

Design space optimization using design space adjustment and refinement

In Gwun Jang · Byung Man Kwak

Received: 22 October 2006 / Revised: 3 January 2007 / Published online: 20 March 2007
© Springer-Verlag 2007

Abstract To deal with large-scale problems that often occur in industry, the authors propose design space optimization with design space adjustment and refinement. In topology optimization, a design space is specified by the number of design variables, and their layout or configuration. The proposed procedure has two efficient algorithms for adjusting and refining design space. First, the design space can be adjusted in terms of design space expansion and reduction. This capability is evolutionary because the design domain expands or reduces wherever necessary. Second, the design space can be refined uniformly or selectively wherever and whenever necessary, ensuring a target resolution with fewer elements, especially for selective refinement. Accordingly, the proposed procedure can handle large-scale problems by solving a sequence of smaller problems. Two examples show the efficiency of the proposed approach.

Keywords Design space optimization · Topology optimization · Design space adjustment · Design space refinement · Large-scale optimization

1 Introduction

Topology optimization has greater impact on the downstream of a design process than size or shape optimization

because topology should be determined before the size or shape. The literature features several popular methods of topology optimization, namely, the homogenization method (Bendsoe and Kikuchi 1988; Suzuki and Kikuchi 1991; Diaz and Kikuchi 1992; Hassani and Hinton 1998a–c), the solid isotropic material with penalization (SIMP) method (Bendsoe 1989; Zhou and Rozvany 1991; Yang and Chuang 1994; Bendsoe and Sigmund 1999), and the evolutionary structural optimization (ESO) method (Xie and Steven 1993; Querin and Steven 1998; Kim et al. 2000). Recently, the level set method (Sethian and Wiegmann 2000; Allaire et al. 2004; Wang et al. 2003) that was proposed by Osher and Sethian (1988) has been widely studied in the field of topology optimization as well as fluid mechanics and image processing because this method can handle complex topological changes naturally with a level set function.

A major drawback of these methods is the computational time, especially for the large-scale practical problems that often occur in the automobile industry (Wang et al. 2004) and the aircraft industry (Krog et al. 2004). Parallel processing (Borrvall and Petersson 2001; Kim et al. 2004) might help but that is not in the realm of topology optimization per se and does not resolve the inherent problem of inefficiency. The concept of the design space optimization, which was proposed by Kim and Kwak (2002), introduced an evolutionary method. By starting with a small design space and advancing to a larger space with a large number of design pixels, the method eventually achieves an optimal design space. This original method, however, was not implemented to its full potential in terms of efficiency because only a single layer expansion was possible. While endeavoring to put the evolutionary method into practice, Jang and Kwak (2006) derived the concept of multi-layer adjustment. This method is especially efficient for open design domain problems.

I. G. Jang (✉) · B. M. Kwak
Department of Mechanical Engineering,
Korea Advanced Institute of Science and Technology,
373-1 Guseong-dong,
Yuseong-gu, Daejeon 305-701, Republic of Korea
e-mail: jangin@khp.kaist.ac.kr

B. M. Kwak
e-mail: bmkwak@khp.kaist.ac.kr

Mesh refinement in topology optimization was first used by Maute and Ramm (1995). They used an adaptive mesh refinement strategy of the finite element method (FEM) to change the design patch for topology optimization. Costa and Alves (2003) also proposed a layout optimization method that relied on h -adaptive methods. They combined topology optimization and an adaptive refinement method to improve the definition of domain boundaries and to restrict the strain errors. However, these methods refine all the elements that exist inside a structure, eventually producing a uniform refinement of the entire structure. For efficiency, we need a method of selectively refining elements according to some type of refinement priority, although the elements are inside a structure.

By using the design space adjustment and refinement, we implemented the concept of the design space optimization (Kim and Kwak 2002; Jang and Kwak 2006) in such a way as to develop an efficient methodology for large-scale problems. Our procedure can be easily interfaced with a conventional SIMP method. Both algorithms are based on fixed grid (FG).

2 Design space optimization

2.1 Design space adjustment: from small to large

In many engineering problems, the selection of an initial design domain is not easy for a given problem unless the domain is fully constrained or predefined. To handle an open domain problem in conventional optimization, we need to ensure that the initial domain is large enough to include the anticipated result. Thus, aside from the difficulty of selecting an appropriate working domain, there is a considerable computational cost due to the abundance of unnecessary but unavoidable elements. Design space optimization works well by allowing for domain expansion where necessary, regardless of the shape or size of the initial design domain (Jang and Kwak 2006). Accordingly, the design space evolves from a small space to a large space and can even start from a simple skeletal design space. Thus, this evolutionary aspect makes the proposed method efficient and suitable, especially for large-scale problems in three dimensions. We now describe the algorithm with reference to our previous work (Kim and Kwak 2002; Jang and Kwak 2006).

2.1.1 Design variable sensitivity and design space sensitivity

For structural topology optimization, we considered a minimum compliance problem that was subject to a volume usage constraint. The design variables were the element

densities, ρ . The number of design variables, N , was the same as the number of elements used in the FE model. Using this information, we formulated the optimization problem on the basis of the FEM as follows:

$$\begin{aligned} \text{Minimize } f(\boldsymbol{\rho}) &= \mathbf{U}(\boldsymbol{\rho})^T \mathbf{F} = \mathbf{U}(\boldsymbol{\rho})^T \mathbf{K}(\boldsymbol{\rho}) \mathbf{U}(\boldsymbol{\rho}) \\ \text{Subject to } g(\boldsymbol{\rho}) &= \sum_{i=1}^N \int_{\Omega_i} \rho_i d\Omega \leq V_0 \\ 0 &< \rho_{\min} \leq \rho_i \leq 1 \\ \text{where } \boldsymbol{\rho} &= [\rho_1 \ \rho_2 \ \cdots \ \rho_N]^T \end{aligned} \quad (1)$$

where $\mathbf{U}(\boldsymbol{\rho})$ is the nodal displacement vector, \mathbf{F} is the nodal external force vector, and Ω_i is the domain of the i th design variable. The stiffness matrix, $\mathbf{K}(\boldsymbol{\rho})$, in (1) can be expressed as

$$\mathbf{K}(\boldsymbol{\rho}) = \sum_{i=1}^N \mathbf{k}_i(\rho_i) = \sum_{i=1}^N \int_{\Omega_i} \mathbf{B}_i^T \mathbf{D}(\rho_i) \mathbf{B}_i d\Omega, \quad (2)$$

where \mathbf{k}_i is the local stiffness matrix and \mathbf{B}_i is the strain interpolation matrix. We then expressed the elasticity matrix, $\mathbf{D}(\rho_i)$, for a 3D isotropic case as follows:

$$\mathbf{D}(\rho_i) = \frac{E(\rho_i)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (3)$$

where ν is Poisson's ratio. In the conventional SIMP method, Young's modulus, $E(\rho_i)$, was penalized as

$$E(\rho_i) = \rho_i^n E_0, \quad (4)$$

where n is a penalty exponent and E_0 is a reference property of a given isotropic material.

Using (2), (3), and (4), we easily obtained the design variable sensitivity (DVS) equation as (Haug et al. 1986),

$$\psi' = \frac{df}{d\rho_i} = -\mathbf{U}_i^T \frac{\partial \mathbf{k}_i}{\partial \rho_i} \mathbf{U}_i. \quad (5)$$

In contrast to the DVS, the design space sensitivity (DSS) refers to the effect of the new design variable addition on the objective function or constraints. Because changes in the design space are mathematically a discontinuous process, Kim and Kwak (2002) calculated the DSS by using a pivot phase and a directional derivative. Figure 1 illustrates the pivot phase of design space optimization. To calculate the DSS, we put a layer of new elements with a very low density near zero. The total number of elements is

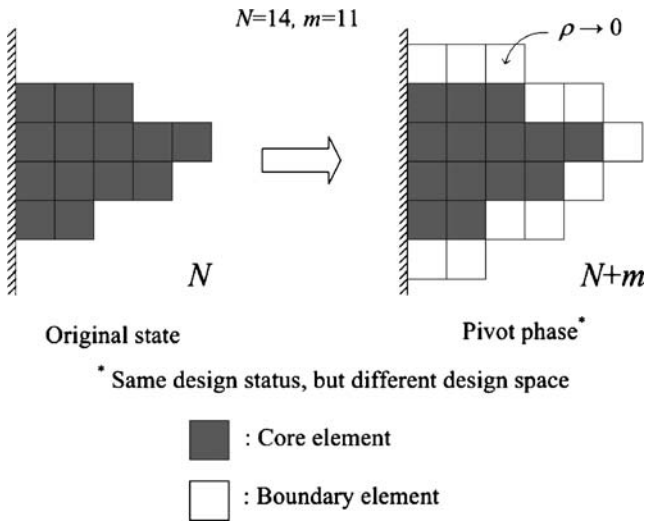


Fig. 1 The concept of pivot phase

changed from N to $N + m$ (where m is the total number of newly created elements), but this addition makes no change to the structural status, such as compliance or weight. The resulting space is the pivot phase. For convenience, we named the original existing elements before the pivot phase as “core elements” and the elements that were newly created for the pivot phase as “boundary elements.” With this pivot phase, we could then obtain the DSS or the directional derivative by taking a derivative with respect to ρ_i and then taking the limits as $\rho_i \rightarrow 0+$.

For the case of compliance, we derived the DSS as

$$\psi'_{+, \rho_i \rightarrow 0+} = \left(\frac{df}{d\rho_i} \right)_{+, \rho_i \rightarrow 0+} = -\mathbf{U}_i^T \left(\frac{\partial \mathbf{k}_i}{\partial \rho_i} \right)_{+, \rho_i \rightarrow 0+} \mathbf{U}_i \quad (6)$$

In (6), the stiffness is solely the function of Young’s modulus, E , which is considered a function of density, ρ , as in (4). Thus,

$$\psi'_{+, \rho_i \rightarrow 0+} = \left(\frac{df}{d\rho_i} \right)_{+, \rho_i \rightarrow 0+} = -\mathbf{U}_i^T \left(\frac{\partial \mathbf{k}_i}{\partial E} \right) \left(\frac{dE}{d\rho_i} \right)_{+, \rho_i \rightarrow 0+} \mathbf{U}_i \quad (7)$$

From the definition of a pivot phase and (7), we were able to compute the DSS when the function $E(\rho_i)$ had the following properties:

$$E(\rho_i) = 0 \quad \text{as } \rho_i \rightarrow 0+ \quad (8)$$

$$\frac{dE(\rho_i)}{d\rho_i} \neq 0 \quad \text{as } \rho_i \rightarrow 0+.$$

Note that the typical function used for the DVS, namely $E(\rho_i) = \rho_i^3 E_0$, is unsuitable for the DSS because it had a zero slope at $\rho_i = 0$. One of the material models to satisfy these properties is the rational approximation for material

properties (RAMP), which was proposed by Stolpe and Svanberg (2001). This has the alternative interpolation for penalization as follows:

$$E(\rho_i) = E_{\min} + \frac{\rho_i}{1 + q(1 - \rho_i)} (E_0 - E_{\min}). \quad (9)$$

With $E_{\min} = 0$ in (9), we can get the simplified form $E(\rho_i) = \frac{\rho_i}{1 + q(1 - \rho_i)} E_0$. After simple arithmetic, we can check that this model satisfies (8) as follows:

$$E(\rho_i) = 0 \quad \text{as } \rho_i \rightarrow 0+ \quad (10)$$

$$\frac{dE(\rho_i)}{d\rho_i} = \frac{1}{1 + q} E_0 \neq 0 \quad \text{as } \rho_i \rightarrow 0+.$$

In this paper, we used $q=3$. For the theoretical aspects and mathematical proofs, refer to the work of Stolpe and Svanberg (2001).

2.1.2 Design space expansion based on the DSS

To accelerate the expansion of a design space, we proposed the following expansion strategy in which multiple layers are added in relation to the magnitude of the DSS:

- Step 1: Calculate DSS for the boundary elements by decoupling boundary elements.
- Step 2: Calculate the DVS for the core elements and select the maximum DVS as the absolute value.
- Step 3: Use (11) to calculate the number of expansion layers, r_i , for the i th boundary element, and then expand the design space around the i th boundary element with r_i . That is,

$$\frac{|(\psi'_i)_{\text{boundary}}|}{\max |(\psi'_i)_{\text{core}}|} \times \left(\frac{V_{\max}}{v_i} \right) = \zeta \times (2r_i^2 + 2r_i + 1) \quad \text{in 2D case}$$

$$= \zeta \times \left(\frac{4}{3} r_i^3 + 2r_i^2 + \frac{8}{3} r_i + 1 \right) \quad \text{in 3D case.} \quad (11)$$

where $(\psi'_i)_{\text{boundary}}$ is the DSS of the i th boundary element, $\max |(\psi'_i)_{\text{core}}|$ is the maximum absolute DVS among the core elements, V_{\max} is the maximum volume allowed by the volume constraint, v_i is the volume of the i th element, and ζ is a scaling constant. For the scaling constant, we used a value of 100 for two dimensions and 50 for three dimensions when we used the sensitivity filtering of Sigmund and Petersson (1998) to eliminate the checkerboard pattern and mesh dependency.

The value of ζ has an impact on the magnitude of the expansion layer, r_i , in (11). We suggested the value of ζ for 2D and 3D conditions by way of numerical experiments to expedite the design space expansion in a suitable and efficient manner. In (11), the right-hand term $2r_i^2 + 2r_i + 1$ refers to the number of elements enclosed in the diamond area of Fig. 2 and $\frac{4}{3}r_i^3 + 2r_i^2 + \frac{8}{3}r_i + 1$ refers to the number of elements enclosed in the octahedron in which the thickness of the expansion layer is r_i .

The heuristic formula in (11) effectively calculates the area or volume in terms of the number of element layers that are influenced by the strength of the contribution of a boundary element in comparison with the strength of the contribution of the core element that has the maximum sensitivity.

In this research, we used FG, or often called the background mesh, for an easy and efficient mesh generation. It was generated by superimposing a rectangular grid of equally sized elements on the given structure instead of generating a mesh to fit the structure. A predetermined systematic order was applied to the nodes and elements in FG. So, if we know the location of an element, then we can identify the corresponding element number and node numbers that belong to that element. We set a FG domain as large as possible to cover any anticipated results because the optimal design space is not generally known. Even if we set a FG domain quite large, it does not affect the FE calculation time at all. The FG is imaginary for FE modeling and is used to transfer information for selected nodes and elements to a FE model. It is especially efficient to adjust design space as well as to refine it in which the number and locations of design variables are to be changed.

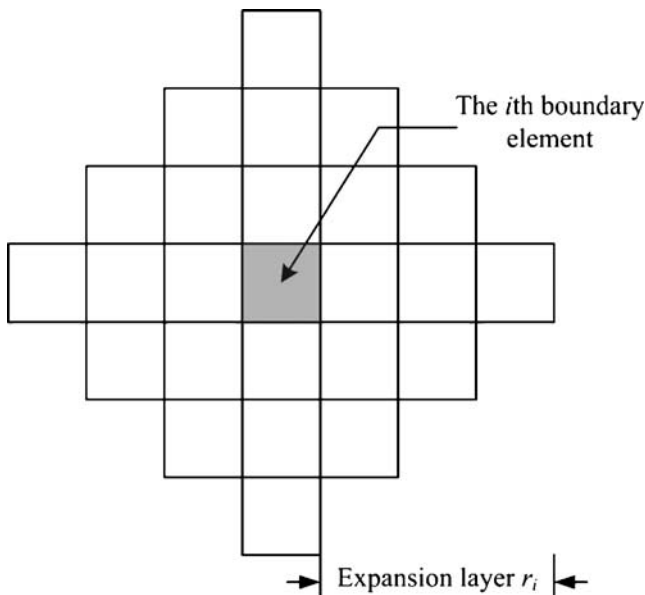


Fig. 2 Design space expansion around the i th boundary element with the expansion layer r_i in 2D

2.2 Design space refinement: from coarse to fine

2.2.1 Procedure of design space refinement

The most time-consuming part of a design space optimization is searching for an optimal design space. The design space in this paper is defined as

$$S \equiv \{N, L_N, \{b_1, b_2, \dots, b_N\}\}, \tag{12}$$

where L_N denotes a set of layouts or configurations for N design variables $\{b_1, b_2, \dots, b_N\}$. In (12), L_N is strongly related to N . That is, as N becomes larger, L_N becomes larger. Basically, as shown in Fig. 3, we can represent the shape of a structure more clearly when L_N is relatively large. If we start a topology optimization with the finest mesh resolution, the subsequent large number of design variables, N , and layouts, L_N , in the design space requires too much calculation time.

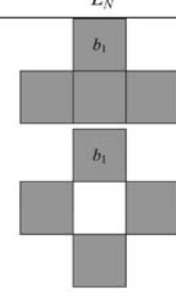
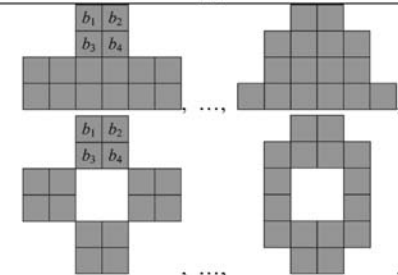
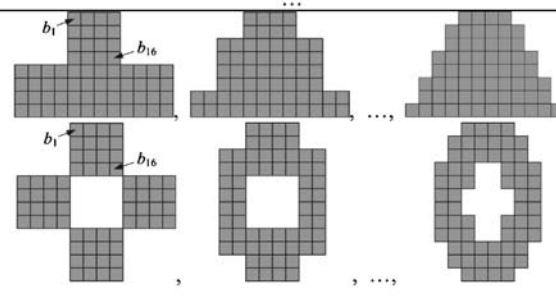
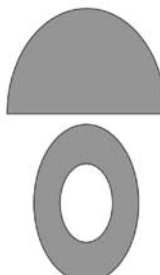
A better way is to start with a coarse mesh and to adopt a sequence of design space refinements. Before starting such a procedure, we set the number of refinements that will be done during the procedure. At the beginning, we can find a rather rough optimal design space by using a low refinement level. A converged design space actually denotes the design space of the optimum topology. This means there is no need to expand or reduce the design domain, that is, an expansion would not bring in any new structural material. When we find an optimal design space at the refinement level, we can refine the FE model and thereby improve the design space for the next optimization process. This process continues until we get the desired optimal design space at the target refinement level. For this purpose, we used uniform refinement and selective refinement as shown in Fig. 4.

The selective design space refinement proposed was aimed to approximate a structure’s global property (compliance in this paper) within the prescribed accuracy bound with fewer elements for computational efficiency, as well as for the clearness of the domain boundary, compared to the uniform refinement. First, we refined the elements on the border of the structure for a clearer image. This was easily done by checking the densities of elements in the FG. Due to a sensitivity filtering, the elements on the border of the structure usually have intermediate densities. The set of elements to be refined for a clearer image, S_1 , is given by

$$S_1 \equiv \{i \mid \rho_{\min} \leq \rho_i \leq 0.7, i = 1, \dots, N\}. \tag{13}$$

Next, to check the refinement effect of each element on the objective function, we first *uniformly* refined the whole FE model as shown in Fig. 5 and then obtained the DVS of all refined elements. Note that due to a refinement, the number of elements and design variables change. Uniformly refined elements that descend from the i th element are called

Fig. 3 Design space change through a sequence of refinements: refinement level 1, refinement level 2, and refinement level 3

	N	L_N	b_i
Refinement level 1	4		$\{b_1, \dots, b_4\}$
Refinement level 2	16		$\{b_1, b_2, b_3, b_4, \dots, b_{16}\}$
Refinement level 3	64		$\{b_1, b_2, \dots, b_{16}, \dots, b_{64}\}$
Real structure	∞		

sub-elements of the i th element. The FE model can be refined adaptively on the basis of the obtained DVS information. The underlying idea taken is that if there is some large difference in the sensitivity values among sub-elements, then this element is better refined further from the empirical point of view. At the moment, its mathematical rigor is left as a future work. To implement the idea, we calculated the standard deviation of the DVS for sub-elements and ranked in descending order. Denoting the standard deviation corresponding to the i th element by σ_i ,

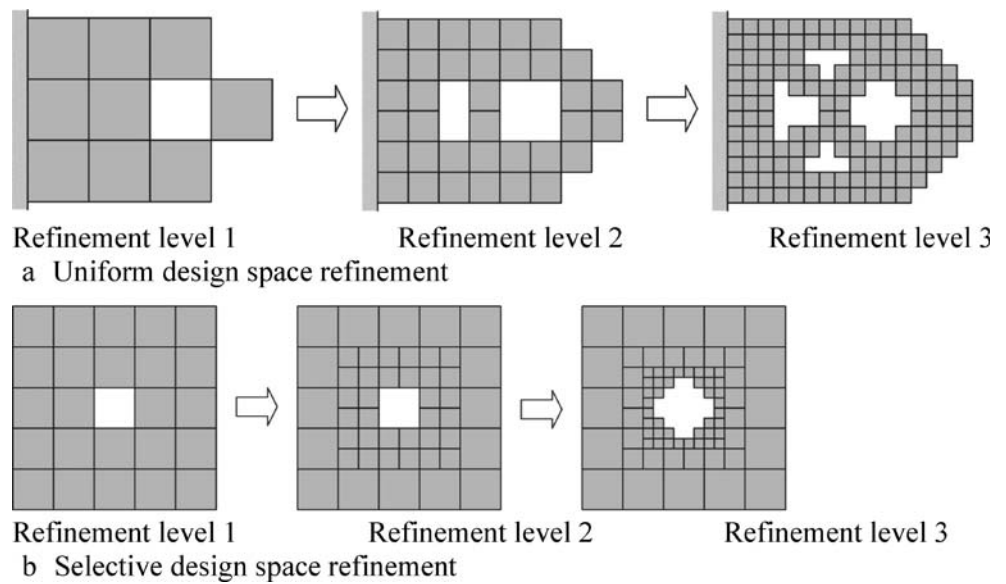
$$\sigma_i = \sum_{k=1}^n \frac{(\psi'_{i|k} - \overline{\psi'_{i|k}})^2}{n}, \tag{14}$$

where $\psi'_{i|k}$ is the DVS for the k th sub-element of the i th element, $\overline{\psi'_{i|k}}$ is the mean value of the DVS for the sub-elements of the i th element, and n is the total number of sub-elements for the i th element.

Instead of using a fixed threshold value, the extent of a refinement is better controlled by the percentage of elements that may be taken differently for various structures. That is, all the elements whose rank fraction is less than a number α ($0 \leq \alpha \leq 1$) are to be refined. The set of elements to be refined for a compliance approximation, S_2 , is given by

$$S_2 \equiv \{i \mid \frac{Rank[\sigma_i]}{N} \leq \alpha, i = 1, \dots, N\} \tag{15}$$

Fig. 4 The concept of uniform and selective design space refinement



where Rank $[a_i]$ denotes the rank of a_i in descending order in the set $\{a_i\}$; and a is the refinement constant that designates the extent of the refinement. This condition means that the top a fractions of N elements are to be refined. An appropriate a value in (15) can be taken in relation to the uniform refinement with a prescribed accuracy measure, β , by the following bisectioning algorithm:

- Step 1: Obtain the objective function value, ψ_0 , at the prerenement state, that is, when $\alpha=0.0$ in (15).
- Step 2: Obtain the objective function value, ψ_{target} , at the state of the uniform refinement as shown in Fig. 5 and get all the DVS values of the sub-elements of all elements to obtain the value of σ_i in (15).
- Step 3: In accordance with (13) and (15), refine the corresponding elements to their next refinement level and then obtain the objective function value, ψ_α , at the state of the selective refinement with the following refinement constant:

$$\alpha = \frac{\alpha_l + \alpha_u}{2}, \tag{16}$$

where α_l is the lower limit of refinement constant (0.0 initially) and α_u is the upper limit (1.0 initially).

Step 4: Check the following convergence requirement:

$$\left| \frac{\psi_{target} - \psi_\alpha}{\psi_{target}} \right| \leq \beta. \tag{17}$$

If satisfied, stop the routine. Otherwise, $\alpha_1 = \alpha$. Then, return to step 3.

As mentioned earlier in this paper, the compliance is taken as an indicator of refinement. The ratio in (17) indicates how near the compliance of ψ_α to ψ_{target} of the uniform refinement. The constant β denotes a preset level for the nearness. While (14) gives an ordering of elements for selection based on the gradient that was effectively calculated by the finite difference of the sensitivities of the compliance, it is found more practical to set β instead of the percentage number of elements, α . In the following examples, β is taken as 0.01 and shown working well.

So, the final set of elements for the selective refinement, S , becomes

$$S = S_1 \cup S_2. \tag{18}$$

In the proposed strategy, we estimated the number of elements after a selective refinement, N_{est} , as follows:

$$N_{est} = N + N_{ref} \times (2^n - 1), \tag{19}$$

where N is the number of elements before a refinement, N_{ref} is the number of elements in the set S in (18), and n is the dimension of the problem (2 in 2D and 3 in 3D).

2.2.2 Nodal constraint equation for the transition elements based on FG

When we use selective refinement in the FG as explained above, then, as shown in Fig. 6, we inevitably get transition elements. A simple and convenient way of preserving the

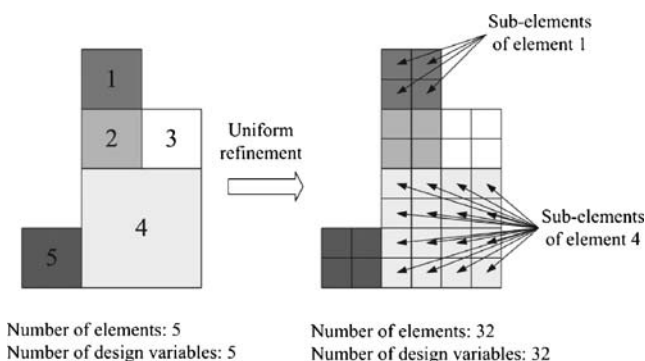
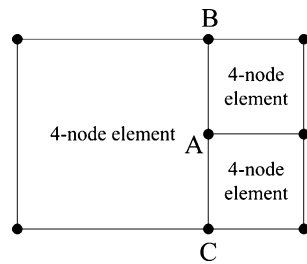


Fig. 5 Refining all the elements uniformly to their sub-elements to check the effect of refinement

Fig. 6 A four-node to four-node element transition case in 2D FEM



compatibility and continuity of transition elements is to use the following nodal constraint equation:

$$u_A = \left(\frac{u_B + u_C}{2} \right), \tag{20}$$

where u_A , u_B , and u_C are the displacements of node A , B , and C , respectively.

Figure 7 shows the 3D transition elements, which we considered for the sake of generalization. For compatibility and continuity, we should express the displacement of transition node 5 as a combination of the displacement of nodes 1–4. During the optimization with the selective refinement, we do not know how many and where such transition nodes happen before the refinement procedure. So, we proposed a systematic scheme using the shape function of FEM to generate all the nodal constraint equations only with the FG information.

Because linear elements are used for the finite element analysis, the following 2D shape function is necessary for the interfacing plane in the 3D:

$$u_i = \frac{1}{4}(1-r)(1-s)u_1 + \frac{1}{4}(1+r)(1-s)u_2 + \frac{1}{4}(1+r)(1+s)u_3 + \frac{1}{4}(1-r)(1+s)u_4, \tag{21}$$

where u_i is the displacement of any transition node i ; u_1 , u_2 , u_3 , and u_4 are the displacements of nodes 1, 2, 3, and 4, respectively; and r and s are the local coordinates that vary from -1 to 1 . Transforming to range $[0, 1]$, we had the following modified shape functions:

$$u_i = (1-r')(1-s')u_1 + r'(1-s')u_2 + r's'u_3 + (1-r')s'u_4, \tag{22}$$

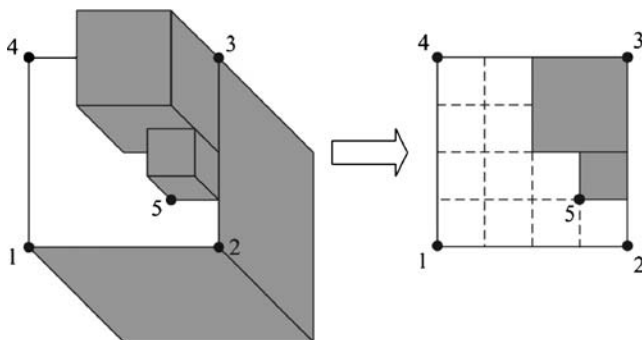


Fig. 7 An example of general element transition case in 3D

where r' and s' are local coordinates that vary from 0 to 1. To adopt the concept of selective design space refinement, we introduced new local coordinates, r^* and s^* , for the purpose of measuring r' and s' in the units of the FG. That is,

$$\alpha^2 \times u_i = (\alpha - r^*)(\alpha - s^*)u_1 + r^*(\alpha - s^*)u_2 + r^*s^*u_3 + (\alpha - r^*)s^*u_4, \tag{23}$$

where $\alpha = 2^m$; m is the number of refinements to be made to the target state in the current element, which includes nodes 1–4 and r^* and s^* are the relative orders of position of node i as a natural number in local coordinates r and s , respectively. For node 5 in Fig. 7, m is 2, r^* is 3, and s^* is 1. By using this information and (23), we can get the following nodal constraint equation:

$$16u_5 = 3u_1 + 9u_2 + 3u_3 + u_4. \tag{24}$$

As shown in (23), the geometric information in the FG is all we need to make a nodal constraint equation. This makes it simple and convenient to apply the selective design space refinement to any kind of structure.

Unfortunately, the imposition of the nodal displacement constraints in the selective refinement may cause a significant overhead in computation cost. For example, given the set of L linear simultaneous equations in the unknown u_j ,

$$\sum_{j=1}^L K_{jk}u_j = F_k \quad (1 \leq k \leq L), \tag{25}$$

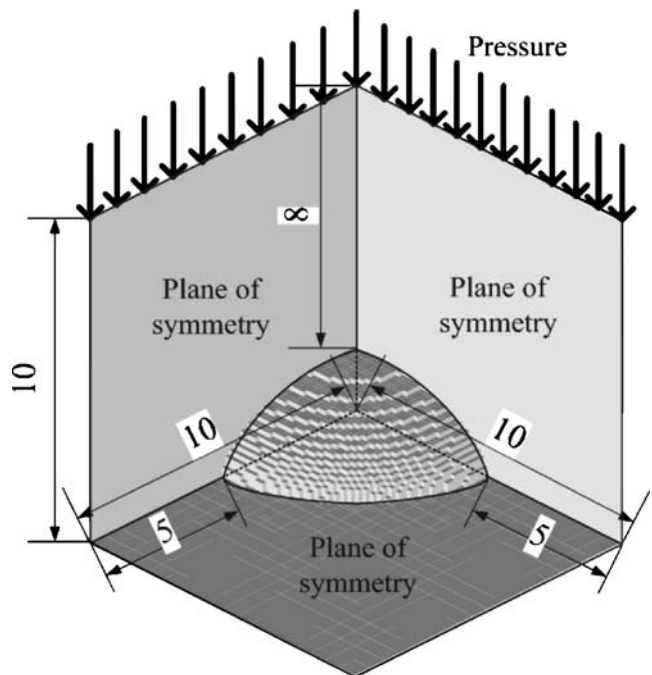


Fig. 8 A quarter model with an elliptic hole in 3D

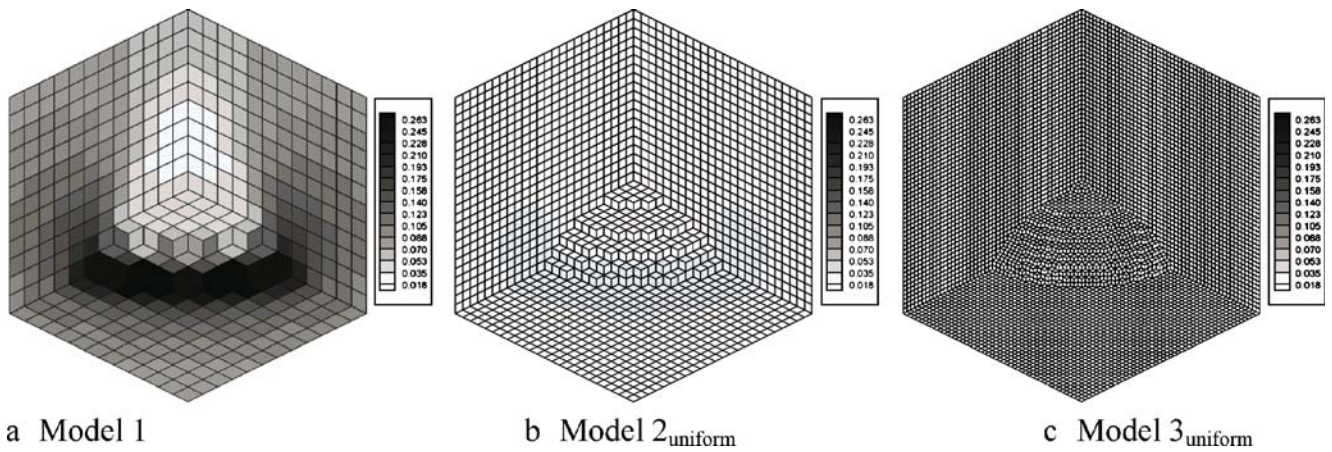


Fig. 9 Sensitivity level plot for a quarter model with an elliptic hole using the uniform design space refinement

is subject to a nodal constraint equation

$$\sum_{j=1}^L C_j \mathbf{u}_j = C_0 \tag{26}$$

We normalized (26) with respect to the prime DOF, \mathbf{u}_i , by dividing \mathbf{u}_i by C_i . From the normalization, we get

$$\sum_{j=1}^L C_j^* \mathbf{u}_j = C_0^* \tag{27}$$

where $C_j^* = C_j/C_i$ and $C_0^* = C_0/C_i$. After some manipulation (ANSYS verification manual, ANSYS release 8.0 documentation) to consider the linear constraint, we can get the following modified equations:

$$\sum_{j=1}^{L-1} \mathbf{K}_{kj}^* \mathbf{u}_j = \mathbf{F}_k^* \quad (1 \leq k \leq L-1), \tag{28}$$

where $\mathbf{K}_{kj}^* = \mathbf{K}_{kj} - C_j^* \mathbf{K}_{ki} - C_k^* \mathbf{K}_{ij} + C_k^* C_j^* \mathbf{K}_{ii}$ and $\mathbf{F}_k^* = \mathbf{F}_k - C_0^* \mathbf{K}_{ki} - C_k^* \mathbf{F}_i + C_k^* C_0^* \mathbf{K}_{ii}$. As the number of constraint equations increases, the modified stiffness matrix, \mathbf{K}_{kj}^* , and

the load vector, \mathbf{F}_k^* , in (28) become more complex, eventually causing a calculational overhead. However, the reduced number of elements in the selective refinement makes the calculation faster. The net effect of these opposite aspects cannot be determined analytically; hence, in the following section, we check this effect with the aid of numerical examples. For detailed information, we used the preconditioned conjugate gradient (PCG) method in ANSYS as an equation solver. This requires less disk file space and is faster than other methods for large scale models. It is also known that it can robustly solve equations with constraint equations. These properties fit well with the situation in this research. Also, all element stiffness matrices according to the FG were rebuilt in every FE analysis.

2.2.3 Numerical aspects of design space refinement

The refinement procedure is basically the same as improving finite element solution in terms of compliance. The compliance is equivalent to twice of the strain energy. In

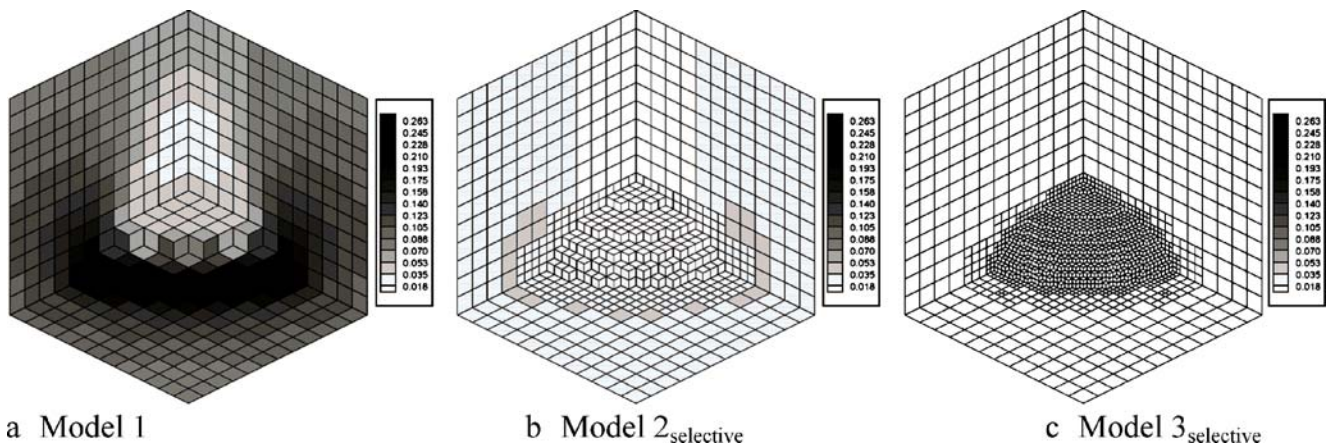


Fig. 10 Sensitivity level plot for a quarter model with an elliptic hole using the selective design space refinement

finite element analysis, as a sequence of refinements is made, the strain energy is supposed to converge to the exact one, monotonically increasing from below (Hughes 1987) and so must be the compliance as follows:

$$\lim_{i \rightarrow \infty} \psi^{i, \text{uniform}} = \psi, \tag{29}$$

where $\psi^{i, \text{uniform}}$ is the compliance after i times uniform refinements and ψ is the exact compliance.

We can rewrite the convergence requirement of the selective refinement, (16), as

$$\left| \frac{\psi^{i, \text{uniform}} - \psi^{i, \text{selective}}}{\psi^{i, \text{uniform}}} \right| \leq \beta, \tag{30}$$

where $\psi^{i, \text{selective}}$ is the compliance after i times selective refinements.

Using the monotonicity of the compliance, this condition is shown to imply that

$$(1 - \beta)\psi^{i, \text{uniform}} \leq \psi^{i, \text{selective}} \leq \psi^{i, \text{uniform}}. \tag{31}$$

Therefore, as $\beta \rightarrow 0$ and $i \rightarrow \infty$, $\psi^{i, \text{selective}}$ converges to $\psi^{i, \text{uniform}}$, which in turn converges to the exact solution. As β is taken smaller, however, more elements are to be selected for refinement and more computational time is necessary.

For the accuracy measure, β , we used a value of 0.01. Thus,

$$0.99\psi^{i, \text{uniform}} \leq \psi^{i, \text{selective}} \leq \psi^{i, \text{uniform}}. \tag{32}$$

A quarter model with an elliptic hole in 3D with various mesh resolutions was chosen as a verification example for both the uniform refinement and the selective refinement. Figure 8 shows the prescribed pressure on the top surface and geometric boundary conditions for the model where there are three symmetry planes. We based our construction of the initial model, called model 1, with the mesh resolution $12 \times 12 \times 12$ on the FG. With a uniform refinement, as shown in Fig. 9, we refined model 1 to model 2_{uniform} (first refinement) and then model 2_{uniform} to model 3_{uniform} (second refinement). With a selective refinement, as shown in Fig. 10, we obtained model $2_{\text{selective}}$ and model $3_{\text{selective}}$. Figure 10 shows, as expected, that the mesh plot of the selective design space refinement compares well with the sensitivity level plot. Note that the number of nodes in model $3_{\text{selective}}$ is only 4.8% of the number of nodes in model 3_{uniform} . However, the objective function value of compliance remains within a range of 0.04% of the corresponding value in model 3_{uniform} . This result satisfies the prescribed accuracy bound of 1% in (32). The numbers of iterations

Table 1 Comparison between uniform and selective design space refinement for a quarter model with an elliptic hole

Initial resolution		Compliance value	No. of constraint equations	No. of nodes in model 3	(No. of constraints equations)/ (reduced no. of nodes)	Calculation time (s)
3×3×3	Uniform	56.483 (1.000)	0	2,152 (1.000)	0.263	2.797 (1.000)
	Selective	56.336 (0.997)	385	396 (0.184)		1.109 (0.396)
6×6×6	Uniform	56.396 (1.000)	0	15,265 (1.000)	0.082	17.328 (1.000)
	Selective	56.309 (0.998)	1,035	1,312 (0.086)		1.906 (0.111)
9×9×9	Uniform	56.473 (1.000)	0	49,424 (1.000)	0.047	56.359 (1.000)
	Selective	56.420 (0.999)	2,029	2,952 (0.060)		3.516 (0.062)
12×12×12	Uniform	56.436 (1.000)	0	114,760 (1.00)	0.033	141.547 (1.000)
	Selective	56.400 (0.999)	3,412	5,543 (0.048)		6.234 (0.044)
15×15×15	Uniform	56.446 (1.000)	0	221,328 (1.000)	0.026	293.188 (1.000)
	Selective	56.422 (0.999)	5,210	9,493 (0.043)		10.844 (0.037)
18×18×18	Uniform	56.432 (1.000)	0	379,240 (1.000)	0.021	521.344 (1.000)
	Selective	56.415 (0.999)	7,428	14,833 (0.039)		17.812 (0.034)

The values in parentheses are the fraction ratio of the value to the corresponding value from the uniform refinement.

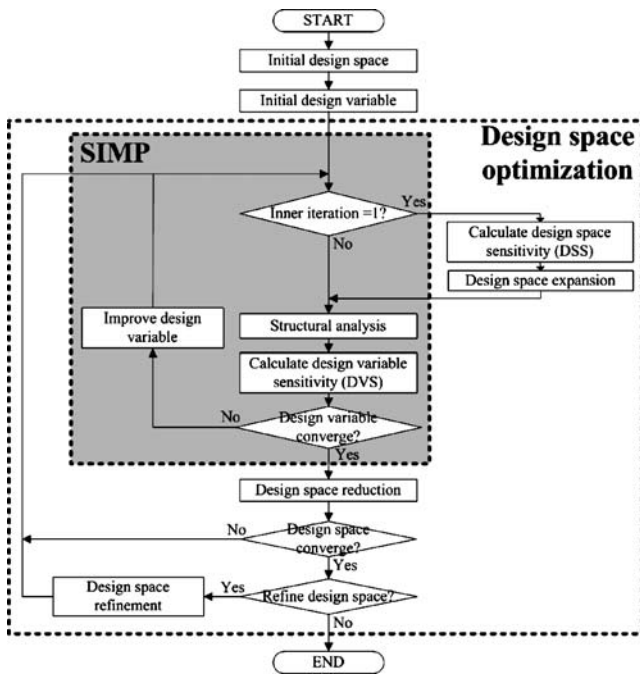


Fig. 11 Flow chart for design space optimization using design space adjustment and refinement

needed for a in the first and second selective refinements are all 5. The computing time is 141.4 s in the case of model 3_{uniform} and 6.2 s in the case of model $3_{\text{selective}}$; that is, model $3_{\text{selective}}$ is 22.7 times faster than model 3_{uniform} .

To check the net effect of the reduced number of nodes and the imposition of nodal constraint equations, we tested the same problem with different mesh resolutions. Table 1

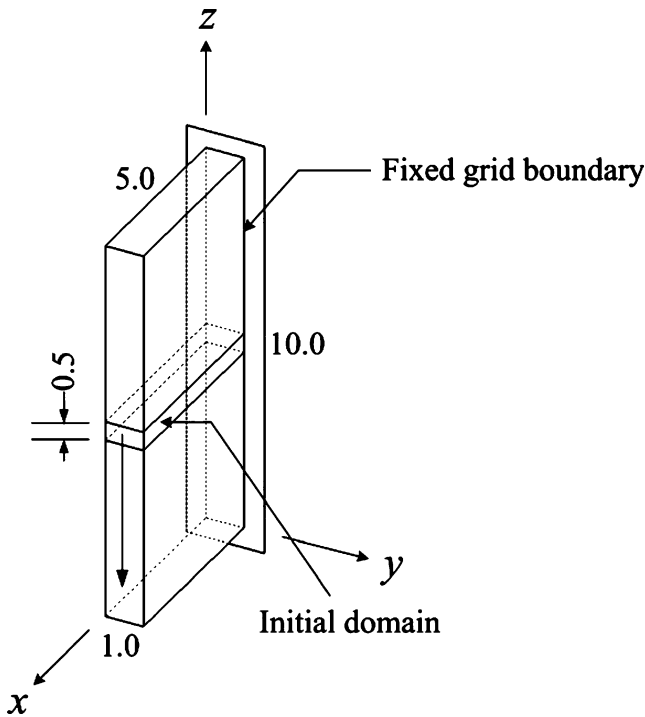
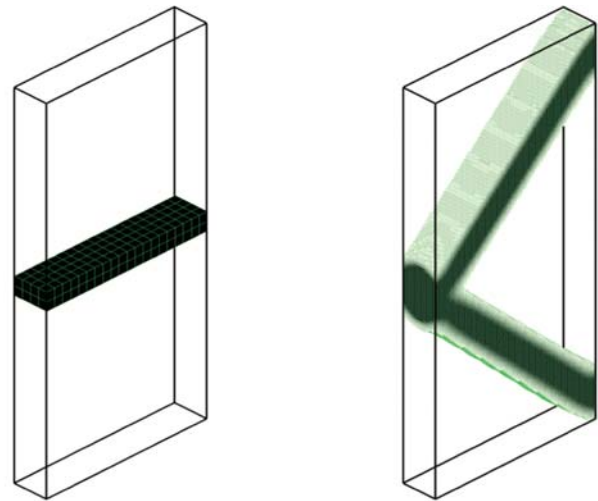


Fig. 12 Initial model and boundary condition for 3D cantilever beam

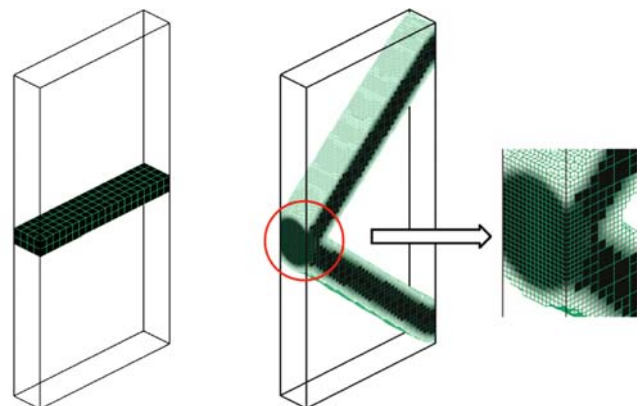


a Initial domain **b** Optimal result

Fig. 13 Optimization results of 3D cantilever beam in the case of uniform design space refinement

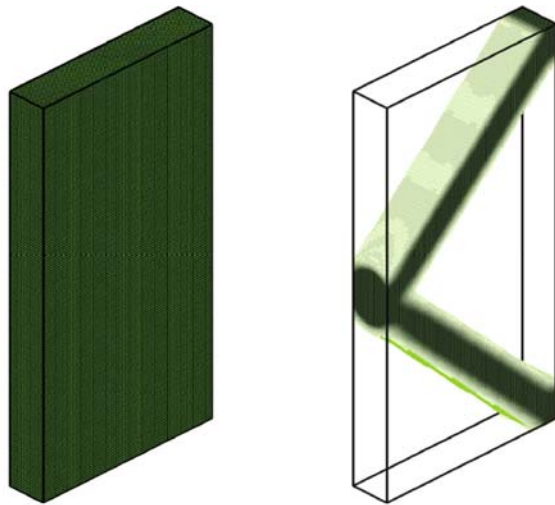
presents the detailed results. It is noted that, as initial resolution goes higher, the ratio of the number of constraint equations to the reduced number of nodes goes lower. As this ratio decreases, calculation time is reduced. From this result, we expect that we will get a good result if we apply the proposed scheme to the structure with low ratio of the number of constraint equations to the reduced number of nodes. On the other hand, if we apply the method to the high-ratio structure, then cost gain may not be much.

These verification examples confirm that we can use the proposed selective refinement to approximate the structural behavior, or compliance, within the prescribed accuracy bound with fewer elements, as well as to improve the clearness of boundaries.



a Initial domain **b** Optimal result

Fig. 14 Optimization results of 3D cantilever beam in the case of selective design space refinement



a Initial domain b Optimal result

Fig. 15 Optimization results of 3D cantilever beam in the case of conventional SIMP

2.3 Procedure of design space optimization with design space adjustment and refinement

There are two routines in design space optimization. The inner routine is the same as a conventional SIMP method (the gray box in Fig. 11) and uses the optimality criterion method. We can express the convergence criterion of the inner routine as follows:

$$\max_i |\rho_i^m - \rho_i^{m-1}| \leq 0.01, \tag{33}$$

where m is the iteration number.

When the inner routine is converged, the outer routine (the white box in Fig. 11) seeks out a better design space for the stage. The design space adjustment, namely the design space expansion and reduction, is made at the outer routine, and a design space refinement is subsequently done to achieve a target refinement level. For the design space reduction, any element whose density is below a threshold value (0.1 here) is simply excluded at the end of the inner routine. A density of 0.1 has little effect on the mechanical characteristics of the structure.

The convergence criterion of the outer routine is expressed as follows:

$$\frac{\sum_{i=1}^{N_{boundary}} (\rho_i)_{boundary}}{N_{boundary}} \leq 0.1, \tag{34}$$

where $N_{boundary}$ denotes the number of boundary elements added by the design space adjustment and $(\rho_i)_{boundary}$ is the density of the i th boundary element. This outer routine can be easily interfaced with the conventional SIMP as shown in Fig. 11.

3 Numerical examples

For our computations, we used an Intel Pentium 4 PC, with a clock frequency of 3.0 GHz and a memory of 3 GB. We solved the examples with four node plane elements, PLANE42 in ANSYS for 2D and eight node brick elements, SOLID45 in ANSYS for 3D.

Table 2 Comparison of optimization results for 3D cantilever beam among uniform and selective refinements and conventional SIMP

Indicators		Design space adjustment					
		1	2	3	4	5 ^d	6 ^e
Number of elements	Uniform ^a	160	1,496	1,368	1,324	9,736	73,708
	Selective ^b	160	1,496	1,368	1,324	7,696	44,488
	Conventional ^c	204,800					
Number of iterations	Uniform	1	24	99	6	55	12
	Selective	1	24	99	6	11	7
	Conventional	54					
Compliance	Uniform	18863.0	89.2	84.2	84.3	115.2	179.7
	Selective	18863.0	89.2	84.2	84.3	115.5	177.4
	Conventional	179.5					

^a Design space optimization with uniform refinement (using both adjustment and refinement)

^b Design space optimization with selective refinement (using both adjustment and refinement)

^c Conventional SIMP

^d First refinement done

^e Second refinement done

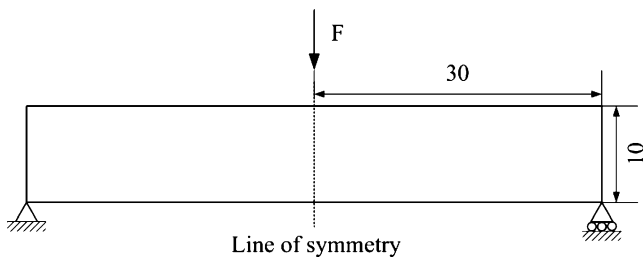


Fig. 16 Initial model and boundary condition for the MBB beam.

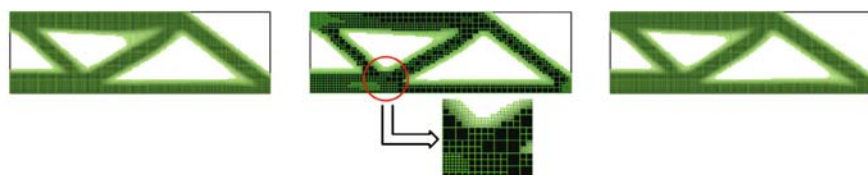
3.1 3D cantilever beam

As a test problem for the open design domain problem, we chose a 3D cantilever structure. Figure 12 shows the initial model and boundary conditions. The volume constraint was $V_0=4V_{\text{initial}}$ and the radius of sensitivity filtering was $R=0.4$. Young's modulus was $E=210.0$ GPa and Poisson's ratio was $\nu=0.3$.

In the case of both uniform and selective refinements, the initial domain was $20 \times 4 \times 2$ with 160 elements. As the optimization proceeded, the design domain expanded in relation to the DSS. The optimum was reached after five steps of design space adjustment and two refinements. The process with the uniform refinement took 197 function calls and 0.92 h of total run time. The maximum number of elements used was 73,708. For the selective refinement, there were 154 function calls, including six for two α . The maximum number of elements was 44,488 and the total run time was 0.52 h. Figures 13 and 14 show the optimization process.

For comparison, the same problem was solved with a conventional SIMP method without a sequence of refinements. The initial domain was $80 \times 16 \times 160$ to ensure that the mesh resolution and aspect ratio were the same as those of our final solution. So, the number of elements was 204,800, which is much larger than the number in the design space optimization, and the process involved 54 function calls and lasted 6.67 h. Figure 15 shows the optimization results. Note that, although the number of function calls in our approach is larger, the total run time is 13.8% of conventional SIMP in the case of the uniform refinement and 7.8% in the case of the selective refinement.

Fig. 17 Optimization results of the MBB beam



a Uniform design space refinement b Selective design space refinement c Conventional SIMP

Table 3 Comparison of optimization results for the MBB beam among uniform and selective refinements, and conventional SIMP

Indicators		Design space adjustment			
		1	2	3 ^d	4 ^e
Number of elements	Uniform ^a	675	595	2,470	9,747
	Selective ^b	675	595	1,886	6,226
	Conventional ^c	10,800			
Number of iterations	Uniform	89	12	13	10
	Selective	89	12	10	7
	Conventional	106			
Compliance	Uniform	8.971	8.981	8.944	9.059
	Selective	8.971	8.981	8.903	8.877
	Conventional	9.049			

^a Design space optimization with uniform refinement (using both adjustment and refinement)

^b Design space optimization with selective refinement (using both adjustment and refinement)

^c Conventional SIMP

^d First refinement done

^e Second refinement done

Table 2 summarizes the comparative optimization results of the following: design space optimization with uniform refinement; design space optimization with selective refinement; and a conventional SIMP method. The number of elements in the step of design space adjustment, where most iterations are needed, was 1,368, 1,368, and 204,800, respectively. These numbers show that we can search for an approximate optimal design space with fewer elements by using either uniform refinement or selective refinement, thereby, considerably shortening the total run time.

3.2 MBB beam

The second example is the MBB beam shown in Fig. 16, with the symmetry line at the center and vertical load at the middle of the upper side. A volume constraint was $V_0=0.5 V_{\text{initial}}$, and the radius of sensitivity filtering was $R=1.0$. Young's modulus and Poisson's ratio were $E=210.0$ GPa and $\nu=0.3$, respectively. In contrast to the first example, the FG boundary has prescribed lines

considering fully constrained design domain that may make the procedure of design space adjustment ineffective. In open design domain problems, one of the major functions of design space optimization, design space adjustment is especially useful to search unknown optimal design space, while in fully constrained design domain, this procedure seems useless. We will check how efficiently the proposed scheme with design space refinement works even in a fully constrained design domain.

In the case of both uniform and selective refinements, the initial design space had 675 elements. Starting from this, we reached an optimal result after four steps of design space adjustments and two refinements. The process with the uniform refinement took 124 function calls and 5.55 min of total run time. The maximum number of elements was 9,747. For the selective refinement, there were 124 function calls, including six for two α . The maximum number of elements was 6,226, and the total run time was 4.91 min.

For comparison, the same problem was solved with a conventional SIMP method. The number of elements was 10,800 to ensure that the mesh resolution was the same as those of our final solution. The process involved 106 function calls and required 14.70 min. As before, the total run time is 37.8% of a conventional SIMP in the case of the uniform refinement and 33.4% in the case of the selective refinement. This example shows that even for a fully constrained design domain problem, design space optimization with both design space adjustment and refinement can be one of the best solution methods. Detailed results are summarized in Fig. 17 and Table 3.

4 Conclusion

We implemented the concept of design space optimization to significantly improve the efficiency and capability of dealing with large-scale problems. Our method involves design space adjustment, which is an evolutionary process of design space expansion and reduction, and design space refinement, which can be done uniformly or selectively whenever and wherever necessary. Moreover, selective refinement is effective for obtaining a target resolution with much fewer elements. With the proposed method of implementation, we first find an approximate design space at a low refinement level, and then increase the refinement level to achieve a more detailed design space. As shown with a 3D cantilever beam, the total computational costs were 13.8 and 7.8% in the case of uniform and selective refinements, respectively, compared to the conventional SIMP with no refinements, and with the MBB beam, they were 37.8 and 33.4%, respectively.

Acknowledgment This work was partially supported by the endowment fund for Samsung Chair Professorship.

References

- Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level set method. *J Comput Phys* 194:363–393
- Bendsoe MP (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1:193–303
- Bendsoe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71:197–224
- Bendsoe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. *Arch Appl Mech* 69:635–654
- Borrvall T, Petersson J (2001) Large-scale topology optimization in 3D using parallel computing. *Comput Methods Appl Mech Eng* 190:6201–6229
- Costa JCA Jr, Alves MK (2003) Layout optimization with h -adaptivity of structures. *Int J Numer Methods Eng* 58:83–102
- Diaz AR, Kikuchi N (1992) Solutions to shape and topology eigenvalue optimization problems using a homogenization method. *Int J Numer Methods Eng* 35:1487–1502
- Hassani B, Hinton E (1998a) A review of homogenization and topology optimization I—homogenization theory for media with periodic structure. *Comput Struct* 69:707–717
- Hassani B, Hinton E (1998b) A review of homogenization and topology optimization II—analytical and numerical solution of homogenization equations. *Comput Struct* 69:719–738
- Hassani B, Hinton E (1998c) A review of homogenization and topology optimization III—topology optimization using optimality criteria. *Comput Struct* 69:739–756
- Haug EJ, Choi KK, Komkov V (1986) Design sensitivity analysis of structural systems. Academic, New York, pp 104–108
- Hughes TJR (1987) The finite element method: linear static and dynamic finite element analysis. Prentice-Hall, Englewood Cliffs, NJ, pp 185–192
- Jang IG, Kwak BM (2006) Evolutionary topology optimization using design space adjustment based on fixed grid. *Int J Numer Methods Eng* 66(11):1817–1840
- Kim IY, Kwak BM (2002) Design space optimization using a numerical design continuation method. *Int J Numer Methods Eng* 53:1979–2002
- Kim H, Garcia MJ, Querin OM, Steven GP, Xie YM (2000) Introduction of fixed grid in evolutionary structural optimization. *Eng Comput* 17:427–439
- Kim TS, Kim JE, Kim YY (2004) Parallelized structural topology optimization for eigenvalue problems. *Int J Solids Struct* 41:2623–2641
- Krog L, Tucker A, Kemp M, Boyd R (2004) Topology optimization of aircraft wing box ribs. In: Proceedings of the 10th AIAA/ISSMO MAO conference. AIAA, Albany
- Maute K, Ramm E (1995) Adaptive topology optimization. *Struct Optim* 10:100–112
- Osher S, Sethian JA (1988) Front propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 78:12–49
- Querin OM, Steven GP (1998) Evolutionary structural optimization (ESO) using a bidirectional algorithm. *Eng Comput* 15:1031–1048
- Sethian J, Wiegmann A (2000) Structural boundary design via level-set and immersed interface methods. *J Comput Phys* 163:489–528
- Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-independencies and local minima. *Struct Optim* 16:68–75

- Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. *Struct Multidiscipl Optim* 22:116–124
- Suzuki K, Kikuchi N (1991) A homogenization method for shape and topology optimization. *Comput Methods Appl Mech Eng* 93:291–318
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192:227–246
- Wang L, Basu PK, Leiva JP (2004) Automobile body reinforcement by finite element optimization. *Finite Elem Anal Des* 40:879–893
- Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. *Comput Struct* 49:885–896
- Yang RJ, Chuang CH (1994) Optimal topology design using linear programming. *Comput Struct* 52:265–275
- Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: topology geometrical and generalized shape optimization. *Comput Methods Appl Mech Eng* 89:309–336