}$$

In equation (14), $\partial x/\partial s_k$ may be computed as indicated in equation (4), while the term $\partial \mathbf{B}/\partial s_k$ is computed using a forward finite difference scheme as follows

$$\frac{\partial \mathbf{B}}{\partial s_k} = \frac{\Delta \mathbf{B}}{\Delta s_k} = \frac{\mathbf{B}(s_k + \Delta s_k) - \mathbf{B}(s_k)}{\Delta s_k}. \tag{15}$$

The "exact" semi-analytical (ESA) method. The CSA method may suffer some drawbacks in particular types of shape optimization problems. This is due to the fact that, in the numerical differentiation of the element stiffness matrix with respect to shape design variables, the components of the pseudo load vector associated with the rigid body rotation do not vanish. A similar inaccuracy problem is not found if the analytical or the global finite difference method is employed. The solution suggested by Olhoff *et al.*[10] alleviates the problem by performing an "exact" numerical differentiation of the elemental stiffness matrix based on computationally inexpensive first order derivatives. These derivatives can be obtained on the element level as follows

$$\frac{\partial k}{\partial s_k} = \sum_{i=1}^{n} \frac{\partial k}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial s_k} \tag{16}$$

where n is the number of element nodal co-ordinates affected by the perturbation of the design variable s_k and α_j is the nodal co-ordinate of the element which can be either a x-co-ordinate or a y-co-ordinate. The derivative $\partial \alpha_j/\partial s_k$ can be computed "exactly" by means of a forward difference scheme

$$\frac{\partial \alpha_j}{\partial s_k} = \frac{\Delta \alpha_j}{\Delta s_k} = \frac{\alpha_j(s_k + \Delta s_k) - \alpha_j(s_k)}{\Delta s_k} \tag{17}$$

while the derivative $\partial_k/\partial \alpha_j$ is computed by differentiating the element stiffness matrix expression

$$k = \int_{\Omega} \mathbf{B}^T \mathbf{DB} \,|\, J \,|\, d\Omega \tag{18}$$

with respect to a perturbation of the nodal co-ordinate α_j:

$$\frac{\partial k}{\partial \alpha_j} = 2\int_{\Omega} \left[\mathbf{B}^T \mathbf{D} \frac{\partial \mathbf{B}}{\partial \alpha_j} \right] |\, J \,|\, d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \frac{\partial |\, J \,|}{\partial \alpha L j} d\Omega \tag{19}$$

after using symmetry properties of **D** and k. Equation (19) can be rewritten as

$$\frac{\partial k}{\partial \alpha_j} = 2\int_{\Omega} \hat{\mathbf{B}}^T \mathbf{D} \hat{\mathbf{B}}^{(j)} \,|\, J \,|\, d\Omega \tag{20}$$

where $\hat{\mathbf{B}}^{(j)}$ is given as

$$\hat{\mathbf{B}}^{(j)} = \frac{\partial \mathbf{B}}{\partial \alpha_j} + 2 \frac{\mathbf{B}}{2\,|\, J \,|} \frac{\partial |\, J \,|}{\partial \alpha_j}. \tag{21}$$

Since equations (18) and (20) have the same structure the existing computer code for calculating the stiffness matrix can be used for the computation of the derivative of the elemental stiffness matrix, as long as it is possible to substitute $\hat{\mathbf{B}}^{(j)}$ in the place of **B**. The question as to whether $\partial k/\partial \alpha_j$ can be computed "exactly" is now reduced to an investigation of whether the derivatives of B and $|J|$ with respect to a nodal co-ordinate perturbation α_j can be computed with the required accuracy. The derivative $\partial |J| / \partial \alpha_j$ can be computed "exactly" with a forward difference scheme

$$\frac{\partial |\, J \,|}{\partial \alpha_j} = \frac{\Delta |\, J \,|}{\Delta \alpha_j} = \frac{\left| J(\alpha_j + \Delta \alpha_j) \right| - \left| J(\alpha_j) \right|}{\Delta \alpha_j}. \tag{22}$$

For the differentiation of **B**, the derivatives of the shape functions with respect to a change in the nodal co-ordinates need to be evaluated. This can be done "exactly" using the following formula

$$\frac{\partial}{\partial \alpha_j} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} = -\, J^{-1} \frac{\partial |\, J \,|}{\partial \alpha_j} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} \tag{23}$$

where $\partial J / \partial \alpha_j$ can also be computed with a forward difference scheme

$$\frac{\partial J}{\partial \alpha_j} = \frac{\Delta J}{\Delta \alpha_j} = \frac{J(\alpha_j + \Delta \alpha_j) - J(\alpha_j)}{\Delta \alpha_j}. \tag{24}$$

Applying equation (23) to all terms in the **B** matrix, the "exact" derivative of $\partial \mathbf{B}/\partial \alpha_j$ can be evaluated. The evaluation of the "exact" derivatives of the stiffness matrix is computationally more expensive than the corresponding derivatives of the CSA method. This overhead, however, is counterbalanced by the gains in accuracy and the improvement on the overall optimization procedure. The stress sensitivities are computed in the same way as with the CSA method using equation (14), where the term $\partial \mathbf{B}/\partial s_k$, instead of using equation (15), is evaluated in this case with respect to the nodal perturbations of co-ordinate α_j using equations (23) and (24).

Hybrid solution methods

The implementation of hybrid solution schemes in structural optimization, which are based on a combination of direct and preconditioned iterative methods, has not yet received the attention it deserves from the research community, even though in finite element linear solution problems and particularly when dealing with large-scale applications their efficiency is well documented. In this work several hybrid methods are applied in the context of topology and shape optimization based on PCG and Lanczos algorithms. The methods are properly modified to address the special features of the particular optimization problems at hand, while mixed precision arithmetic operations are proposed resulting in additional savings in computer storage without affecting the accuracy of the solution.

The preconditioned conjugate gradient (PCG) method

The PCG method has become very popular for the solution of large-scale finite element problems. An efficient preconditioning matrix makes PCG very attractive even for ill-conditioned problems without destroying the characteristic features of the method. Several global preconditioners have been used in the past for solving finite element linear problems[11]. The most important factor for the success of the hybrid methods is the preconditioning technique used to improve the ellipticity of the coefficient matrix. Preconditioning techniques based on incomplete Cholesky factorization are capable of increasing the convergence rate of the basic iterative method, at the expense of more storage requirements. In the present study the incomplete procedure by magnitude proposed by Bitoulas and Papadrakakis[12], is implemented in a mixed precision arithmetic mode, and a compact storage scheme is used to store both the coefficient and the preconditioning matrices row-by-row.

The reason for performing an incomplete factorization is to obtain a reasonably accurate factorization of the coefficient matrix without generating too many fill-ins, whereas a complete factorization produces the strongest possible preconditioner, which in exact precision arithmetic is actually the inverse of K. Both approaches lead to the factorization of the coefficient matrix $\mathbf{K}$: $LDL^T = \mathbf{K}_0 + \Delta\mathbf{K} – \mathbf{E}$, where $\mathbf{E}$ is an error matrix which does not have to be formed. For this class of methods, $\mathbf{E}$ is defined by the computed positions of “small” elements in L which do not satisfy a specified magnitude criterion and therefore are discarded. If we have to deal with a reanalysis problem, which is the case of GFD sensitivity analysis and topology optimization problems, the matrix $\mathbf{E}$ is taken as the $\Delta\mathbf{K}$ matrix, whereas in the case of SA sensitivity analysis both $\mathbf{E}$ and $\Delta\mathbf{K}$ are taken as null matrices.

Non-zero terms are stored in a real vector, while the corresponding column numbers are stored in an integer vector of equal length. An additional integer vector with length equal to the number of equations is used to record the start of each row inside the compact scheme. Thus, the total storage requirements are NA (stiffness in double precision arithmetic), NL (preconditioner in single precision arithmetic) and NA + NL + 2*(N + 1) short integers (for the addressing), where N is the number of equations. Also the complete Cholesky factorization of the "initial" stiffness matrix $\mathbf{K}_0$ is used as the preconditioning matrix in its original skyline form and is stored in single precision arithmetic, thus it does not require the NL short integers for the addressing the compact.

Another important factor affecting the performance of the PCG iterative procedure for solving $\mathbf{Kx} = \mathbf{b}$ is the determination of the residual vector. The accuracy achieved and the computational labour of the method is largely determined by how this is calculated. A study performed in [11] revealed that the computation of the residual vector of the equilibrium equations $\mathbf{Kx} = \mathbf{b}$ from its defining formula $r^{(m)} = \mathbf{Kx}^{(m)} - \mathbf{b}$, with an explicit or a first order differences matrix-vector multiplication $\mathbf{Ku}^{(m)}$, offers no improvement in the accuracy of the computed results. In fact, it was found that, contrary to previous recommendations, the calculation of the residuals by the recursive expression $r^{(m+1)} = r^{(m)} + \alpha_m\mathbf{Kd}^{(m)}$, where $\mathbf{d}$ is the direction vector, produces a more stable and well-behaved iterative procedure. Based on this observation a mixed precision arithmetic PCG implementation is proposed in which all computations are performed in single precision arithmetic, except for double precision arithmetic computation of the matrix-vector multiplication involved in the recursive evaluation of the residual vector. This implementation is a robust and reliable solution procedure even for handling large and ill-conditioned problems, while it is also computer storage-effective. It was also demonstrated to be more cost-effective, for the same storage demands, than double precision arithmetic calculations[10,11].

The Neumann-CG method (NCG)

The approximation of the inverse of the stiffness matrix using a Neumann series expansion has been used in the framework of stochastic finite element analysis, structural reanalysis and damage analysis problems. In all these cases the method was implemented on an "as is" basis, without any corrections to improve the quality of the solution, thus the results were satisfactory only in the vicinity of the initial design and unacceptable for large modifications of the stiffness matrix. In a recent study by Papadrakakis and Papadopoulos[13] the method was successfully integrated with the conjugate gradient algorithm resulting in a significant improvement on the accuracies achieved with low additional computational cost. It can also handle cases with significant changes of the stiffness matrix.

The solution of a typical reanalysis problem

$$(\mathbf{K}_0 + \Delta\mathbf{K})\,\mathbf{x} = \mathbf{b} \tag{25}$$

yields

$$\mathbf{x} = (I + \mathbf{K}_0^{-1}\Delta\mathbf{K})^{-1}\mathbf{K}_0^{-1}\mathbf{b}. \tag{26}$$

The term in parenthesis can be expressed in a Neumann expansion giving

$$\mathbf{x} = (I - P + P^2 - P^3 + \ldots)\mathbf{K}_0^{-1}\mathbf{b} \tag{27}$$

with $P = \mathbf{K}_0^{-1}\,\Delta\mathbf{K}$. The response vector can now be represented by the following series

$$\mathbf{x} = x_0 - Px_0 + P^2x_0 - P^2x_0 + \cdots \tag{28}$$

or

$$\mathbf{x} = x_0 - = x_1 + x_2 - x_3 + \cdots \tag{29}$$

The series solution can also be expressed by the following recursive equation

$$\mathbf{K}_0 x_i = \Delta\mathbf{K}x_{i-1} \quad i = 1, 2, \ldots \tag{30}$$

The advantage of this expression is that the stiffness matrix has to be factorized once while the additive terms x_i to the solution of equation (29) can be computed by successive backward and forward substitutions. The series may be truncated after a fixed number of terms or according to some error norm given by

$$\left\|x_i\right\| / \left\|\sum_{k=0}^{i}(-1)^k x_k\right\| \le \varepsilon_1 \tag{31}$$

or

$$\left\|r_i\right\| / \left\|b\right\| \le \varepsilon_2 \tag{32}$$

where $r_i = \mathbf{b} - (\mathbf{K}_0 + \Delta\mathbf{K})x_i$. The first criterion is most frequently used since it is computationally more efficient. The second criterion requires the evaluation of the residual vector which involves an additional matrix vector multiplication for each Neumann term.

In order to improve the quality of the preconditioning matrix, $\mathbf{M}$, used in the PCG method, a Neumann series expansion is implemented for the calculation of the preconditioned vector $z^{(m)} = \mathbf{M}^{-1}r^{(m)}$ of the PCG algorithm. The preconditioning matrix is now defined as the complete global stiffness matrix $\mathbf{K} = \mathbf{K}_0 + \Delta\mathbf{K}$, but the solution for z is performed approximately using a truncated Neumann series expansion. Thus, the preconditioned vector, $\mathbf{z}$, of the PCG algorithm is obtained at each iteration by

$$\mathbf{z} = z_0 - z_1 + z_2 - z_3 + \cdots \tag{33}$$

z_0 is given by

$$z_0 = \mathbf{K}_0^{-1}r_0 \tag{34}$$

and

$$\mathbf{K}_0 z_i = \Delta\mathbf{K}z_{i-1} \quad i = 1, 2, \ldots \tag{35}$$

The incorporation of the Neumann series expansion in the preconditioned step of the PCG algorithm can be seen from two different perspectives. From the PCG point of view an improvement of the quality of the preconditioning matrix is achieved by computing a better approximation to the solution of $\mathbf{x} = (\mathbf{K}_0 + \Delta\mathbf{K})^{-1}\mathbf{b}$ than the one provided by the preconditioning matrix $\mathbf{K}_0$. From the Neumann series expansion point of view, the inaccuracy entailed by the truncated series is alleviated by the conjugate gradient iterative procedure. It remains to be seen, however, whether the anticipated improvement on the convergence properties of the PCG method or of the Neumann series expansion implies also a reduction on the overall computational effort by counteracting the additional cost involved at each iteration.

The storage requirements for NCG in mixed precision arithmetic are the following: the compact stiffness $\mathbf{K}$ is stored in double precision arithmetic and the factorized skyline stiffness and $\Delta\mathbf{K}$ are stored in single precision arithmetic, while 2*[NA + (N + 1)] + (N + 1) short integers are needed for the addressings.

The preconditioned Lanczos method for multiple right-hand sides

When a sequence of right-hand sides has to be processed, direct methods possess a clear advantage over the conventional application of iterative methods. The major effort concerned with the factorization of the coefficient matrix is not repeated and only a back and forward substitution is required for each subsequent right-hand side. In the case of iterative methods the whole work has to be repeated from the beginning for every right-hand side.

Papadrakakis and Smerou[14] presented a new implementation of the Lanczos algorithm for solving linear systems of equations with a sequence of right-hand sides. This algorithm handles all approximations to the solution vectors simultaneously without the necessity for keeping in fast or secondary storage the tridiagonal matrix or the orthonormal basis produced by the Lanczos method. Thus, when the first solution vector has converged to a required accuracy, good approximations to the remaining solution vectors have simultaneously been obtained. It then takes fewer iterations to reach the final accuracy by working separately on each of the remaining vectors.

The equilibrium equations for multiple right-hand sides are stated as follows:

$$\mathbf{K}[x_1 \ldots x_k] = [\mathbf{b}_1 \ldots \mathbf{b}_k] \tag{36}$$

or

$$\mathbf{KX} = B \tag{37}$$

and the characteristic equations of the Lanczos algorithm become

$$T_j Y_j = Q_j^T R_o \tag{38}$$

and

$$X_j = Q_j Y_j \tag{39}$$

where Y_j = [y_1, …, Y_k]$_j$, X_j = [x_1, …, X_k]$_j$ consist of the jth approximation to the k auxiliary and solution vectors y and x respectively, and R$_o$ = [r_o^1, …, r_0^k] with r_o^i = b^i – Kx_0^i consists of the residual vectors. By using a Cholesky root-free decomposition of T_i we get

$$L_j D_j Z_j = Q_j^T R_o \tag{39}$$

$$X_j = C_j Z_j \tag{40}$$

with $\mathbf{Z}_j$ = [z_1, …, z_k] and $L_j C_j^T$ = Q_j^T. The last components of matrix $\mathbf{Z}_j$ are now given by

$$\zeta_{ji} = (q_j^T r_o^i - \delta_j d_{j-1} \zeta_{j-1,i}) / d_j \tag{41}$$

and the new approximation to the solution vectors by

$$[x_1 \ldots x_k]_j = [x_1 \ldots x_k]_{j-1} + c_j [\zeta_{jl} \ldots \zeta_{jk}]. \tag{42}$$

The algorithm can be stated as follows (where k is the number of right-hand sides, i is the right-hand side counter, j is the iteration counter):

```
Phase I. Initialization
  $r_1^{(1)} = b^{(1)} - \mathbf{Kx}_o^{(1)}$
  For $i = 2, \ldots, k$
    $x_0^{(i)} = M^{-1}[b^{(i)} - r_1^{(1)}]$.
  For $i = 2, \ldots, k$
      $r_1^{(i)} = b^{(i)} - \mathbf{Kx}_o^{(i)}$
      $\beta_1 = \sqrt{\left(r_1^{(1)^T} M^{-1} r_1^{(1)}\right)}$
      $q_1 = r_1^{(1)} / \beta_1$
      $q_o = c_o = 0 \quad d_o = \delta_1 = 0.$
Phase II. Calculate $\alpha_j$, $\beta_{j+1}$, $q_{j+1}$
  For $j = 1, 2, \ldots$
    $u_j = M^{-1} q_j$
    $\alpha_j = u_j^T K u_j$
    $r_{j+1} = K u_j - \alpha_j q_j - \beta_j q_{j-1}$.
```

$$\beta_{j+1} = \sqrt{r_{j+1}^T M^{-1} r_{j+1}}$$

$$q_{j+1} = r_{j+1} / \beta_{j+1}.$$

Phase III. LDL^T factorization of T_j

$$d_j = \alpha_j - \delta_j^2 d_{j-1}$$

$$\delta_{j+1} = \beta_{j+1} / d_j.$$

Phase IV. Calculate next $x^{(i)}$

For $i = 1, 2, \ldots, k$

$$\text{if } j = 1 \begin{cases} \text{if } i = 1 \rightarrow \zeta_{11} = \beta_1 / d_1 \\ \text{else } i \neq 1 \rightarrow \zeta_{1i} = M^{-1} q_1^T r_1^{(i)} / d_1 \end{cases}$$

$$\text{else } j \neq 1 \begin{cases} \text{if } i = 1 \rightarrow \zeta_{j1} = (\delta_j d_{j-1} \zeta_{j-1,1}) / d_j \\ \text{else } i \neq 1 \rightarrow \zeta_{ji} = (M^{-1} q_1^T r_1^{(i)} - \delta_j d_{j-1} \zeta_{j-1,i}) / d_j \end{cases}$$

$$c_j^T = M^{-1} q_j^T - \delta_j c_{j-1}^T$$

$$x_j^{(i)} = x_{j-1}^{(i)} + \zeta_{ji} c_j.$$

Phase V. Convergence check

$$\text{if } \frac{\left\| r_j^{(1)} \right\|}{\left\| r_1^{(1)} \right\|} < \varepsilon \Rightarrow \frac{\left| \zeta_{j1} \right| \left\| r_{j+1}^{(1)} \right\|}{\left\| r_1^{(1)} \right\|} < \varepsilon \text{ else go to Phase II.}$$

Phase VI. Continue iterations (separately) for the remaining $b^{(i)}$ (i = 2, …, k) with the PCG algorithm. The initial vectors $x_0^{(i)}$ for PCG are those finally obtained from phase IV.

The Lanczos method with multiple right-hand sides is applied in shape optimization with SA sensitivity analysis, while the preconditioning techniques used are identical to those used in the PCG method.

Numerical tests

The performance of the proposed solution methods is demonstrated and compared with the conventional direct skyline solver in a number of benchmark test examples. The convergence tolerance of the hybrid solution methods varies according to the problem. In topology optimization the convergence tolerance is taken 10^{-1} since only a crudely optimized distribution of the material of the initial structure is needed before the final refinement of the layout of the structure is performed during the shape optimization phase. In shape optimization, a higher level of convergence tolerance (10^{-3}) is chosen since the efficiency of the optimizer, as well as the quality of the final solution, is highly dependent on the attained accuracy of the sensitivity analysis subproblem. A renumbering scheme for bandwidth minimization is used after the mesh generation. For all test examples considered, plane stress conditions and isotropic material properties are assumed (elastic modulus E = 210,000N/mm^2 and Poisson's ratio ν = 0.3). All examples were run on a SGI Indigo R4000 workstation.

Topology optimization

For the test examples considered in topology optimization the design domain for the problems covers a large portion of the reference domain. Around areas of loading and support, a non-design domain is used to avoid taking into consideration high stress concentrations. The domains are discretized using a fine mesh in order to give a good resolution of the final topology, and to compute the stresses accurately. The initial values for the rejection rate, the evolution rate and the cut-off stress are taken as: ratre = 1 per cent, ratev = 1 per cent and strmx = 2,000 N/mm^2, the rest of the parameters governing the convergence history (maximum-minimum percentage of elements to be switched, etc.) are given for each problem separately. No element of growth is allowed. The switching-off of the elements in all examples is accomplished by dividing the elastic modulus for active elements by a factor of 10^5. The following abbreviations are used for the topology optimization prolems: Direct is the conventional skyline direct solver; PCG-n is the PCG solver in which the preconditioning matrix is formed with a complete Cholesky factorization and is updated with a refactorization when the number of PCG iterations becomes greater than *n*; NCG-*n* is the NCG solver using two terms in the Neumann series expansion in which a refactorization of the coefficient matrix is performed when the number of PCG iterations becomes greater than *n*.

Short cantilever beam problem. The short cantilever beam (Figure 2(a))[7] is clamped on one side and loaded at the top right-hand corner with a point load. The finite element mesh used is depicted in Figure 2(b) with 5,400 degrees of freedom (dof). A typical evolution history is shown in Figure 3. The basic control parameters for this example are chosen as follows: minimum and maximum number of elements to be switched off in every optimization step are taken as switl = 0.1 per cent and switu = 2 per cent, respectively. Convergence is achieved when all stress levels are within rspst = 75 per cent of the maximum stress or when the number of active elements is less than rspel = 25 per cent of the total number of elements. The evolution criterion is based on the principal stress, while composite elements are suppressed.

After convergence has been achieved, the rejection rate for the case of NCG-2 and NCG-3 reached 13 per cent and for the other solution methods reached 15 per cent. NCG-2 requires the smallest number of global optimization steps to

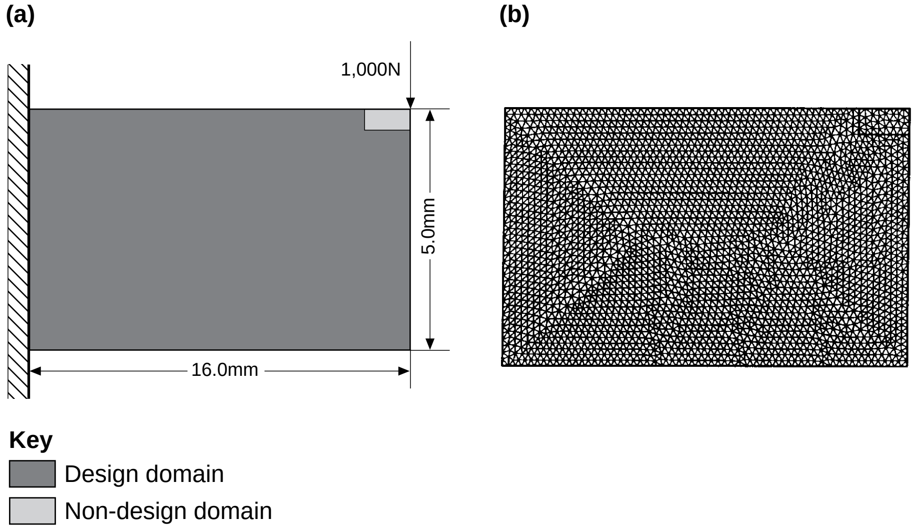


Figure 2.
Short cantilever beam:
(a) initial layout;
(b) initial mesh

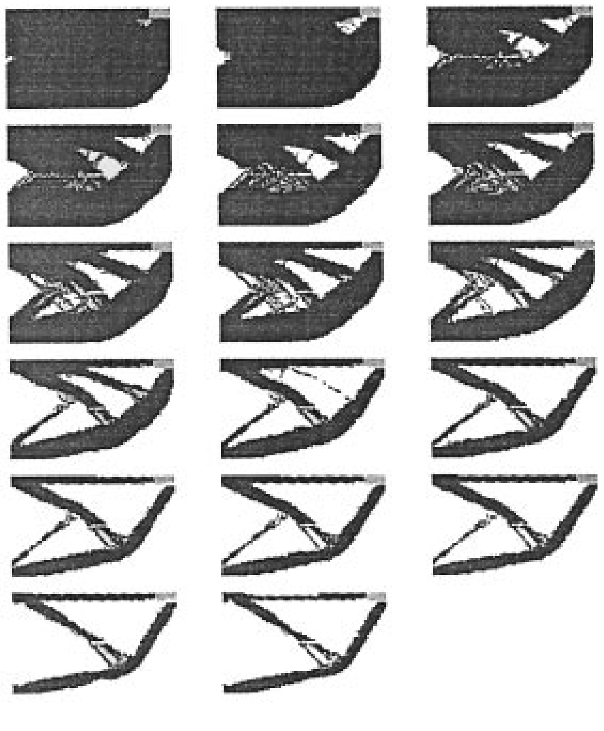


Figure 3.
Evolution history of topology optimization for the short cantilever beam problem

converge (Nsteps = 102), whereas PCG-5 takes the largest number of iterations to converge (Nsteps = 129). The resulting topologies for all cases are virtually identical, and can be interpreted as a four-bar truss. Figure 4 depicts the solution time, the total optimization time and the storage requirements. The

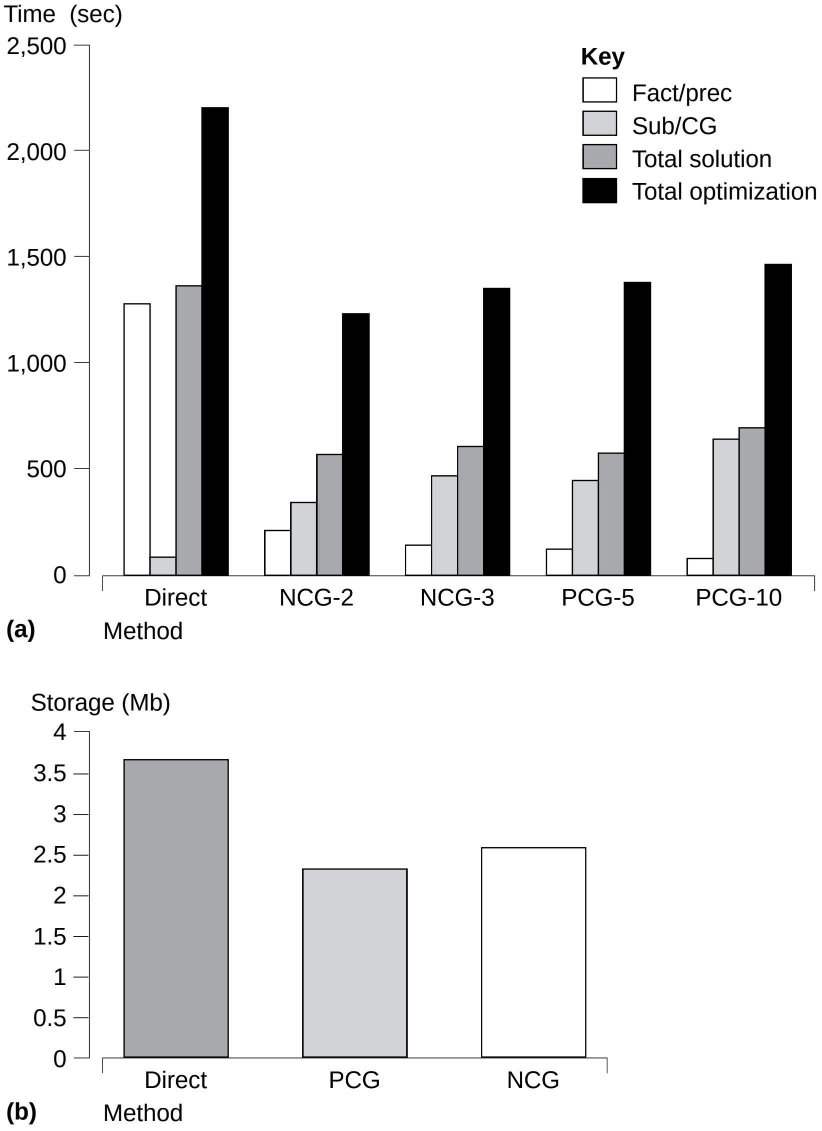


Figure 4.
Short cantilever beam – performance of methods: (a) computing time; (b) storage requirements

solution time is divided to factorization or preconditioning time (Fact/Prec) and substitution or PCG iterations time (Sub/CG). It can be observed that NCG-2 is 2.5 times faster than the direct method and reduces the overall optimization time by a factor of 50 per cent. In terms of computer storage, PCG requires 40 per cent and 10 per cent less storage than the direct and NCG methods, respectively.

Michell truss problem. The objective of this problem using the fully stressed design approach is to obtain a Michell-like truss structure. An analytical solution to this problem using Michell-like truss members was presented by Prager[15], while Bendsoe and Kikuchi[16] solved this problem using the homogenization method. The main difference between a Michell-like truss and a structure generated by a continuum approach, such as the homogenization or the fully stress design approach, is that the former structure is built assuming hinged joints, as opposed to rigidly connected joints produced by the latter method. The rectangular reference domain (Figure 5(a)) is clamped at the circumference of the circular cut-out. It is loaded by a tangentially acting, downward pointing point load which is applied at the centre of the right-hand edge. For the fully stressed design approach, the domain is discretized with 4,768 linear triangular elements with 5,000 dof (Figure 5(b)). A typical evolution history is shown in Figure 6. The basic control parameters for this example are chosen as follows: minimum and maximum number of elements to be switched off in every optimization step are taken as switl = 0.05 per cent and switu = 2 per cent, respectively. Convergence is assumed when all stress levels are within rspst = 65 per cent of the maximum stress or when the number of the active elements is less than rspel = 35 per cent of the total number of elements. The evolution criterion is based on the principal stress and composite elements are suppressed.

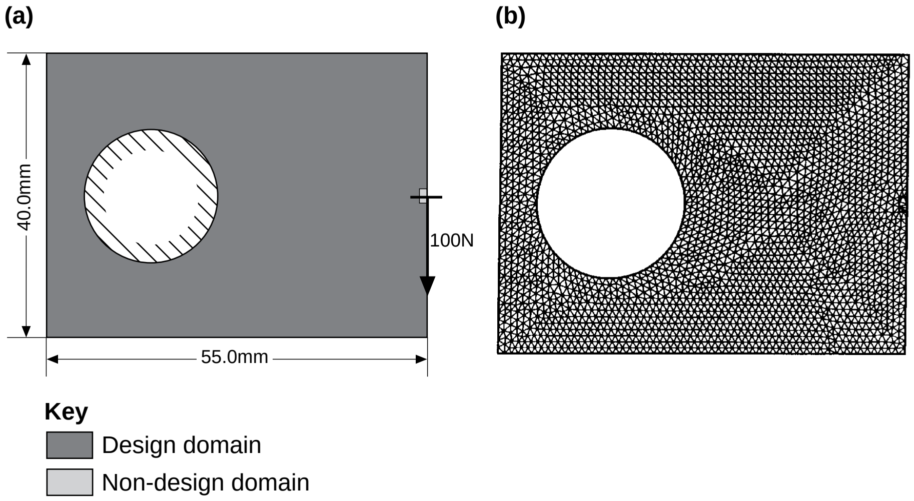


Figure 5.
Michell truss:
(a) intitial layout;
(b) initial mesh

After convergence has been achieved, the rejection rate for the direct method reached 13 per cent and for the other methods reached 17 per cent. NCG-3 requires the smallest number of optimization iterations (Nsteps = 106) to converge, whereas the direct solver requires the largest number of optimization steps to converge (Nsteps=132). In all cases the resulting topologies are virtually identical but they are unsymmetric and they cannot be interpreted as a Michell-like truss. Figure 7 depicts the solution time, the total optimization

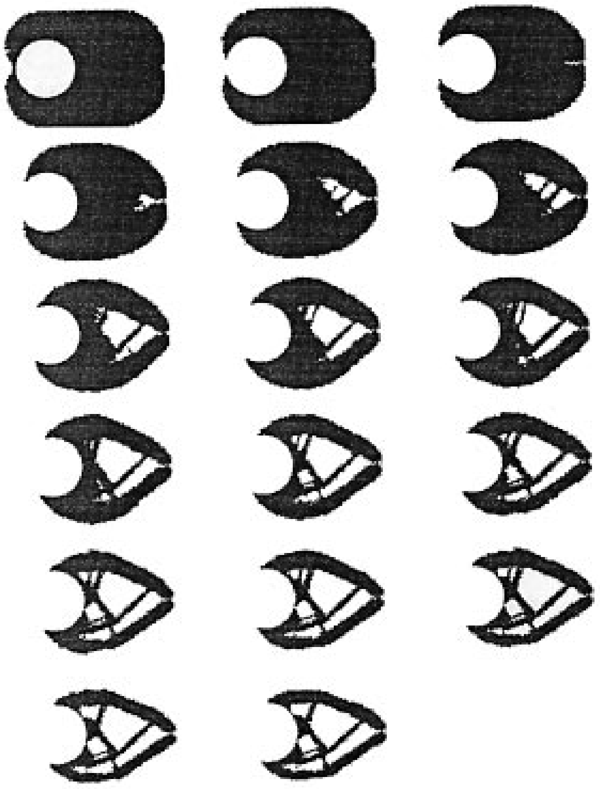


Figure 6.
Evolution history of topology optimization for the Michell truss problem

time and the storage requirements. It can be seen that NCG-2 is 3.5 times faster than the direct method and reduces the overall optimization time by a factor of 55 per cent. In terms of computer storage PCG requires 40 per cent and 10 per cent less storage than the direct and NCG methods, respectively.

Bridge problem. This problem is a thick beam fixed at the two bottom corners with a vertical point load applied at the centre of the bottom edge (Figure 8(a)). It was solved among others by Papalambros and Chirehdast[17] with the homogenization method, while Thomsen[18] presented a Michell-like solution for this problem. In this study the problem is solved using 5,833 elements (Figure 8(b)) with 6,000 dof. A typical evolution history is shown in Figure 9. The basic control parameters for this example are chosen as follows: minimum and maximum number of elements to be switched off in every optimization step are taken as switl = 0.05 per cent and switu = 2 per cent, respectively. Convergence is achieved when all stress levels are within rspst = 65 per cent of the maximum stress or when the number of active elements is less than rspel = 35 per cent of the total number of elements. The evolution criterion is based on the principal stress and composite elements are suppressed.

All analyses performed produced similar topologies which compare well with those obtained in references[17,18]. As the corner point is clamped there is

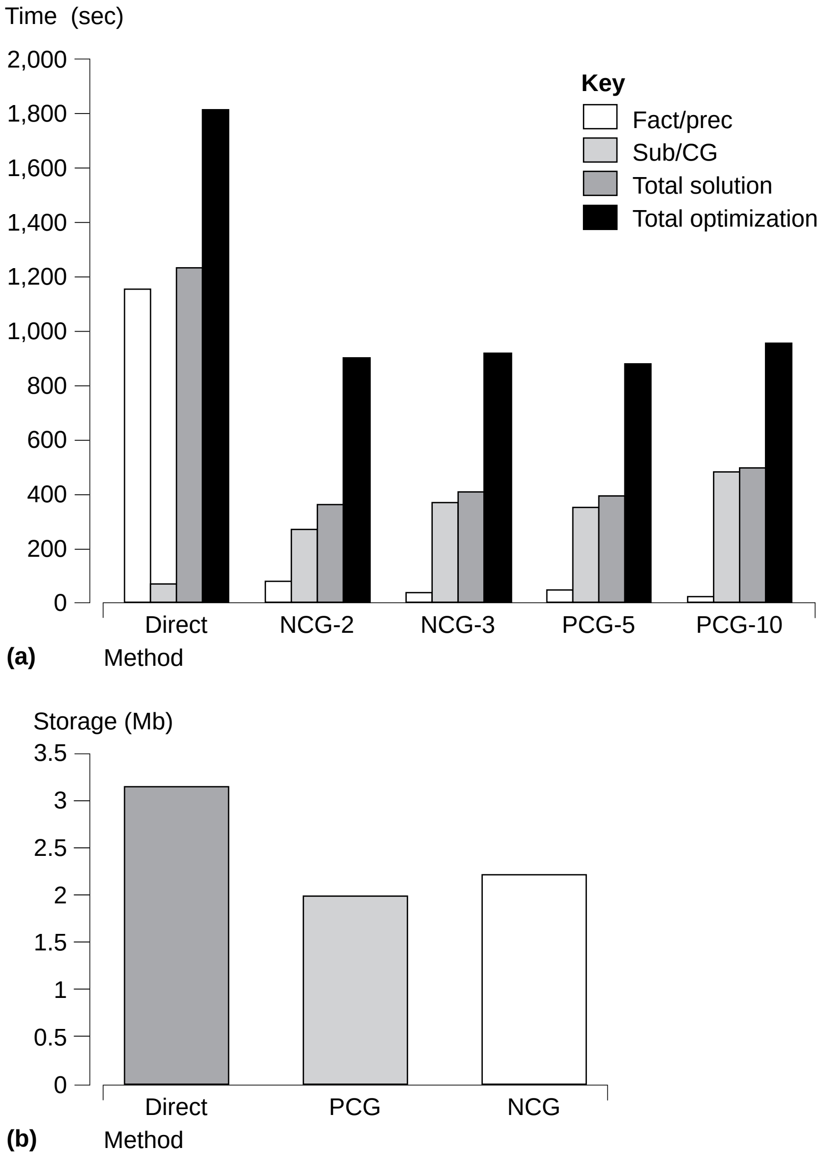


Figure 7.
Michell truss – performance of the methods: (a) computing time; (b) storage requirements

no mobility problem for the final layout, which can be interpreted as a four-bar truss. After convergence has been achieved, the rejection rate for all methods is 10 per cent. NCG-3 requires the smallest number of optimization steps (Nsteps = 89) to converge whereas the direct solver requires the largest number of optimization steps to converge (Nsteps = 109). Figure 10 depicts the solution time, the total optimization time and the storage requirements. It can be seen that NCG-2 is five times faster than the direct method and reduces the overall optimization time by a factor of 65 per cent. In terms of computer storage PCG

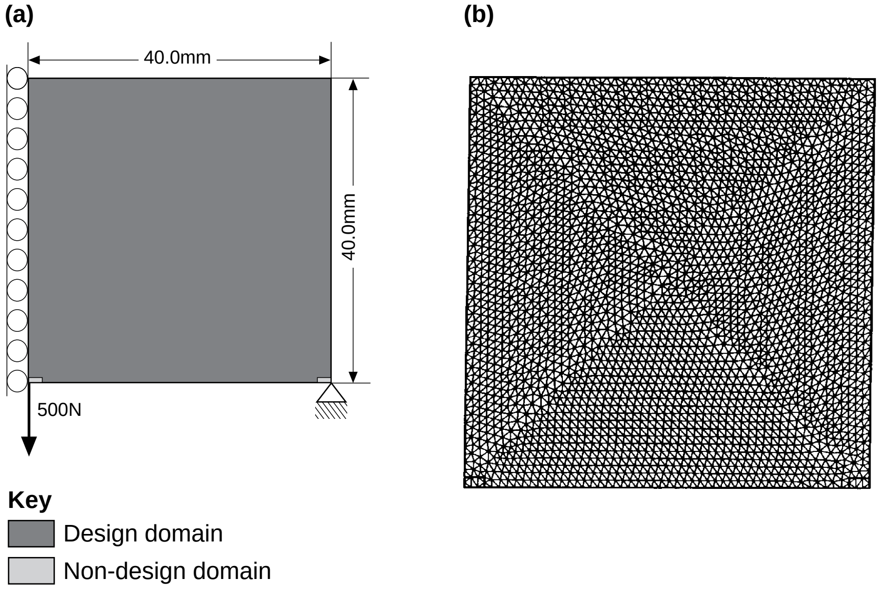


Figure 8. Bridge example: (a) initial layout; (b) initial mesh

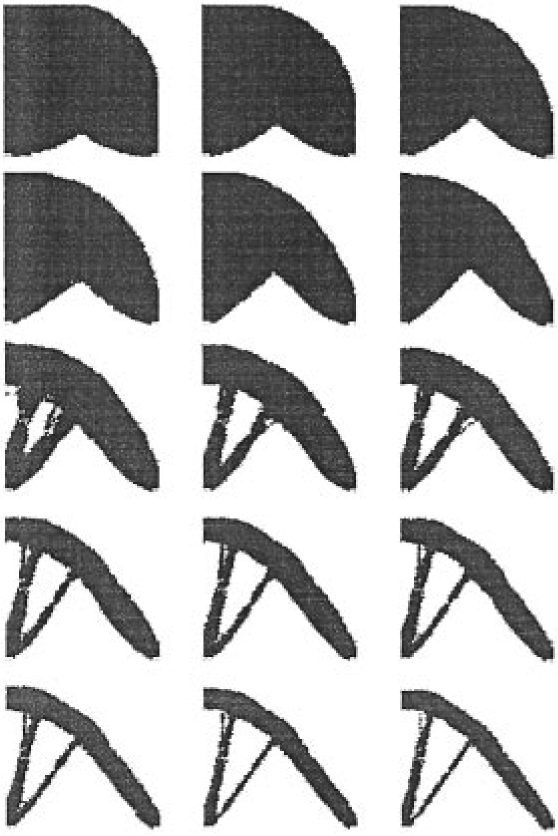


Figure 9. Evolution history of topology optimization for the bridge problem

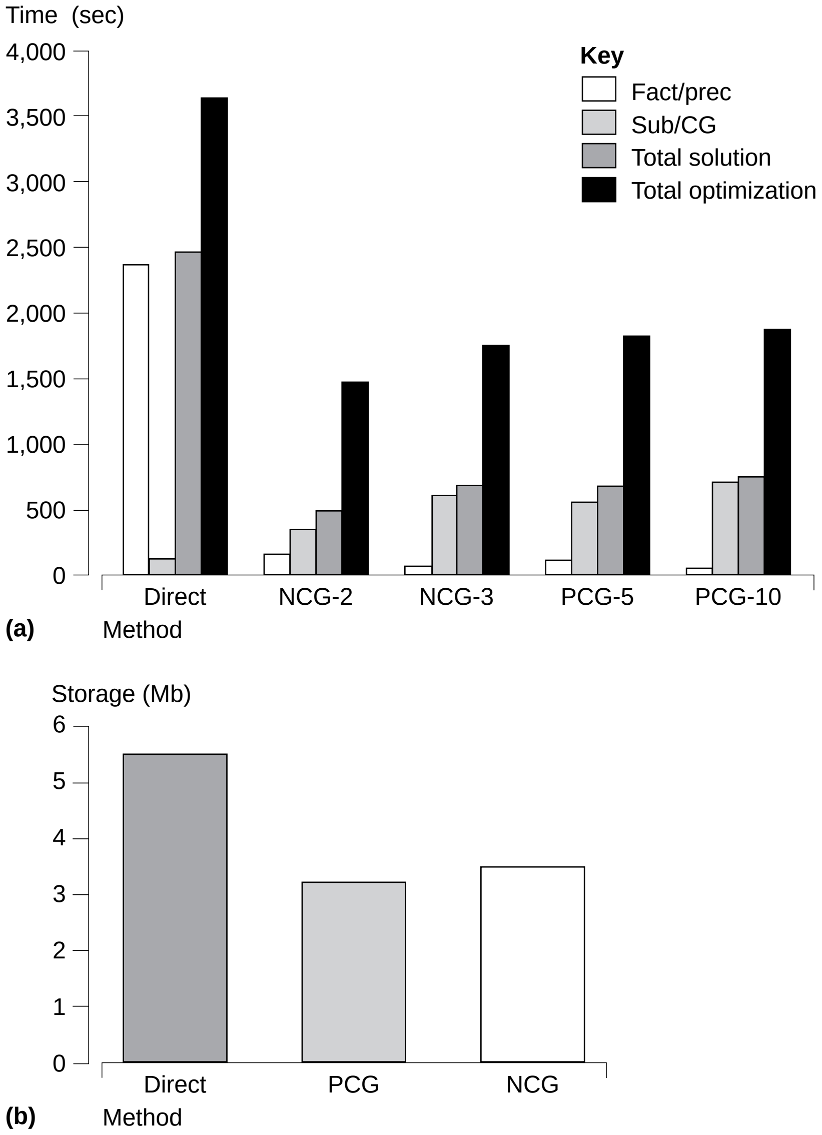


Figure 10.
Bridge example – performance of the methods: (a) computing time; (b) storage requirements

requires 40 per cent and 10 per cent less storage than the direct and NCG methods, respectively. Figure 11 depicts the convergence history of the hybrid methods showing the number of iterations per optimization step.

Shape sensitivity analysis

The test examples for the shape optimization are solved using a fine initial finite element mesh in order to compute the sensitivities accurately. The following abbreviations are used in this section: PCG(ψ) and Lanczos(ψ) are the

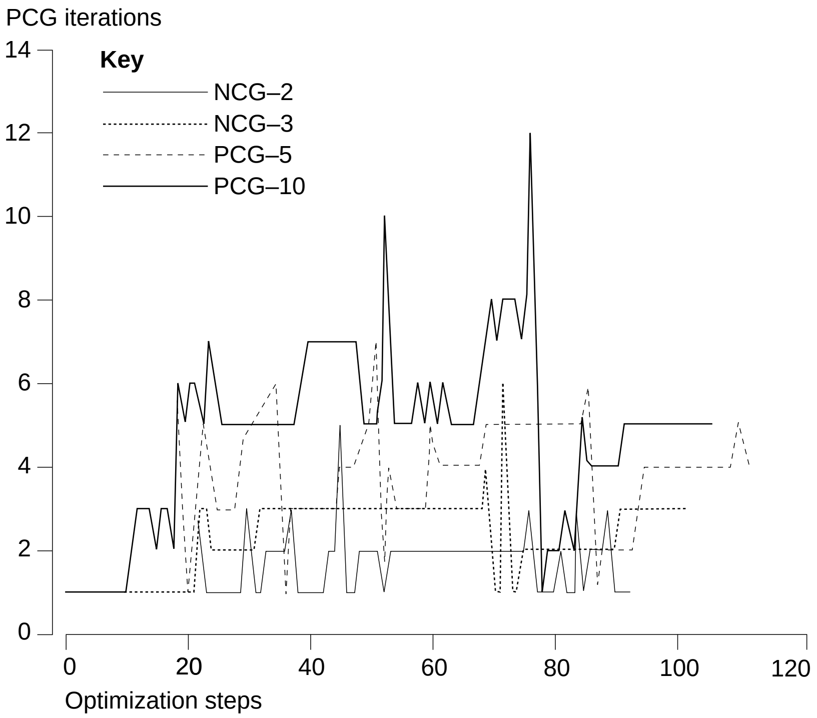


Figure 11. Bridge example: convergence history of the hybrid solution methods

PCG and Lanczos solvers respectively, with the preconditioning matrix produced via the rejection parameter ψ. A value of ψ between 0 and 1 corresponds to an incomplete Cholesky preconditioner, while $\psi = 0$ gives the complete factorized matrix. NCG-*i* is the NCG solver with *i* terms of the Neumann series expansion. The term function evaluation is related to the computations required for the finite element analysis and the sensitivity analysis for the calculation of the objective and constraint functions as well as their derivatives.

Connecting rod problem. In the following, the connecting rod example[19] is used to demonstrate the efficiency of the proposed solution methods in shape sensitivity analysis problems. This test case was originally solved by Kimmich[9]. The problem definition is given in Figure 12(a), and the final shape is shown in Figure 12(b). The linearly varying line load between key points 4 and 6 has a maximum value of p = 500N/mm. The objective is to minimize the volume of the structure subject to an equivalent stress limit of $\sigma_{\max}$ = 1,200N/mm^2. The design model, which makes use of symmetry, consists of 12 key points, four primary design variables (7, 10, 11, 12) and six secondary design variables (7, 8, 9, 10, 11, 12). The stress constraint is imposed as a global constraint for all the Gauss points and as key point constraint for the key points 2, 3, 4, 5, 6 and 13. The movement directions are indicated by the dashed arrows. Key points 8 and 9 are linked to 7 so that the shape of the arc is preserved

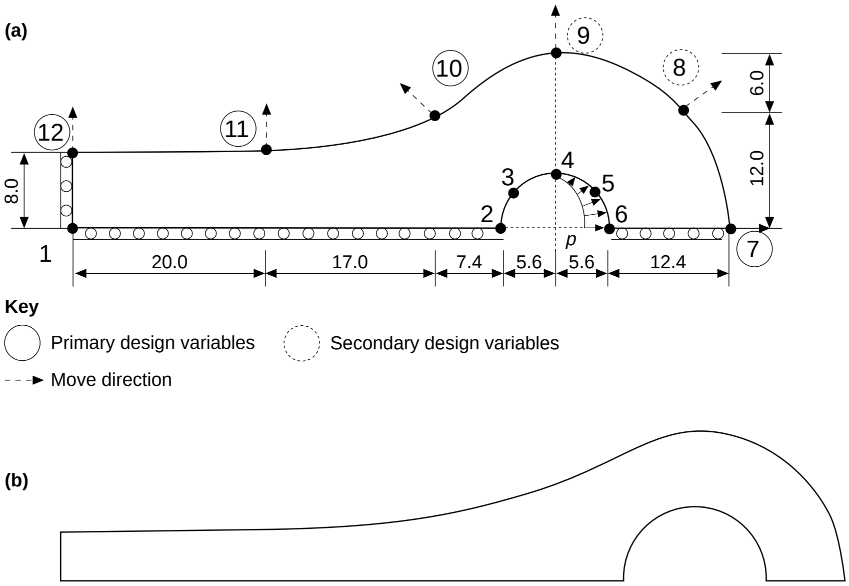
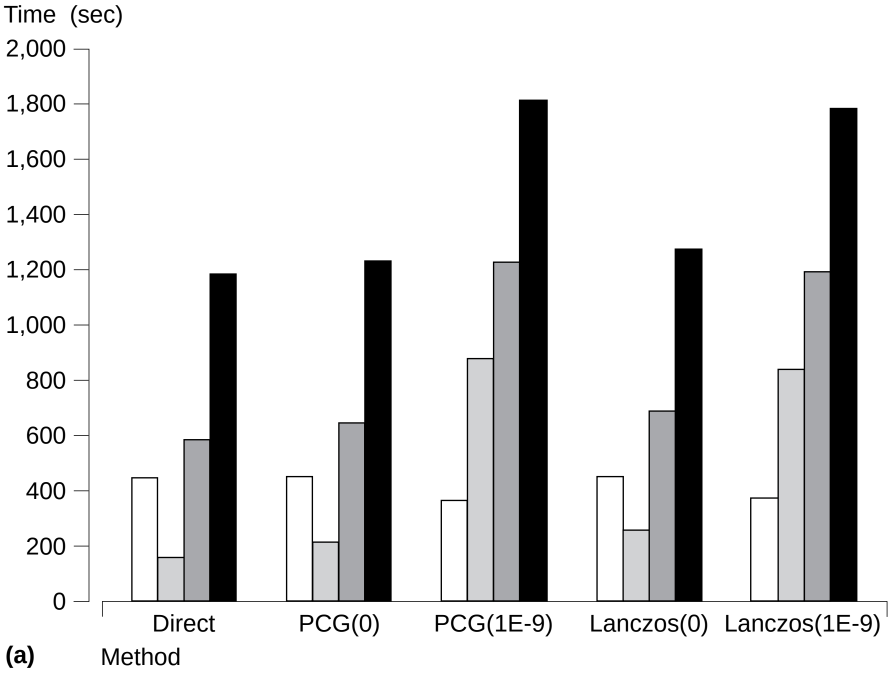


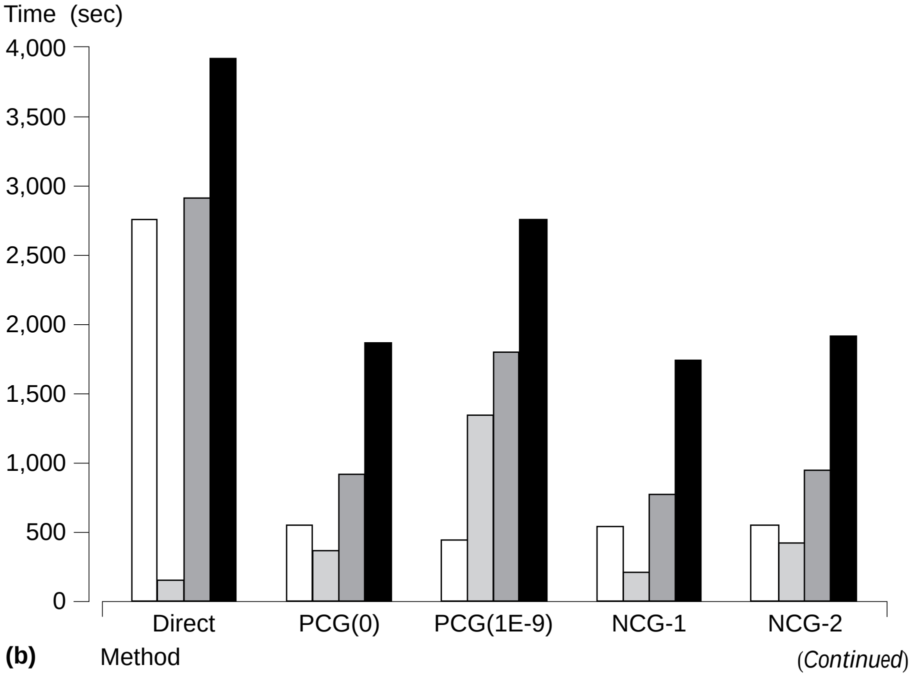
Figure 12.
Connecting rod:
(a) initial shape;
(b) final shape

throughout the optimization. The problem is analysed with a fine initial mesh having 13,000 dof. The optimization algorithm employed to solve the shape optimization problem is a SQP algorithm from the NAG library[20]. The ESA and the GFD sensitivity analysis methods are used to compute the sensitivities with perturbation value $\Delta s = 10^{-4}$.

The initial volume of the problem is 726 mm^3, which becomes 345.8 mm^3 at the end of the shape optimization procedure. Using the ESA approach the optimum is reached after seven optimization steps and 12 function evaluations, while with the GFD approach the optimum is reached after seven optimization steps and 15 function evaluations. Figure 13 depicts the solution time, the total optimization time and the storage requirements. In the case of ESA sensitivity analysis, the direct method is slightly faster than the most efficient hybrid solution methods PCG(0) and Lanczos(0), while in terms of storage requirements it has a 35 per cent overhead. For the GFD sensitivity analysis the NCG-1 method is almost four times faster than the direct solver and 15 per cent better than PCG(0) method analysis with regard to the solution time, and reduces by 55 per cent the overall optimization time. In terms of computer storage, NCG requires 25 per cent less storage than the direct method. Using a weaker preconditioner ($\psi = 10^{-9}$) reduces 15 per cent the storage but takes 30 per cent more time, in both types of sensitivity analysis, compared with the stronger preconditioner of $\psi = 0$.

Square plate with central cut-out problem. The square plate example with a central cut-out[19] is also used as a test case for comparison. The problem





(Continued)

Figure 13. Connecting rod – performance of the methods; (a) computing time for semi-analytical sensitivity analysis; (b) computing time for global finite difference sensitivity analysis; (c) storage requirements

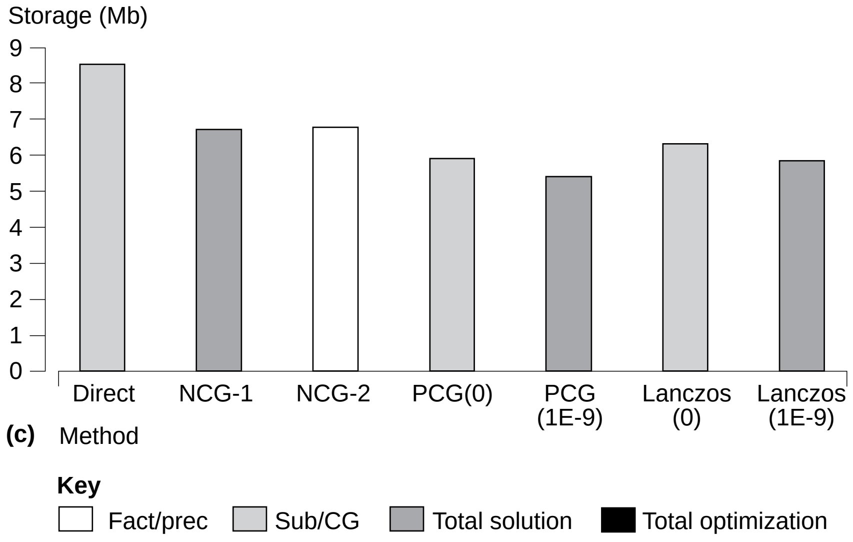


Figure 13.

definition is given in Figure 14a, where due to symmetry only a quarter of the plate is modelled, and the final shape is shown in Figure 14(b). The two exterior sides of the plate are loaded with a distributed loading p = 0.65N/mm^2, as shown in Figure 14(a). The objective is to minimize the volume of the structure subject to an equivalent stress limit of σ_{max}=7.0N/mm^2. The design model, consists of eight evaluations, five primary design variables (2, 3, 4, 5, 6) which can move along radial lines. The movement directions are indicated by the dashed arrows. The stress constraint is imposed as a global constraint for all the Gauss points and as key point constraint for the key points 2, 3, 4, 5, 6 and 8. The problem is analysed with a fine initial mesh with 11,000 dof. This problem is more sparse

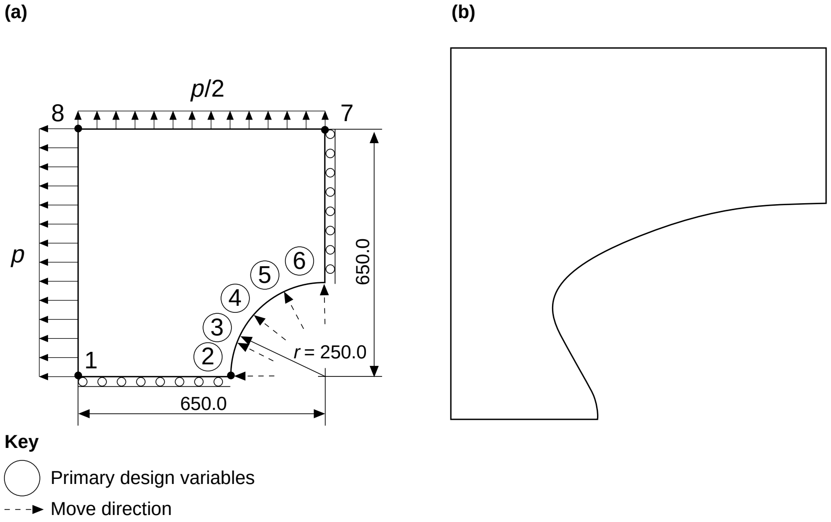
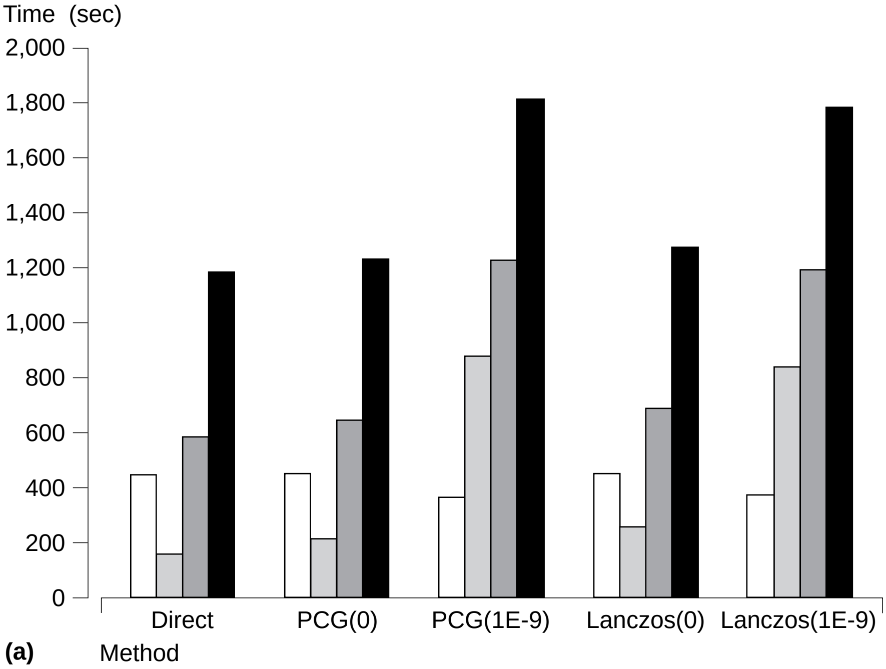
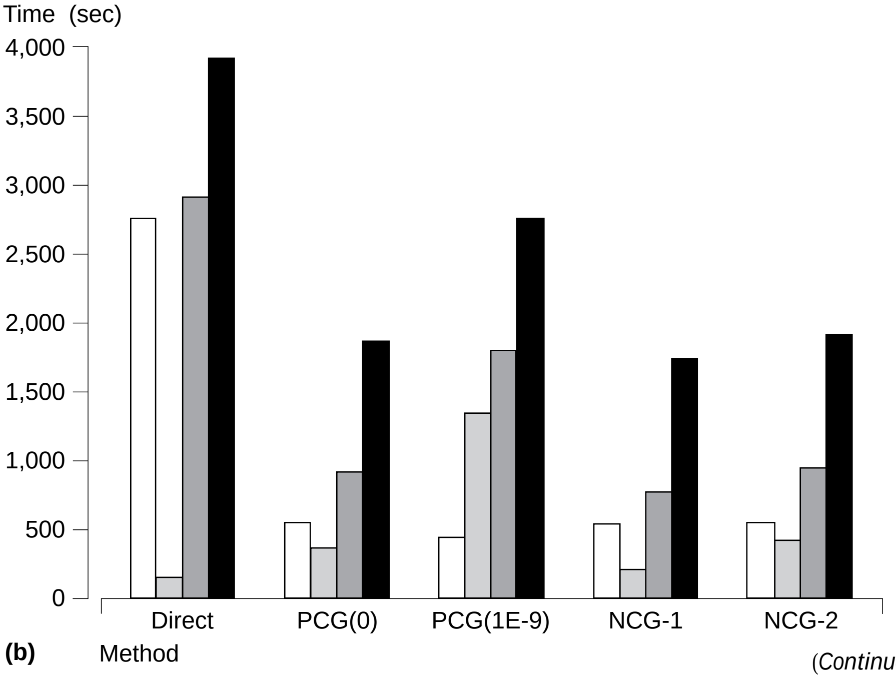


Figure 14.
Square plate:
(a) initial shape;
(b) final shape





(*Continued*)

Figure 15. Square plate – performance of the methods: (a) computing time for semi-analytical sensitivity analysis; (b) computing time for global finite difference sensitivity analysis; (c) storage requirements

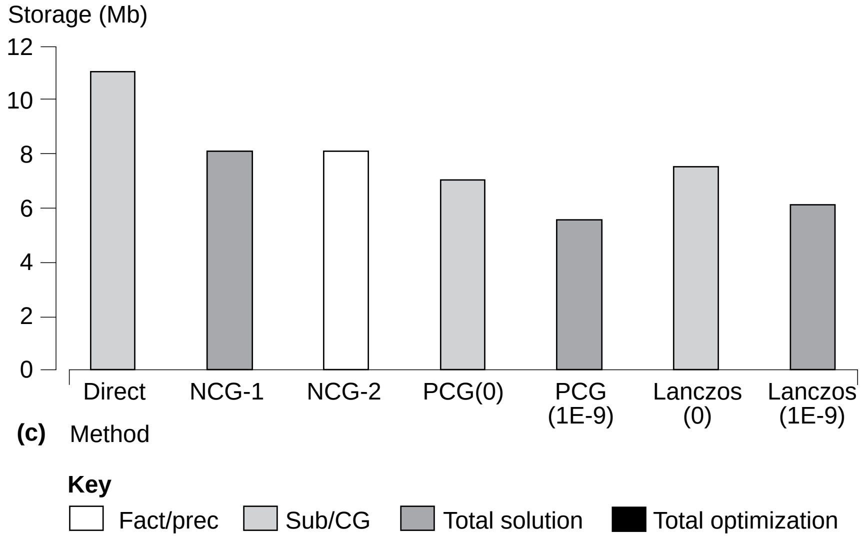


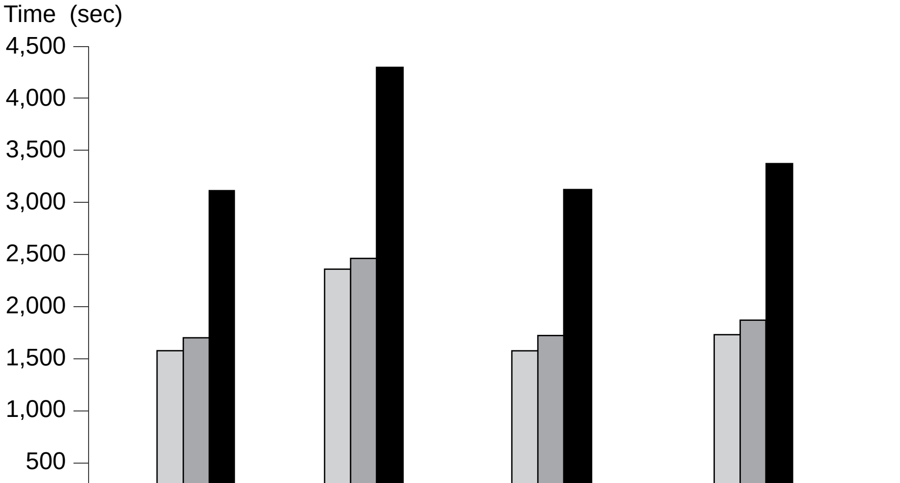
Figure 15.

and has more design variables than the connection rod example. The optimization algorithm employed to solve the shape optimization problem is a SQP algorithm from the DoT package[21]. The ESA and the GFD methods are used to compute the sensitivities with $\Delta s = 10^{-5}$.

The initial volume of the full problem is 149.37mm^3 which becomes 112mm^3 at the end of the shape optimization procedure. Using the semianalytical sensitivity analysis the optimum is reached after eight optimization steps and 14 function evaluations, while with the GFD sensitivity analysis method the optimum is reached after seven optimization steps and 12 function evaluations. Figure 15 demonstrates the solution time, the total optimization time and the storage requirements. In the case of ESA sensitivity analysis, the direct method is only 10 per cent faster, regarding the solution time, compared to the most efficient hybrid solution methods PCG(10^{-9}) and Lanczos(10^{-9}), while in terms of computer storage requirements it has a 50 per cent overhead. It can also be seen that a relatively weaker preconditioner ($\psi = 10^{-9}$) results in a 20 per cent saving in computer memory over the stronger preconditioner $\psi = 0$. Figure 16 shows the performance of Lanczos method compared to PCG when storage effective preconditioners are used with larger ψ values. For the case of GFD sensitivity analysis, NCG-1 and PCG(0) are five times faster than the direct solver with regard to the solution time and reduce by 65 per cent the overall optimization time, while in terms of computer memory PCG(0) is 40 per cent and 10 per cent better than the direct and the NCG method, respectively.

Discussion of the results and conclusions

For the two-dimensional test problems considered in topology optimization with regard to the solution time NCG proved to be 2.5 to 5 times faster than the direct skyline solver, and 10 per cent to 25 per cent faster than PCG. In terms of computer storage, PCG requires 40 per cent and 10 per cent less memory than the direct NCG methods, respectively. It is anticipated that in bigger problems, especially three-dimensional, the superiority of the proposed methods, both in computing time and storage, will be even more pronounced. With the PCG method it is advisable to use a strong preconditioner ($\psi = 0$) stored in single precision arithmetic since the number of PCG iterations remains small for many optimization steps and to recalculate the preconditioning matrix when the number of iterations becomes greater than 5 or 10. With the NCG method it is



(*Continued*)

Figure 16.
Square plate – performance of Lanczos and PCG methods for semi-analytical sensitivity analysis and six right-hand sides; (a) computing time; (b) CG iterations per right-hand side; (c) storage requirements

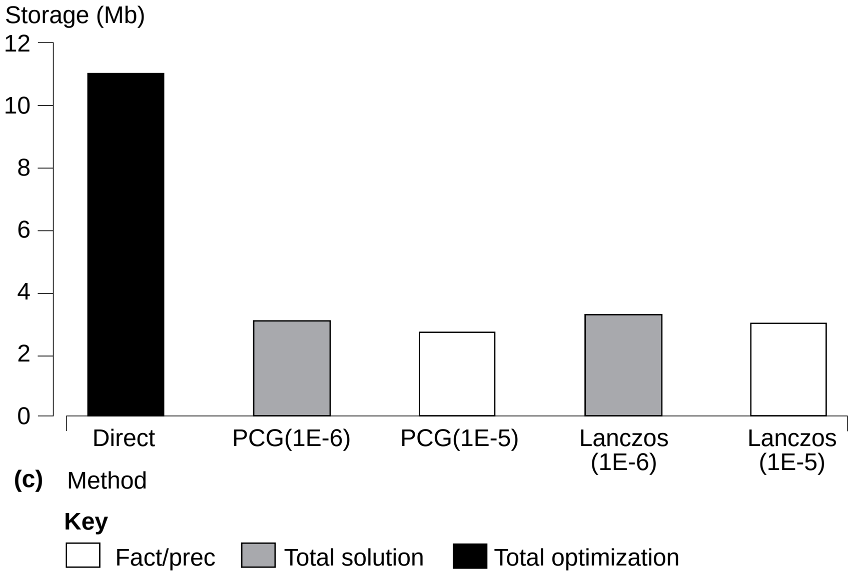


Figure 16.

more efficient, both for computational effort and accuracy of the results, to use two terms of the Neumann series expansion of the stiffness matrix, stored in single precision arithmetic, while it is more efficient to refactorize the stiffness matrix when the number of PCG iterations becomes greater than 2, or 3.

In the case of sensitivity analysis with the ESA approach the direct solver is at most 10 per cent better than PCG and Lanczos methods in terms of computing time, but requires 35 per cent to 50 per cent more storage. It is expected, however, that in large-scale three-dimensional problems with large bandwidth the computational performance of the hybrid solvers will be more competitive and even superior to those of the direct method. When using the PCG solver it was found that if a rather dense problem (connecting rod example) has to be solved it is more efficient to use a strong preconditioner ($\psi = 0$). But when a sparse problem (square plate example) is solved then it is more efficient to use a weaker preconditioner ($\psi = 10^{-9}$) in order to achieve additional memory savings. Using the Lanczos algorithm it was found that when a strong preconditioner ($\psi = 0$) is used no reduction in the number of iterations for the subsequent right-hand sides can be achieved, while using relatively weaker preconditioners a 40 per cent to 70 per cent reduction in the number of CG iterations is observed. For sparse problems with larger optimum values for ψ the superiority of the Lanczos method over the PCG method will be even more pronounced.

In the case of the GFD shape sensitivity analysis, the efficiency of the proposed methods depend on the size of the problem in terms of dof, and the number of design variables. The NCG-1 and PCG(0) methods are about four to five times faster in terms of solution time than the direct method and reduce by up to 65 per cent the total optimization time, while in terms of computer storage NCG and PCG require 25 per cent to 50 per cent less storage than the direct method. For large-scale problems and large number of design variables the superiority of the hybrid methods is expected to increase. Usually the GFD method produces oscillation around the optimum and thus it needs more optimization steps and function evaluations for convergence than the SA method. In the present study, it was observed that the use of fine finite element meshes alleviates the inaccuracies inherent in the GFD method and makes it perform similarly to the SA method. This observation, in conjunction with the improved performance of the hybrid solution methods in the GFD sensitivity analysis renders it the first choice for the calculation of sensitivities, since it is simpler and easier to implement than the other methods.

References

1. Hinton, E. and Sienz, J., "Fully stressed topological design of structures using an evolutionary procedure", *Engineering Computations,* Vol. 12, 1993, pp. 229-44.
2. Eschenauer, H.A. and Schumacher, A., "Simultaneous topology and shape optimization", in Rozvant, G.I.N. and Olhall, N. (Eds), *Proceedings of the First World Congress of Structural Optimization,* Goslar, Germany, May 1995, Pergamon Press, London.
3. Maute, K. and Ramn E., "General shape optimization – an integrated model for topology and shape optimization", *Proceedings of the First World Congress of Structural Optimization,* Goslar, Germany, May, 1995, Pergamon Press, London.
4. Rozvany, G.I.N. and Zhou, M., "Layout and generalised shape optimization by iterative COC methods", in Rozvany, G.I.N. (Ed.), *NATO/DFG ASI Optimization of Large Structural Systems, Berchtesgaden, Germany,* Kluwer Academic Publishers, Dordrecht, 1993, pp. 103-20.
5. Bendsoe, M.P. and Kikuchi, N., "Generating optimal topologies in structural design using a homogenization method", *Computational Methods in Applied Mechanical Engineering,* Vol. 71, 1988, pp. 197-224.
6. Eschenauer, H.A. Schumacher, A. and Vietor, T., "Decision makings for initial designs made of advanced materials", in Bendsoe, M.P. and Soares, C.A.M. (Eds), *NATO ARW "Topology Design of Structures", Sesimbra, Portugal,* Kluwer Academic Publishers, Dordrecht, 1993, pp. 469-80.
7. Xie, Y.M. and Steven, G.P., *Optimal Design of Multiple Load Case Structures Using an Evolutionary Procedure,* FEARC-9205, Finite Element Analysis Research Centre, University of Sydney, Australia, 1992.
8. Hinton, E. and Sienz, J., *Adaptive Finite Element Analysis and Shape Optimization,* Saxe-Coburg Publications, Edinburgh, 1996.
9. Kimmich, S., "Strukturoptimierung und Sen mit finiten Elementen", PhD thesis, Bericht Nr. 11, Institut für Baustatik der Universtät Stuttgart, Germany, 1990.
10. Olhoff, N., Rasmussen, J. and Lund, E., *Method of Exact Numerical Differentiation for Error Estimation in Finite Element Based Semi-analytical Shape Sensitivity Analyses,* Special Report No. 10, Institute of Mechanical Engineering, Aalborg University, Aalborg, 1992.
11. Papadrakakis, M. and Bitoulas, N., "Accuracy and effectiveness of preconditioned conjugate gradient method for large and ill-conditioned problems", *Computational Methods in Applied Mechanical Engineering,* Vol. 109, 1993, pp. 219-32.

12. Bitoulas, N. and Papadrakakis, M., "An optimised computer implementation of the incomplete Cholesky factorization", *Computer Systems in Engineering,* Vol. 5 No. 3, 1994, pp. 265-74.

13. Papadrakakis, M. and Papadopoulos, V., "Efficient solution procedures for the stochastic finite element analysis of space frames using the Monte Carlo simulation", *Computational Methods in Applied Mechanical Engineering,* forthcoming.

14. Papadrakakis, M. and Smerou, S., "A new implementation of the Lanczos method in linear problems", *International Journal for Numerical Methods in Engineering,* Vol. 29, 1990, pp. 141-59.

15. Prager, W., "Optimization in structural design", in Bellman, R., *Mathematical Optimization Techniques,* RAND Corporation Report P-396-PR, 1963, USA, pp. 279-89.

16. Bendsoe, M.P. and Kikuchi, N., "Topology and generalized layout optimization of elastic structures", in Bendsoe, M.P. and Soares, C.A.M. (Eds), *NATO ARW "Topology Design of Structures", Sesimbra, Portugal*, Kluwer Academic Publishers, Dordrecht, 1993, pp. 159-205.

17. Papalambros, P.Y. and Chirehdast, M., "Integrated structural optimization system", in Bendsoe, M.P. and Soares, C.A.M. (Eds), *NATO ARW "Topology Design of Structures" in Sesimbra, Portugal,* Kluwer Academic Publishers, Dordrecht, 1993, pp. 501-14.

18. Thomsen, J., *Optimerung af anisotrope materialers egenskaber og konstruktioners topologi*, Special Report No. 14, Institute of Mechanical Engineering, Aalborg University, Denmark, 1992.

19. Hinton, E. and Sienz, J., "Studies with a robust and reliable structural shape optimization tool", in Topping, B.H.V. (Ed.), *Developments in Computational Techniques for Structural Engineering,* Civil-Comp Press, Edinburgh, 1995, pp. 343-58.

20. NAG, *Software Manual,* NAG, Oxford, 1988.

21. DoT, *User Manual,* VMA Engineering, USA, 1994.