7th World Congress on Structural and Multidisciplinary Optimisation
COEX Seoul, 21 May – 25 May 2007, Korea

Smooth Boundary Based Optimisation Using Fixed Grid

Caroline S Edwards[1], H Alicia Kim[1], Chris J Budd[2]

[1]Department of Mechanical Engineering, University of Bath, United Kingdom. C.S.Edwards@bath.ac.uk
[2]Department of Mathematical Sciences, University of Bath, United Kingdom.

1. Abstract
This paper presents a boundary based structural optimisation in the finite element fixed grid environment. The boundary is represented by smooth B-spline curves, as they typically require a relatively small number of control points to describe a curve. Utilising B-spline control points reduces the number of design variables whilst achieving a smooth boundary representation. An appropriate control point coordinate modification formulation with addition/removal of control points, derived from the optimality criteria of compliance based optimisation, is introduced in the paper. The paper also examines a range of structural analysis solvers appropriate for frequent boundary modifications in topology optimisation. Previous studies have found that iterative pre-conditioned conjugate gradient solvers with appropriate pre-conditioners were most efficient for analysis and reanalysis. Whilst it has been shown that LU decomposition of the stiffness matrix can offer a fast convergence for reanalysis, it cannot guarantee convergence. The performance of the pre-conditioned conjugate gradient solver for optimisation is carefully studied with some of the latest direct solvers developed for elasticity and FE problems. The careful implementation of the analysis solver together with the reduced number of design variables achieves an efficient and reliable structural optimisation scheme. This is demonstrated in the paper numerically via simple shape optimisation problems, followed by preliminary results for topology optimisation.

2. Keywords: Fixed grid, Boundary optimisation, Topology optimisation, B-splines

3. Introduction
In a fast and continually growing world where computational power is ever increasing, the need for efficiency continues to be prevalent. Structural topology optimisation is a research field which is heavily reliant on computational power. Structural topology optimisation also needs to be reliable, which currently is heavily reliant on numerical parameters to avoid instabilities, whose appropriate values can sometimes be transparent [1, 2].
Reliability issues surrounding the results of topology optimisation are largely due to the coupling of the optimisation method with the finite element (FE) mesh. One such occurrence is the inability of a method to produce a discrete solution. Solid isotropic microstructure with penalisation (SIMP) [3], [4], [5] is a topology optimisation method which uses the continuous material distribution of the problem as the design variable. Problems with continuous design variables tend to result in optimal designs which are also continuous. Thus to encourage a discrete design, penalties are applied to the continuous design variables. However, this introduction of penalisation can introduce multiple local optima [6], [2].
Another reliability issue with topology optimisation methods is the dependency on the shape of the finite elements. A simple and thus popular element choice is 4-node quadrilateral elements which take the shape of a square. The square shape of the element can give rise to re-entrant corners, which in turn causes numerical instabilities. Evolutionary structural optimisation (ESO) is one method which in particular suffers from problems with re-entrant corners [7], [2]. Also, the combination of piecewise linear elements and piecewise constant material design variables can result in numerical instabilities [8]. These numerical instabilities can produce an undesirable formation of chequerboard patterns of solid and void material within the optimal topology [9]. Various methods have been suggested to overcome numerical instabilities; such as the use of filtering techniques to smooth out local sensitivity concentrations [10].
Recently, a number of topology optimisation methods were introduced which attempt to decouple optimisation from the FE formulation. One such method combines a concept known as fixed grid (FG) [11] for the finite element analysis with the topology optimisation method ESO [12]. FG is a mesh free FE method where the structural boundaries are superimposed onto an FE mesh. As in previous topology optimisation methods, the stiffness of elements that are inside or outside of the boundary is altered to reflect the solid or void nature of the element. The stiffness of the elements on the boundary is also altered to reflect the bi-material nature of the element.
The implementation of FG with ESO predominantly uses boundary points joined linearly, in which one boundary point is defined per element, to model a boundary for a structure [12]. Therefore, the resulting topology is not completely smooth and a vast number of design variables is used to modify the boundary.
The extended finite element method (XFEM) [13] is also a method which can model holes within a structure without the need to rebuild the mesh. Originally designed to model crack growth, XFEM enriches the FE approximation using additional functions. The additional functions model the cracks and holes within the structure.

Another method which decouples optimisation from the FE mesh is the use of level sets [14] in conjunction with shape optimisation to model the boundaries of a structure [15], [16]. The level set formulation produces smooth boundaries; however, changing the topology of structure is not a straight forward task and is thus an active field of research. One emerging idea is the use of topological derivatives [17] to calculate potential hole locations and combining them with the level set optimisation [18], [19], [20].
The aim of this paper is to develop an efficient and reliable topology optimisation method by using FG. In addition to the efficiency of FG in reanalysis, a range of direct solvers are investigated to determine the most suitable solver for optimisation with fast solution time and reliable convergence. The efficiency of optimisation is also addressed by implementing B-spline control points as design variables thereby reducing the number of design variables.
The outline of this paper is as follows: the next section details the FG approximation. Section 5 contains details of the test problems which are used within this paper both for the linear solver investigation and for application of the optimisation method. The linear solver investigation is also performed in section 5. The problem formulation and shape optimisation algorithm is outlined in section 6. Section 7 contains the application of the boundary based optimisation method to the previously defined test problems. Section 8 outlines modifications to the shape optimisation algorithm to incorporate topology optimisation along with the preliminary results, which is followed by conclusions.

4. Fixed Grid Approximation

In the FE environment, a set of n elements over a design domain Ω is used to represent a structure. Fixed grid is used within this FE environment to quickly solve linear elasticity problems such as (1), for global stiffness matrix $\mathbf{K}$, load vector $\mathbf{f}$ and displacement vector $\mathbf{u}$.

$$\mathbf{Ku} = \mathbf{f} \tag{1}$$

Fixed grid is different from the traditional FE approach in that a structure is represented by a rectangular area of regular sized elements rather than the fitted mesh, as shown in Figure 1. Each element within the domain is assigned material properties which reflect the status of the element. There are three types of element which are also shown in Figure 1: the first is the inside element (I), where all 4 nodes of an element are contained within the boundaries of the structure; the second type of element is the outside element (O), where all 4 nodes of the element are outside the structures boundaries; the third element type is on the boundary and is partially inside (B).

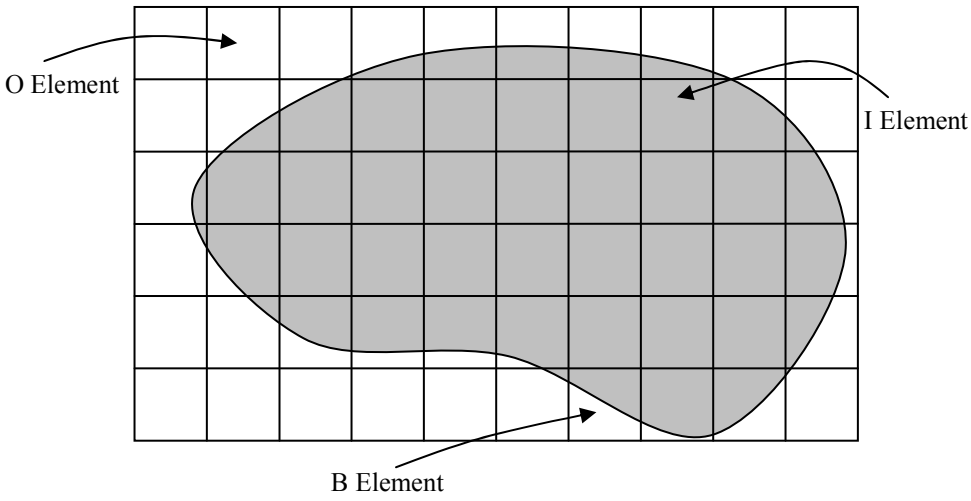


Figure 1: Fixed grid domain with superimposed topology in grey, showing inside (I), outside (O) and Boundary (B) elements

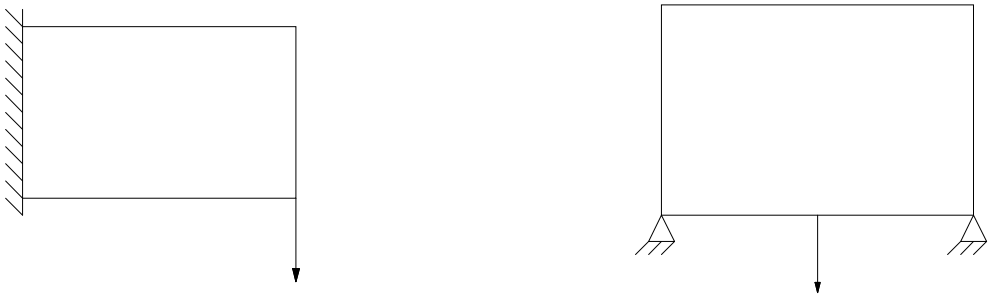
The boundary elements are modelled as a linear homogenisation of inside and outside elements based on their respective area ratios. Whilst the model will reduce the accuracy of the FE approximation, the modelling is simple to implement and the stiffness matrix $\mathbf{K}$ is easy to update without the need to completely regenerate. Also, since the FG elements can have partially inside elements, the design variable of elemental density is no longer piecewise linear, thus avoiding the numerical instabilities such as the chequerboard phenomenon, where the optimum topology contains a chequerboard pattern of solid and void material. Thus FG is ideally suited for structural optimisation [11].
The continuous variable x is used to represent the linear area ratio of material. For an element e, the element stiffness $\mathbf{K}_e$ can be written in terms of $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ as

$$\mathbf{K}_e = x_e \mathbf{K}_0, \tag{2}$$

where $x_e \in (\delta,1]$, δ is a value that is close to 0, and $\mathbf{K}_0$ is the element stiffness constant with $x_e = 1$. For an inside (I) element, x_e takes the value '1' and for outside (O) elements, x_e takes the value 'δ' to maintain a non-singular stiffness matrix. Boundary (B) elements have $x_e = a_B$ where a_B is the area ratio of the B element.

5. Investigation of Linear Solvers

The regularity of FG offers a unique opportunity for the exploitation of a consistent structure of the stiffness matrix. A fixed grid with sequentially ordered elements will provide a consistent connectivity between structural degrees of freedom. Since the fixed grid is independent from the geometry of a structure, the stiffness matrix **K** will have a virtually constant structure. Thus the efficiency of a number of linear solvers is investigated using the design domains of a short cantilevered beam of aspect ratio 1.6 and the popular 'Michell' type structure of aspect ratio 2 [21] shown in Figure 2.

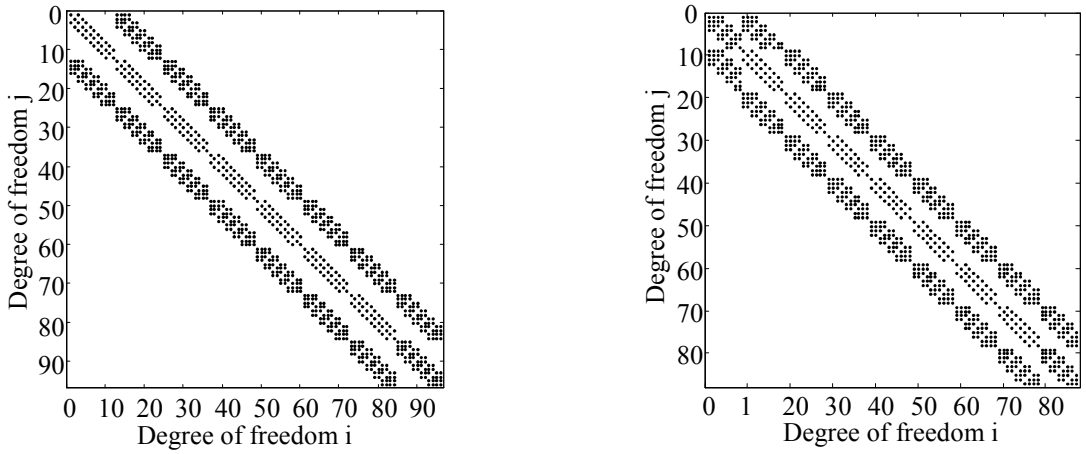


(a) Short cantilever (b) 'Michell' type structure

Figure 2: Test problems

5.1. Stiffness matrix structures of test problems

The connectivity of the stiffness matrix in FG is only dependent on the number of elements. Therefore, the structure of the stiffness matrices associated with the test problems can be identified, Figure 3.



(a) Short cantilever beam (b) 'Michell' type structure

Figure 3: Graphical representation of the structure of the FG stiffness matrix. The black dots represent non-zero entries

From Figure 3, it is clear that FG stiffness matrices are of a similar structure. Both matrices are tri-diagonally banded matrices which are vastly sparse, but the similarities are much more than that. The maximum band size of each diagonal is the same, the layout of the outer diagonal entries is the same and the central diagonal entries generally take the same layout with any differences caused through the boundary conditions. Thus the linear solvers to be investigated are those suitable for sparse, banded, positive definite matrices.

5.2. Frontal Solvers

Frontal solvers are direct solvers which have their origins in the solution of FE problems [22] and exploit the sparse nature of a matrix such as the stiffness matrix. However, they can also be applied to fully assembled matrices. They use the

connectivity information of the finite elements to efficiently implement Gaussian elimination. The Gaussian operation is performed once all of the terms for a degree of freedom have been entered and the matrix has been permuted so that the entry is in the next pivotal position. Therefore the Gaussian elimination and the assembly of **K** can be interleaved and so the matrix is never assembled explicitly. The degrees of freedom that have entries in the stiffness matrix **K** but have not yet been eliminated are referred to as active elements and form a front in the symmetrically permuted stiffness matrix; hence the technique is called the frontal method.

5.3. Multifrontal solver

Whilst the frontal solver is assembled by adding one element stiffness matrix $\mathbf{K}_e$ to the global stiffness matrix **K** at a time, the multifrontal method assembles element stiffness matrices in pairs. Then the bi-element stiffness matrices are assembled in pairs and so forth. This method then has multiple fronts, hence the name and keeps the size of the fronts to a minimum. Less computation is required to assemble matrices with smaller fronts, thus producing a more efficient linear solver.

5.4. Band Solver

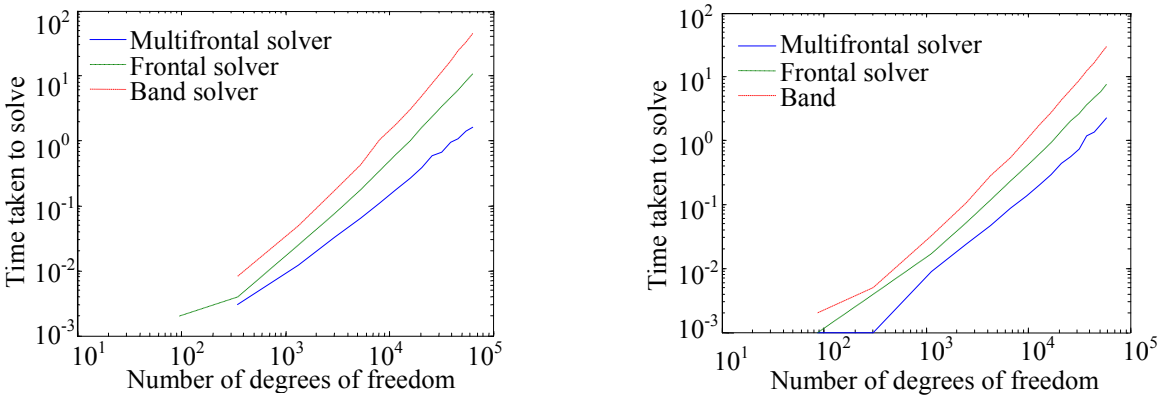
Finally, a solving method which exploits the banded nature of FG stiffness matrices is considered. When the finite elements are numbered in an efficient manner, the resulting stiffness matrix can be banded with a relatively small band size. When the maximum band size along with the band size for each row is known *a priori*, as it can be for the stiffness matrix, the matrix structure can be exploited when attempting to solve.

A popular method for solving banded linear systems with non-zeros pivots is to decompose the matrix, such as the stiffness matrix, into a lower triangular matrix **L** and an upper triangular matrix **U**. This method is referred to LU decomposition. However, when the matrix is symmetric and positive definite, like a stiffness matrix **K**, the factors of **U**, can be chosen such that they are those of the transpose of **L**. This special case of matrix decomposition is referred to as Cholesky factorisation and is twice as efficient as LU decomposition as only half of the number of entries is required. Cholesky factorisation is a suitable factorisation method for banded matrices, even when the actual band width varies as it does with the stiffness matrix, since the lower triangular matrix **L** will have the banded structure of the matrix to be decomposed and the only entries that are required to be calculated are those within the row band.

HSL libraries (MA62, MA57 and MA55) using an Intel® Fortran complier are used for this investigation [23].

5.5. Application of Linear Solvers to Test Problems

The mesh density of both the short cantilever and the 'Michell' type structure is varied considerably in the first instance. The average CPU times taken to solve the linear system (1) in each case are recorded and are presented in Figure 4.



(a) Short cantilever (b) 'Michell' type structure

Figure 4: Variation of solution time for increasing numbers of degrees of freedom

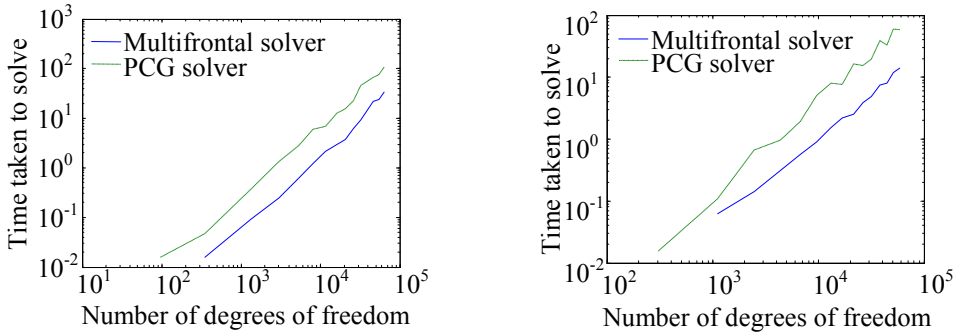
It is clear from Figure 4 that the frontal solvers are much more efficient solvers for this FE problem than for the band solver. The band solver requires a lot of memory to store the entire band of entries. Figure 3 shows that there are a number of zero entries inside the band which require storing and factorising. The frontal solvers only require the non-zero entries to be stored and therefore use much less memory. Although all three methods have efficient factorisation techniques, the band solver requires significantly more memory which slows the solution time considerably, making it less favourable.

Whilst the frontal solvers have considerably less solution time compared to the band solver, in both cases the use of multiple fronts in a general solver has been shown to be far more efficient compared to using a single front in a specifically designed solver. It is therefore clear that the multifrontal solver is the most appropriate direct solver for FG.

5.6. Direct vs. Iterative Linear Solver

Iterative solvers such as a pre-conditioned conjugate gradient (PCG) solver use significantly less memory to solve a linear system compared to direct methods (see [24] for more information on PCG methods). Thus, for large problems with

around 10^6 degrees of freedom, the storage saving can produce a more efficient solver. Therefore, PCG solvers are widely researched and used with FE problems, (see for example [25], [11], [26]).
Fixed grid was originally implemented using a PCG solver which exploits the symmetrical positive definite structure and uses a Jacobi pre-conditioner [11]. Thus the multifrontal linear solver is compared to this Jacobi PCG solver using a modest number of degrees of freedom.
Figure 5 clearly shows that the multifrontal solver is a significantly faster linear solver than the Jacobi PCG solver. The multifrontal solver is the faster solver in this instance compared to the PCG solver because a of the modest number of degrees of freedom considered. Potential solution time savings made by PCG solvers are not apparent at this level since the memory savings become more apparent with larger problems. However, it should be noted that, when complied using the Intel® Fortran complier, the multifrontal solver on an 3.0GHz Intel® Pentium 4 processor with 1GB RAM can solve a system with 10^6 degrees of freedom in under 200 seconds.



(a) Short cantilever beam (b) 'Michell' type structure

Figure 5: Solution time (seconds) for an increasing number of degrees of freedom

Aside from the saving in solution time compared to the PCG, direct solvers also have added an advantage that the convergence is always guaranteed. Thus the multifrontal solver is implemented in the FG finite element analysis.

6. Formulation for structural Optimisation
In this section, the formulation for the optimisation is presented, along with a sensitivity analysis and the optimisation algorithm.

6.1. Objective Function
The objective function is defined to minimise the compliance volume product. Letting the thickness of an element f be v_f and the compliance-volume product be $S(\mathbf{x})$ as represented in (3), the optimisation problem can be written as

$$\left.\begin{aligned} \min_x : S(\mathbf{x}) = CV = \mathbf{u}^T\mathbf{K}\mathbf{u}V = \sum_e x_e \mathbf{u}_e^T \mathbf{K}_0 \mathbf{u}_e \sum_f x_f v_f \\ : x_e \in (0,1], \mathrm{e} = 1,2,..., \mathrm{n}. \end{aligned}\right\}. \tag{3}$$

6.2. Sensitivity Analysis
The Lagrangian L of the optimisation problem (3) is written as

$$L = S(\mathbf{x}) + \sum_{e=1}^{n} \lambda_e^1 (\delta - x_e) + \sum_{e=1}^{n} \lambda_e^2 (x_e - 1), \tag{4}$$

for δ is a close to 0 as detailed above. The Lagrangian multipliers for the lower and upper bounds are λ_e^1 and λ_e^2. An optimal solution is achieved when the derivatives of the Lagrangian function are equal to zero with respect to the design variables, that is

$$\begin{aligned} \frac{\partial L}{\partial x_e} &= 0, \text{ for } e = 1,2,...,n. \\ &= \frac{\partial S}{\partial x_e} - \lambda_e^1 + \lambda_e^2. \end{aligned} \tag{5}$$

The derivative of the Lagrangian function, for elements with constant unit thickness $v_e = 1$, is,

$$\frac{\partial L}{\partial x_e} = \mathbf{u}^T \mathbf{K} \mathbf{u} - V \mathbf{u}_e^T \mathbf{K}_0 \mathbf{u}_e - \lambda_e^1 + \lambda_e^2. \tag{6}$$

Thus, for B elements, where the side constraints are inactive, the optimality condition is:

$$S_e = \frac{V \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e}{\mathbf{u}^T \mathbf{K} \mathbf{u}} = 1. \tag{7}$$

For I elements, where the upper side constraint λ_e^2 is active and hence $\lambda_e^2 > 0$, the optimality condition is:

$$S_e = \frac{V \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e}{\mathbf{u}^T \mathbf{K} \mathbf{u}} > 1. \tag{8}$$

Finally, for O elements, where the lower side constraint λ_e^1 is active, $\lambda_e^1 > 0$, the optimality condition is:

$$S_e = \frac{V \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e}{\mathbf{u}^T \mathbf{K} \mathbf{u}} < 1. \tag{9}$$

6.3. Optimisation Algorithm

The boundary based FG optimisation algorithm presented here removes material by slowly moving the boundaries of a structure from a rectangular design domain, based on the optimality conditions (7)-(9). The slow movement of the boundaries is governed by applying move limits. The optimisation algorithm for the existing boundaries is outlined as follows:

1. The user defines 4 point locations which define the initial rectangular design domain. The user also defines each set of kinematic loading and boundary conditions.
2. A B-spline [27] is defined between any two sets of kinematic conditions using the point locations defined in step 1. If the user wishes to preserve the symmetry of the structure, then extra B-spline can be defined to do so.
3. Control points **c** for the B-spline generation are created at equal distances along each B-spline. The distance between each control point is user defined.
4. FG generation.
5. The stiffness matrix **K** is generated to solve for element sensitivities S_e at the centre of each element.
6. Each B-spline is moved in turn by moving the control points. The movement of the control points is defined in terms of a direction and the magnitude with which it moves and is detailed below. The algorithm for the movement of the B-splines is limited to the removal of material from the structure, however, it can be generalised to both remove and add material.
7. The direction of the control point movement is determined by the position of the B-spline close to the control point and that positions closest point to $S_e = d_i$. d_i, for iteration i, is a small proportion of the optimal sensitivity $S_e = 1$ and therefore acts as a control parameter which limits the direction of movement
8. Move limits $\mathbf{m}_i$ are applied to the magnitude of the control point movement where $\mathbf{m}_i = [m_{min}, m_{max}]$ for iteration i defined in (10), and $\boldsymbol{\eta} = [\eta_{min}, \eta_{max}]$ which is user defined.

$$\mathbf{m}_i = \sqrt{1 - S_e} \cdot \boldsymbol{\eta}. \tag{10}$$

9. Determine the new distance between control points. If the distance between two consecutive control points c_{dist} satisfies (11) for the user defined parameter ξ_{min}, the control points are merged by removing both control points and inserting a new control point midway between the removed control points as shown in Figure 6. However, B-spline end control points must never be removed and in this circumstance, the two previous control points to the end control point are merged as previously described.

$$c_{dist} < \xi_{\min} \tag{11}$$

(a) Before two control points c_j and c_{j+1} are to be merged, (b) New B-spline with merged control point c_j

Figure 6: An example of merging 2 control points

10. If the distance between two control points satisfies (12) for user defined ξ_{max}, a new control point is inserted midway between the two control points as shown in Figure 7

$$c_{dist} < \xi_{\max} \tag{12}$$

(a) B-spline before a control point has been inserted (b) B-spline after a control point has been inserted

Figure 7: An example of inserting a control point

11. The new solution is examined for oscillating solutions or solution convergence
12. An oscillation is detected for small ε at iteration *i* if

$$\| \mathbf{c}_i - \mathbf{c}_{i-1} \| - \| \mathbf{c}_{i-1} - \mathbf{c}_{i-2} \| < \varepsilon \tag{13}$$

13. Convergence is detected for small ε at iteration *i* if

$$\| \mathbf{c}_i - \mathbf{c}_{i-1} \| < \varepsilon \tag{14}$$

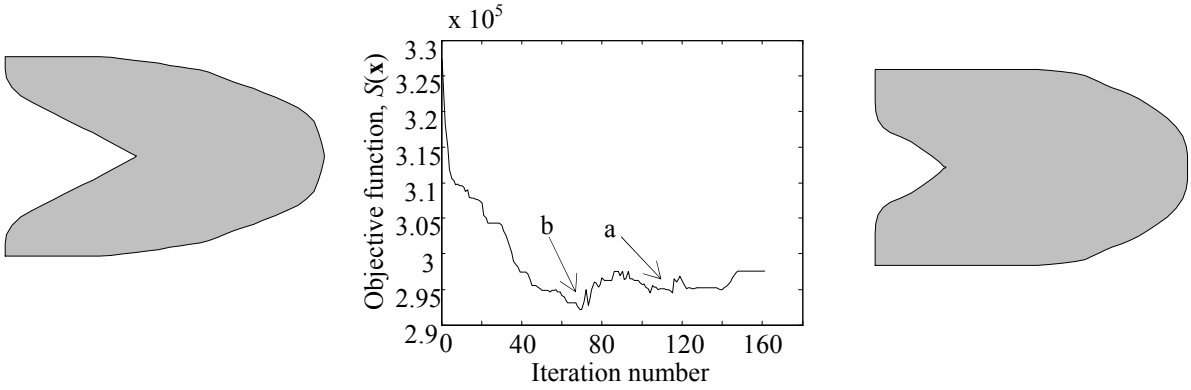
14. If neither (13) nor (14) are satisfied, optimisation returns to step 5, else
15. If $d_i < 1$, d_i is increased by a small amount to a maximum of $S_e = 1$and optimisation returns to step 6
16. If $d_i = 1$ or the stiffness matrix becomes ill-conditioned, optimisation is terminated.
17. The modification history is examined to select the design that is a local minimum of (3) and/or desirable volume ratio. The local and global minimum of the iteration history for a single run is henceforth referred to as the S-history local and S-history single-run-global minimum respectively.

7. Numerical Examples

In this section, the boundary based B-Spline FG optimisation algorithm described in section 6.3 is applied to the test problems of section 5. For both test problems the control point direction parameter is initially set as $d_i = 0.1$and increased by increments of 0.01.

7.1. Short Cantilever

The optimisation algorithm is applied to the short cantilever test problem of Figure 2(a) using a grid of size (48x30). The control point move limits $\mathbf{m}_i$ are governed by $\boldsymbol{\eta}_i = (1/3,1)$ and the distance between control points is limited by $\xi_{min} = 1$ and $\xi_{max} = 7.5$.



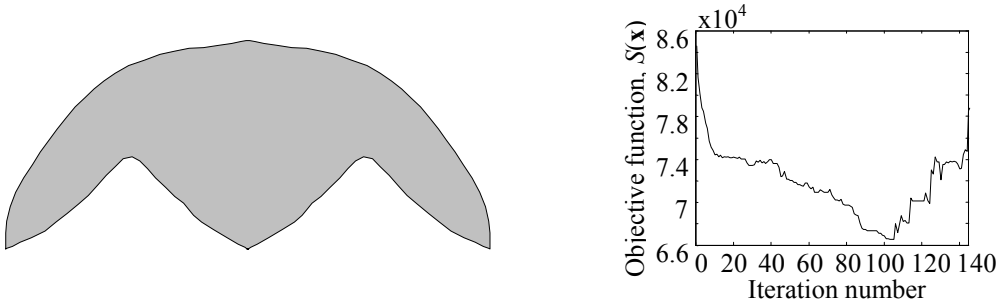
(a) S-history local minimum 'a' (b) Iteration history of $S(\mathbf{x})$ (c) S-history single-run global min.'b'

Figure 8: Shape optimization results for the short cantilever beam

Figure 8(a) presents the S-history local minimum obtained after 115 iterations with a 32% volume reduction. Figure 8(b) shows the iteration history of the objective function $S(\mathbf{x})$ where the S-history local minimum is achieved at 'a'. There are

a number of S-history local minima produced by the convergence of solutions which increases the control point direction parameter d_i.
The S-history single-run-global minimum occurs at iteration 70, has a volume reduction of 17%, is labelled 'b' on Figure 8(b) and is presented in Figure 8(c). The S-history single-run-global minimum has a similar shape to the local solution of Figure 8(a) in that the material has been removed from the corners on the right hand side and a triangular shape void has formed on the left hand side. The difference in $S(\mathbf{x}) < 1\%$ is small and mainly numerical. To further reduce $S(\mathbf{x})$, material needs to be removed from within the structure as there remains areas of low sensitivity S_e that the shape optimisation is unable to resolve. Thus the optimisation algorithm needs to incorporate changes in topology to reduce $S(\mathbf{x})$ further.

7.2. 'Michell' Type Structures
Due to symmetry, only half of the 'Michell' type structure of Figure 2(b) is optimised with a (25x25) grid. The move limits $\mathbf{m}_i$ are governed by $\boldsymbol{\eta}_i = (2/3,2.6)$ and the distance limits are $\xi_{min} = 3$ and $\xi_{max} = 17.5$. The optimisation algorithm minimises the compliance-volume product $S(\mathbf{x})$ until iteration 105, here the S-history single-run-global minimum is obtained with a volume reduction of 49% and is presented in Figure 9(a).



(a) S-history single-run-global minimum (b) Iteration history of the objective function $S(\mathbf{x})$

Figure 9: Shape optimisation results for a 'Michell' type structure

A plot of the objective function $S(\mathbf{x})$ against iteration is given in Figure 9(b) and shows there are again multiple optima which can relate to oscillations that occur before an increase in the direction parameter d_i. An increase in d_i can also cause an increase in $S(\mathbf{x})$ and generate multiple optima as material is being forcibly removed from a converged solution

8. Topology optimisation

8.1. Topology optimisation algorithm

To implement topology optimisation, the following algorithm is implemented after step 10 of the shape optimisation algorithm detailed in section 6.3:

1. Search for I elements with the minimum sensitivity S_e.
2. New holes are created at the minimum S_e satisfying $S_e < d_i$. Holes are created by inserting a closed B-spline with 4 points around the minimum element e. The 4 control points are placed in the centres of the 4 diagonal elements to ensure complete removal of the element.
3. The distances between the new B-spline and the existing B-splines are calculated. If two control points on different B-splines satisfy (15), as in Figure 10(a) for the user defined parameter ζ, the two boundaries are merged. Two boundaries are merged by removing the two closest control points and creating a new B-spline with the remaining control points as in Figure 10(b).

$$\| \mathbf{c}_j - \mathbf{c}_k \|_2 < \zeta . \tag{15}$$

(a) Two control points that satisfy (15) (b) 2 holes have merged into one

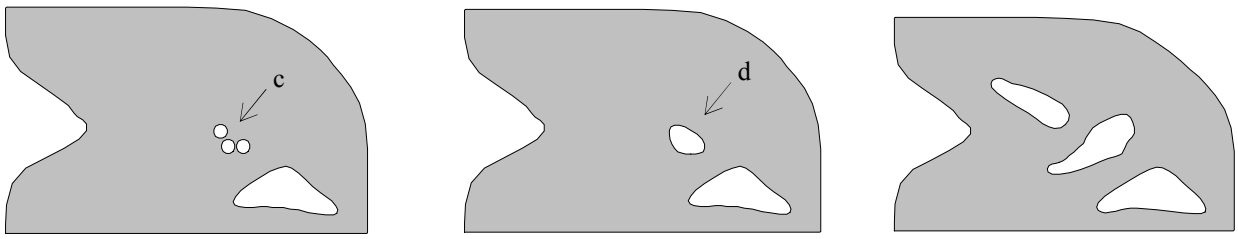
Figure 10: An example of merging two holes

4. Determine the new distances between control points for the merged boundaries
5. Continue with step 11 of shape optimisation algorithm.

8.2. Preliminary results

Boundary based topology optimisation using FG is applied to a simplified short cantilever whereby the load is applied to the bottom right hand corner of the structure. The direction of movement is defined by $d_i = 0.1$, which is increased by 0.01 on convergence. The move limits $\mathbf{m}_i$ are governed by $\boldsymbol{\eta} = (1/3,1)$, the distance between control points is determined by $\xi_{min} = 1$ and $\xi_{max} = 5.625$ and the distance between boundaries is limited by $\zeta = 1.6$.

Figure 11(a) shows iteration 38 before step 3 of the topology optimisation whereby close holes are merged. There is a cluster of 3 new holes introduced towards the centre of the structure at 'c'. These 3 holes are merged using (15) in the same iteration, 38, to give the topology in Figure 11(b) with the merged hole labelled 'd'. After 49 iterations, d_i has converged to 0.21 and several holes have been introduced as in Figure 11(c).



(a) A cluster of 3 holes are introduced (b) The cluster of 3 holes are merged (c) Topology at $d_i = 0.21$

Figure 11: Intermediate topologies for the topology optimization of a short cantilever

Figure 12 shows the iteration history of the objective function $S(\mathbf{x})$. $S(\mathbf{x})$ is minimised to iteration 49 (Figure 11(c)) and shows evidence of converging d_i.

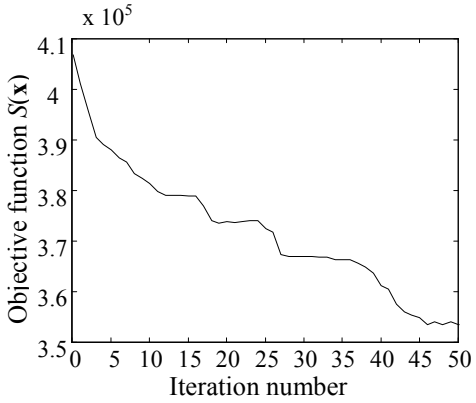


Figure 12: Iteration history of the objective function $S(\mathbf{x})$

9. Conclusions

Structural optimisation has been implemented using an efficient linear solver, within a Fixed Grid environment which uses B-splines to represent a structures boundary and has mathematically based optimality conditions.

The linear solver used was a direct solver known as a multifrontal solver. This multifrontal solver was chosen after a series of investigations showed that it was a fast direct solver for the class of problems created by the FG environment. The investigations also revealed that the multifrontal solver is faster than the iterative Jacobi preconditioned conjugate gradient for a modest number of degrees of freedom.

Using B-splines to represent the boundaries of a structure, smooth solutions were efficiently produced which were free from common numerical instabilities such as the formation of chequerboard patterns. Hence, the removal of these numerical instabilities has increased the reliability of the optimisation method.

In addition, the optimisation algorithm for existing boundaries was extended to create holes and merge holes, and a preliminary result for this effort was presented. The results suggest that the optimisation method produces solutions that agree well with known benchmark test problems and therefore the development of the topology optimisation implementation is continuing.

10. Acknowledgements
The authors thank Prof Iain Duff, Rutherford Appleton Laboratory for his useful advise and discussion on linear solvers and Dr Lars Krog, Airbus UK for his inspiring comments and suggestions. The authors would also like to thank Numerical Analysis group at the Rutherford Appleton Lab for their FORTRAN HSL packages. The authors acknowledge the support of UK Engineering and Physical Sciences Research Council (GR\ S68477) for this research.

11. References
1. Zhou M, Rozvany GIN. On the validity of ESO type methods in topology optimization. Struct Multidisc Optim, 2001, 21: 80-83
2. Edwards CS, Kim HA, Budd CJ. An evaluative study on ESO and SIMP for optimising a cantilever tie-beam. Struct Multidisc Optim, 2007, Online first
3. Bendsøe MP. Optimal shape design as a material distribution problem. Springer, Berlin, 1989
4. Rozvany GIN, Zhou M, Birker T. Generalized shape optimisation without homogenisation, Struct Optim, 1992, 4:250-254
5. Bendsøe MP, Sigmund O. Topology optimization: theory, methods and applications. Springer, Berlin, 2003
6. Stolpe M, Svanberg K. On the trajectories of penalization methods for topology optimization. Struct Multidisc Optim, 2001, 21:128-139
7. Kim H, Querin OM, Steven GP, Xie YM. A method for varying the number of cavities in an optimized topology using evolutionary structural optimization. Struct Multidisc Optim, 2000, 19:140-147
8. Jog CS, Haber RB. Stability of finite element models for distributed-parameter optimization and topology design. Comput Methods Appl Mech Eng, 1996, 130:203-226
9. Díaz A, Sigmund O. Checkerboard patterns in layout optimization. Struct Optim, 1995, 10:40-45
10. Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct Optim, 1998, 16:68-75
11. García MJ. Fixed grid finite element analysis in structural design and optimisation. PhD thesis, University of Sydney, Australia, 1999
12. Kim H, Garcia MJ, Querin OM, Steven GP. Introduction of fixed grid in evolutionary structural optimisation. Engineering Computations, 2000, 17(4):427-439
13. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. Int J Numer Meth Engrg, 1999,46:131-150
14. Sethian JA. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, Cambridge, UK, 1999
15. Wang M, Wang X, Guo D. A level set method for structural topology optimization. Comput Methods Appl Mech Engrg, 2003, 192:227-246
16. Allaire G, Jouve F, Toader AM. Structural Optimization using sensitivity analysis and a level-set method. J Comp Phys, 2004, 194:363-393
17. Sokolowski J, Żochowski A. On the topological derivative in shape optimization. SIAM J Control Optim, 1999, 37(4):1251-1272
18. Burger M, Hackl B, Ring W. Incorporating derivatives into level set methods. J Comp Phys, 2004, 194:344-362
19. Allaire G, de Gournay F, Jouve F, Toader AM. Structural optimization using topological and shape sensitivity via a level set method. Control and Cybernetics, 2005, 34(1):59-80
20. Norato JA, Bendsøe MP, Haber RB, Tortorelli DA. A topological derivative method for topology optimization. Struct Multidisc Optim, 2007, 33:375-386
21. Michell AGM. The limits of economy of material in frame-structures. Philos Mag, 1904, 8:589-597
22. Duff IS, Erisman AM, Reid JK. Direct methods for sparse matrices. Oxford University Press, Oxford, UK, 1986
23. Rutherford Appleton Laboratory. Harwell subroutine library. Numerical Analysis Group, Chilton, Oxfordshire, UK, 2004
24. Golub GH, Van Loan CF. Matrix computations 3rd ed. Johns Hopkins University Press, Baltimore, USA,1996
25. Dickinson JK, Forsyth PA. Preconditioned conjugate gradient methods for three dimensional linear elasticity. Int J Numer Meth Engrg, 1994, 37(17):2211-2234
26. Trevelyan J, Scales DJ, Morris R, Bird GE. Acceleration of boundary element computations in reanalysis of problems in elasticity. WCCM VI in conjunction with APCOM'04, Beijing, China, 2004
27. Komzsik L. Approximation techniques for engineers. CRC Press, Taylor and Francis Group, Boca Raton, 2007