ELSEVIER

Advances in Engineering Software 34 (2003) 321–338

ADVANCES IN ENGINEERING SOFTWARE

www.elsevier.com/locate/advengsoft

Mesh generation for lower bound limit analysis

Andrei V. Lyamin*, Scott W. Sloan

Faculty of Engineering and Built Environment, University of Newcastle, Newcastle, NSW 2308, Australia

Received 13 September 2002; accepted 14 February 2003

Abstract

This paper describes a general strategy for generating lower bound meshes in *D*-dimensions. The procedure is based on a parametric mapping technique, coupled with midpoint splitting of subdomains, and permits the user to control the distribution of the discontinuities and elements precisely. Although it is not fully automatic, the algorithm is fast and automatically generates extension zones for problems with semi-infinite domains.

Keywords: Mesh generation; Lower bound; Limit analysis

1. Introduction

Lower bound finite element methods, such as those described by Lyamin [1] and Lyamin and Sloan [2], have become increasingly popular for predicting continuum collapse loads. Unlike conventional finite element meshes, a lower bound grid must be able to incorporate stress discontinuities along faces that are shared by adjacent elements. Since these stress discontinuities are of zero thickness, several nodes may share the same coordinates and additional complexity is introduced into the mesh generation process. In general, the role of the discontinuities is crucial as their arrangement and distribution has a dramatic influence on the accuracy of the lower bound solution [3]. To ensure that the discontinuities are correctly positioned requires precise control in the mesh generation phase. The need for this control is further reinforced by the fact that only linear finite elements (where the stresses vary linearly over the element) can be used in the lower bound analysis. Higher order elements, which are often effective in conventional finite element analysis, are not an option in the lower bound method since they lead to stress fields which may not be statically admissible.

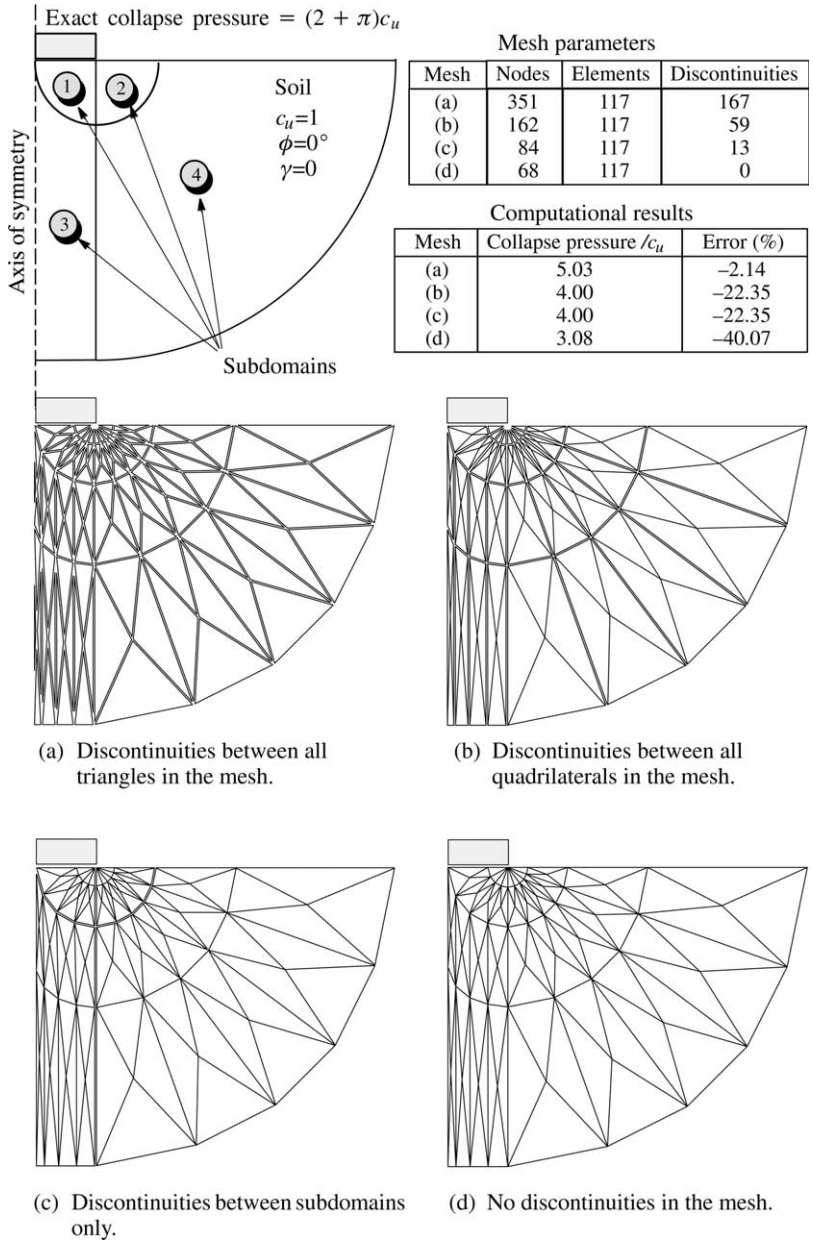
To illustrate the importance of incorporating stress discontinuities in a lower bound analysis, consider the behaviour of a rigid strip footing resting on a purely cohesive weightless soil. The exact collapse pressure for this case is $p = (2 + \pi)c_u$, where c_u is the undrained shear strength of the soil. Fig. 1(a) shows that this solution is matched to within about 2% for a 'fan' mesh with discontinuities at all element edges. Removing these discontinuities leads to a lower bound which is some 40% below the exact collapse pressure (Fig. 1(d)). Meshes with other distributions of discontinuities, as depicted in Fig. 1(b) and (c), furnish solutions which lie between these two extremes.

2. Lower bound mesh generation by parametric mapping

Because of its ability to control the position and distribution of the elements and discontinuities precisely, a parametric mapping algorithm is well suited for generating lower bound meshes. The method described here requires the domain to be initially partitioned into a set of mappable subdomains, each with its own specified subdivision information, that will yield the type of mesh desired. Although automatic subdivision techniques based on feature recognition have recently emerged [5,6], these are not well suited to the present application as they only control the shape of the generated elements and not the distribution of the discontinuities. In the new generator described here, the division of the object into compatible subdomains is done

* Corresponding author. Tel.: +61-2-4921-7073; fax: +61-2-4921-6991.
E-mail address: andrei.lyamin@newcastle.edu.au (A.V. Lyamin).

0965-9978/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0965-9978(03)00032-2



Mesh parameters

Mesh	Nodes	Elements	Discontinuities
(a)	351	117	167
(b)	162	117	59
(c)	84	117	13
(d)	68	117	0

Computational results

Mesh	Collapse pressure /c_u	Error (%)
(a)	5.03	–2.14
(b)	4.00	–22.35
(c)	4.00	–22.35
(d)	3.08	–40.07

Fig. 1. Influence of mesh discontinuities on accuracy of lower bounds for strip footing on purely cohesive soil ('fan' mesh).

manually and then verified and adjusted (if necessary) automatically. This approach has proved to be sufficiently general for a wide variety of geotechnical problems, and is a powerful auxiliary tool for the solution of complex 3D lower bound problems.

Although the need to subdivide the body into mappable subdomains is a disadvantage when the geometry is very complex, it is a natural approach for inhomogeneous geotechnical problems involving layered soils. Additional advantages of adopting this strategy for the lower bound formulation are:

- It is simple to generate a regular pattern of discontinuities throughout the body.
- The same program can be used to generate meshes of different dimensionality.
- The geometry, material properties, loading and boundary conditions can be modelled using the same mapping technique.
- Extension elements can be generated automatically for problems with semi-infinite domains.
- The input data has a readable, regular structure which is convenient to use.
- Apart from the manual subdivision stage, the mesh generation is very fast.

The two major phases in the new mesh generation scheme are:

1. Subdividing the domain into subdomains which are simple to mesh directly.
2. Meshing each subdomain with discontinuities and elements according to a specified density and distribution.

Each of these aspects are now considered in turn.

3. Decomposition into subdomains

As shown in Fig. 2, the domain subdivision for most practical problems is governed by the key features of the geometry, material properties and applied loading. This implies that the resulting subdomains have to be mappable in terms of these quantities. Although a number of different mapping techniques are possible, the simplest and most suitable for lower bound grid generation is the isoparametric mapping. This technique, first described by Zienkiewicz and Philips [9], uses Lagrange polynomials to model each subdomain boundary and provides a natural means of generating the nodal coordinates and element topology arrays automatically. Assuming that all the subdomains are quadrilateral (or hexahedral in the 3D case), the relevant isoparametric mappings for the geometry and the material variation may be written as:

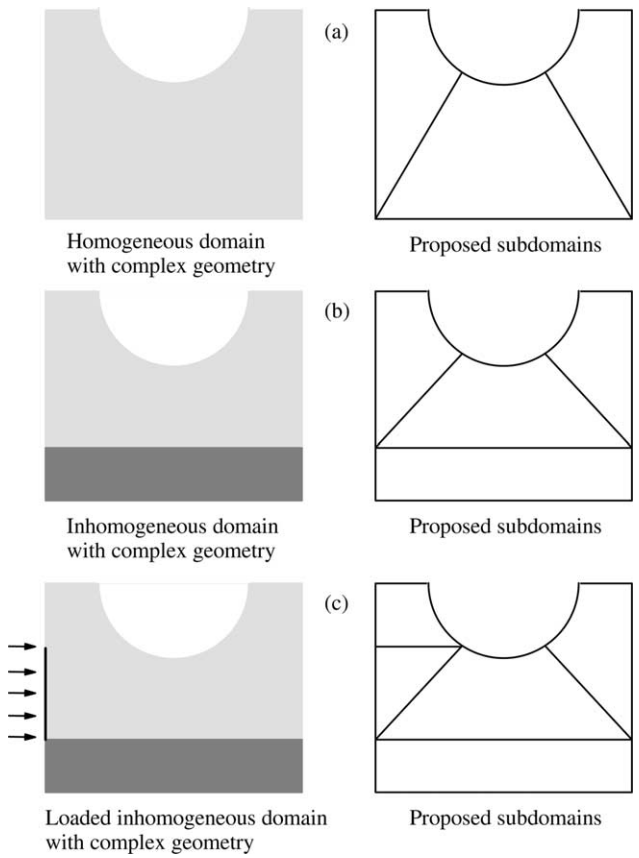


Fig. 2. Domain subdivision considering complexity of (a) geometry, (b) geometry and material properties, and (c) geometry, material properties and applied loading.

2D

$$\mathbf{X}(\xi,\eta)=\sum_{i=1}^{m_X} N_i(\xi,\eta)\mathbf{X}_i \tag{1}$$

$$\mathbf{M}(\xi,\eta)=\sum_{i=1}^{m_M} N_i(\xi,\eta)\mathbf{M}_i$$

3D

$$\mathbf{X}(\xi,\eta,\zeta)=\sum_{i=1}^{m_X} N_i(\xi,\eta,\zeta)\mathbf{X}_i \tag{2}$$

$$\mathbf{M}(\xi,\eta,\zeta)=\sum_{i=1}^{m_M} N_i(\xi,\eta,\zeta)\mathbf{M}_i$$

where $\mathbf{X}$ and $\mathbf{M}$ are the global coordinates and material properties of a point in space, respectively, (ξ,η,ζ) are the local coordinates of the reference subdomain (whose values range from -1 to $+1$), m_X and m_M are the number of subdomain nodes for the geometry and material representation, respectively, N_i is the assumed shape functions associated with subdomain node i, and $\mathbf{X}_i$ and $\mathbf{M}_i$ are the coordinates and material properties of the subdomain nodes, respectively. In the mesh generator, both *linear* and *quadratic* 'serendipity' shape functions are used to describe the variation of the geometry and material properties over each subdomain. Even though the variation of these quantities is always linear throughout each generated element, the use of quadratic shape functions is convenient for modelling curved boundaries or complex material variations with a minimum number of subdomains. For cases with non-linear geometries or complicated material variations, the accuracy of the linearisation can be controlled simply by adopting an appropriate element density. Full descriptions of the subdomains used in two and three dimensions, and their parametric mappings, are presented in Figs. 3 and 4.

It is also convenient to specify the applied surface tractions and body forces using an isoparametric mapping approach. To this end the domain should be subdivided so that the distribution of the forces can be modelled using the following parametric representation

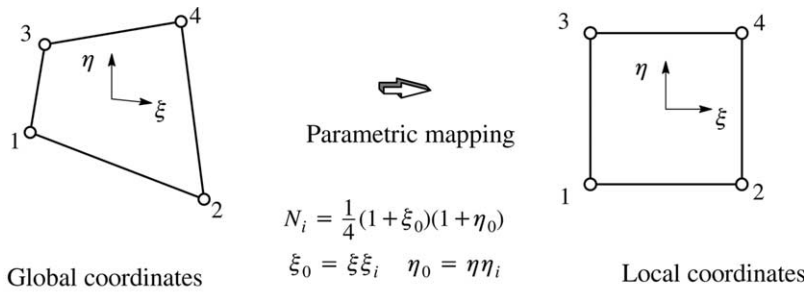
2D

$$\mathbf{t}(\xi')=\sum_{i=1}^{m_t} N_i(\xi')\mathbf{t}_i,\qquad \mathbf{q}(\xi')=\sum_{i=1}^{m_q} N_i(\xi')\mathbf{q}_i \tag{3}$$

$$\mathbf{g}(\xi,\eta)=\sum_{i=1}^{m_g} N_i(\xi,\eta)\mathbf{g}_i,\qquad \mathbf{h}(\xi,\eta)=\sum_{i=1}^{m_h} N_i(\xi,\eta)\mathbf{h}_i$$

3D

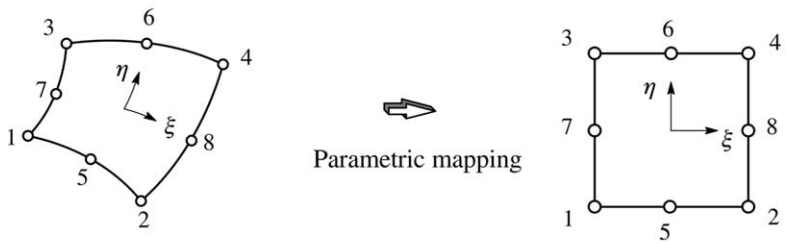
$$\mathbf{t}(\xi',\eta')=\sum_{i=1}^{m_t} N_i(\xi',\eta')\mathbf{t}_i,\qquad \mathbf{q}(\xi',\eta')=\sum_{i=1}^{m_q} N_i(\xi',\eta')\mathbf{q}_i \tag{4}$$



$$N_i = \frac{1}{4}(1+\xi_0)(1+\eta_0)$$

$$\xi_0 = \xi\xi_i \quad \eta_0 = \eta\eta_i$$

(a) Linear subdomain (4 nodes)



Corner nodes

$$N_i = \frac{1}{4}(1+\xi_0)(1+\eta_0)(\xi_0+\eta_0-1)$$

Midside nodes

$$\xi_i = 0 \quad N_i = \frac{1}{2}(1-\xi^2)(1+\eta_0)$$

$$\eta_i = 0 \quad N_i = \frac{1}{2}(1-\eta^2)(1+\xi_0)$$

(b) Quadratic subdomain (8 nodes)

Local coordinates of nodes

Node	1	2	3	4	5	6	7	8
ξ	–1	1	–1	1	0	0	–1	1
η	–1	–1	1	1	–1	1	0	0

Fig. 3. Two-dimensional linear and quadratic 'serendipity' subdomains.

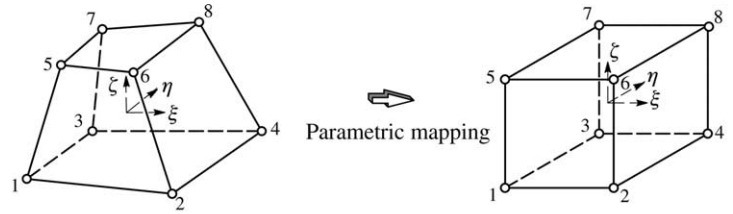
$$\mathbf{g}(\xi,\eta,\zeta) = \sum_{i=1}^{m_g} N_i(\xi,\eta,\zeta)\mathbf{g}_i, \qquad \mathbf{h}(\xi,\eta,\zeta) = \sum_{i=1}^{m_h} N_i(\xi,\eta,\zeta)\mathbf{h}_i$$

where $\mathbf{t}$ are prescribed surface tractions, $\mathbf{g}$ are prescribed body forces, $\mathbf{q}$ are unknown surface tractions which are to be optimised, $\mathbf{h}$ are unknown body forces which are to be optimised, m_t, m_g, m_q and m_h are the numbers of subdomain nodes for $\mathbf{t}$, $\mathbf{g}$, $\mathbf{q}$ and $\mathbf{h}$, respectively, $\mathbf{t}_i$, $\mathbf{g}_i$, $\mathbf{q}_i$ and $\mathbf{h}_i$ are nodal values of $\mathbf{t}$, $\mathbf{g}$, $\mathbf{q}$ and $\mathbf{h}$, respectively, and (ξ', η') are local coordinates along the side of each subdomain. The use of parametric mapping to describe the geometry, material variation and applied loads over a typical subdomain is shown in Fig. 5.

This type of mapping is not only very general, but also has the additional benefit of employing a consistent structure for the input data. One of its important special features is that it permits the generation of a 'fan' of discontinuities that can be centered on a singularity in the stress distribution. These singularities, one of which is shown in Fig. 6, commonly occur where there is an abrupt change in the boundary conditions. The advantage of these stress fans is that each discontinuity allows a jump in the tangential stress, thus providing the potential for a rapid change in the stress field and a higher lower bound. The magnitude of this benefit is shown in Fig. 6, where the number of discontinuities passing through the edge of the rigid footing has a marked influence on the accuracy of the lower bound solution. Note that although the shape of the elements can be quite distorted in regions with a high density of discontinuities, this has little effect on the stability of the lower bound technique as the method does not involve integration over the element volumes.

Since the shape of the generated elements is not a key consideration in our analysis, the domain can be subdivided using only quadrilateral (or hexahedral) subdomains which, if necessary, can have their edges collapsed to furnish the desired discontinuity pattern. These 'degenerate' subdomains still have the same number of nodes as normal subdomains, the only difference is that all of the nodes along a collapsed edge have the same physical coordinates. All the shapes that can be approximated using this technique are shown in Fig. 7. It is worth mentioning here that, like the usual parametric transformation, no internal angle in the collapsed subdomains can be greater than 180°.

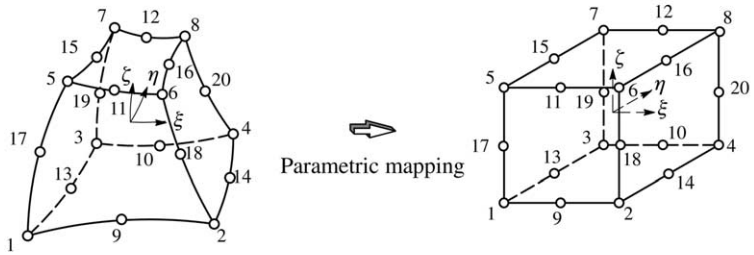
The input data required for the mapped subdomain method has a consistent structure which is simple to use and modify. It may be summarised as follows:



$$N_i = \frac{1}{8}(1+\xi_0)(1+\eta_0)(1+\zeta_0)$$

$$\xi_0 = \xi\xi_i \quad \eta_0 = \eta\eta_i \quad \zeta_0 = \zeta\zeta_i$$

(a) Linear subdomain (8 nodes)



$$\text{Corner nodes} \quad N_i = \frac{1}{8}(1+\xi_0)(1+\eta_0)(1+\zeta_0)(\eta_0+\xi_0+\zeta_0-2)$$

$$\xi_i = 0 \quad N_i = \frac{1}{4}(1-\xi^2)(1+\eta_0)(1+\zeta_0)$$

$$\text{Midside nodes} \quad \eta_i = 0 \quad N_i = \frac{1}{4}(1-\eta^2)(1+\zeta_0)(1+\xi_0)$$

$$\zeta_i = 0 \quad N_i = \frac{1}{4}(1-\zeta^2)(1+\xi_0)(1+\eta_0)$$

(b) Quadratic element (20 nodes)

Local coordinates of nodes

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ξ	−1	1	−1	1	−1	1	−1	1	0	0	0	0	−1	1	−1	1	−1	1	−1	1
η	−1	−1	1	1	−1	−1	1	1	−1	1	−1	1	0	0	0	0	−1	−1	1	1
ζ	−1	−1	−1	−1	1	1	1	1	−1	−1	1	1	−1	−1	1	1	0	0	0	0

Fig. 4. Three-dimensional linear and quadratic 'serendipity' subdomains.

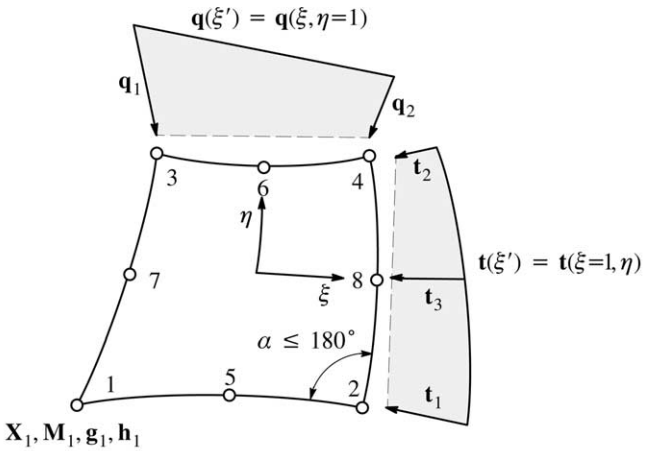
Control information

1. Global mesh description parameters.
2. Numbered set of material properties.
3. Numbered set of extension directions to generate extension elements.
4. Numbered set of prescribed surface tractions.
5. Numbered set of optimisable surface tractions.
6. Numbered set of prescribed body forces.
7. Numbered set of optimisable body forces.

For each subdomain

1. Local mesh description parameters.
2. Number of subdivisions along each local coordinate.
3. Grading coefficients to control the subdivision spacing along each local coordinate.
4. Material property number, prescribed body force number, and optimisable body force number.
5. Coordinates of all subdomain nodes.
6. Numbers of sides with prescribed surface tractions and the numbers of these tractions.
7. Numbers of sides with optimisable surface tractions and the numbers of these tractions.
8. Numbers of sides shared between subdomains and the continuity conditions on these sides (the latter controls the generation of discontinuities between subdomains).
9. Numbers of sides with extension conditions and the numbers of these conditions.

To automate the matching of subdivisions along each boundary between adjacent subdomains, which is known as compatibility control, it is convenient to subdivide the object using the so-called 'chequer board' technique.

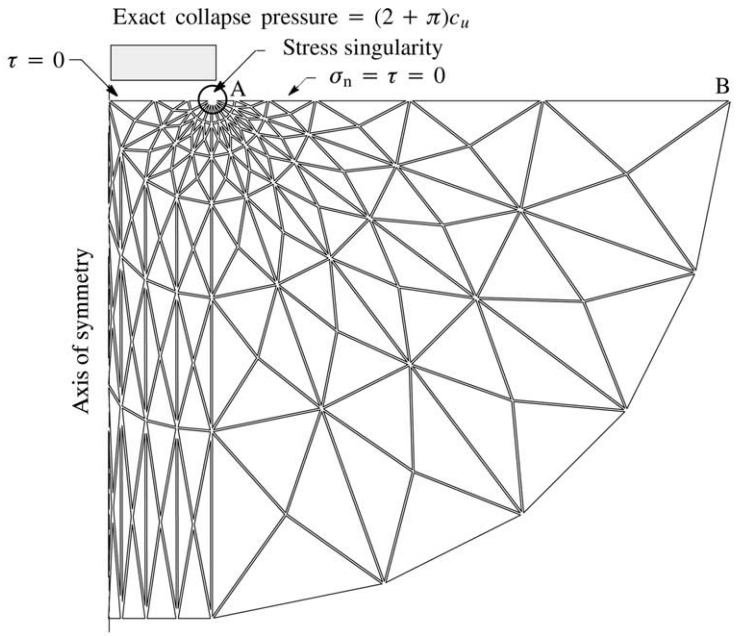


Quantity	X	M	g	h	t	q
Dimensionality	2	2	2	2	1	1
Shape function order	2	1	1	0	2	1
Number of nodes	8	4	4	1	3	2
Nodes	1,2,3,4 5,6,7,8	1,2,3,4	1,2,3,4	1	2,4,8	3,4

Fig. 5. Parametric mapping of geometry, material properties and applied loads for quadrilateral subdomain.

This method views the subdomains as a topologically equivalent rectangular block diagram, forming a chequer board pattern [8] and, if necessary, introduces void zones to account for regions where the mesh is not generated. The chequer board pattern scheme suggested by Zienkiewicz and Phillips [9] consists of four major steps, namely:

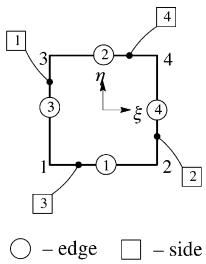
1. Divide the given domain into quadrilateral (hexahedral) subdomains.
2. Transform the subdomains into a topologically equivalent chequer board pattern.
3. Generate the mesh in terms of local coordinates in the transformed plane.



Number of discontinuities passing through point A*	1	2	3	4	5	8	12	20
Collapse pressure/c_u	4.05	4.53	4.80	4.91	4.98	5.05	5.07	5.07
Error (%)	–21.2	–11.9	–6.61	–4.47	–3.11	–1.75	–1.36	–1.36

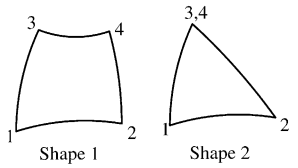
* All results are for the same number of divisions (five) along AB.

Fig. 6. Influence of discontinuity fan on accuracy of lower bounds for strip footing on purely cohesive soil.



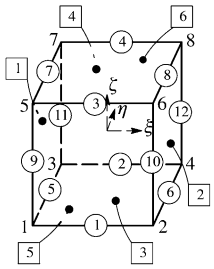
Edge	1	2	3	4
Nodes	1,2	3,4	1,3	2,4

Side	1	2	3	4
Nodes	1,3	2,4	1,2	3,4



Shape no.	1	2
Collapsed edge no.	–	2

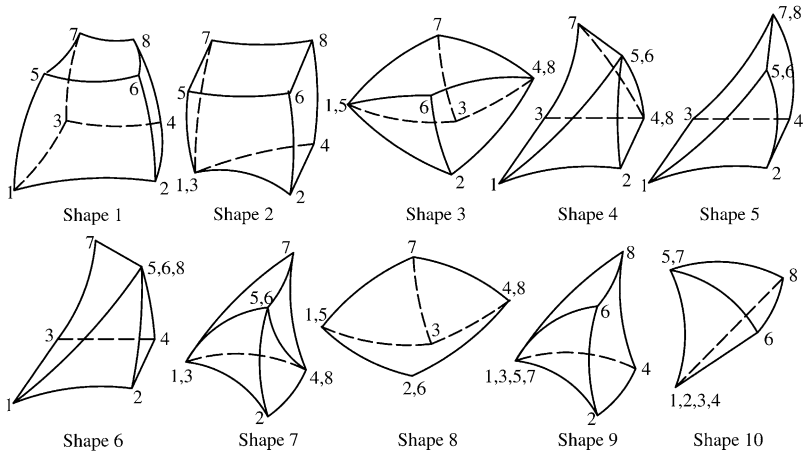
(a) Planar collapsed quadrilateral subdomains



Edge	1	2	3	4	5	6	7	8	9	10	11	12
Nodes	1,2	3,4	5,6	7,8	1,3	2,4	5,7	6,8	1,5	2,6	3,7	4,8

Side	1	2	3	4	5	6
Nodes	1,3,5,7	2,4,6,8	1,2,5,6	3,4,7,8	1,2,3,4	5,6,7,8

Midside nodes omitted from figures for clarity



Shape No.	1	2	3	4	5	6	7	8	9	10
Collapsed edge no.	–	5	9,12	3,12	3,4	3,8	3,5,12	9,10,12	5,7,9,11	1,2,5,6,7

(b) Volumetric collapsed hexahedron subdomains

Fig. 7. Planar and volumetric mappable subdomains with collapsed edges.

4. Transform the local nodal coordinates to global nodal coordinates.

A simple 2D example which illustrates each of these steps is shown in Fig. 8. The results shown in Figs. 1 and 6 highlight the need to have precise control over the location of the stress discontinuities in the mesh generation phase. Indeed, this ability is essential for large scale problems where the exact solution is unknown, as it provides the potential to investigate the sensitivity of the solution to the number and pattern of discontinuities in the mesh. In the lower bound mesh generator, the continuity conditions between each subdomain is controlled by the user, as is the distribution of the discontinuities between each of the generated elements. Using the input information, the program automatically glues the subdomains together and performs compatibility checks along their common boundaries changing, if necessary, the coordinates of midside nodes, the subdivision numbers and the subdivision grading coefficients. As a result of this procedure, a D-dimensional connectivity array is formed and used to verify the subdomain compatibility data. Once this task is completed, the mesh is generated parametrically for each subdomain in an independent fashion. The key steps involved in forming the elements are described in Section 4.

Before discussing how each subdomain is split into finite elements, it is useful to mention the node numbering system that is used in the mesh generation scheme. For the hexahedral subdomain node numbering shown in Fig. 4, the local coordinates for a particular node can be computed

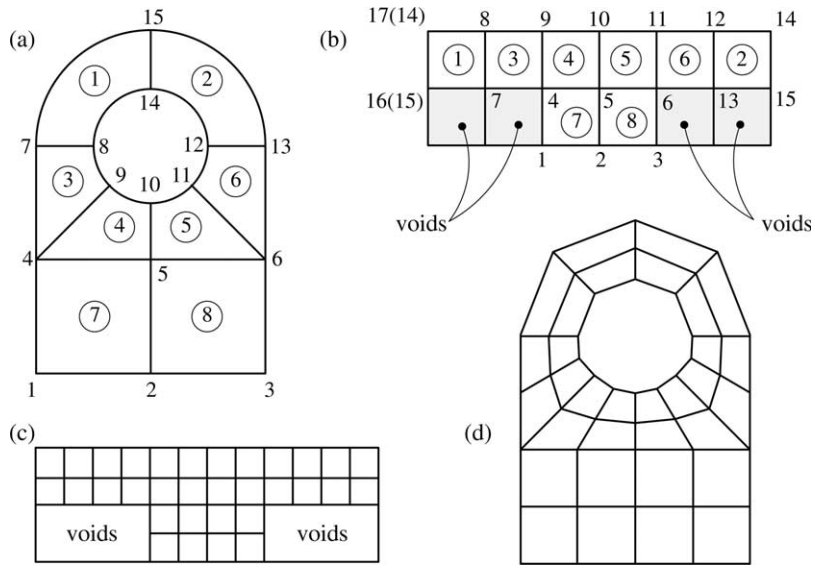


Fig. 8. The four steps in the 'chequerboard' pattern mesh generation scheme.

once we know its number. For example, the local coordinates for the ith corner node of a hexahedron are given by the sequence

$$\begin{aligned} i &= 1,2,\ldots,2^3 \quad j = \frac{i-1}{2^2} \quad \zeta = 2j-1 \\ k &= 1-2^2 j \quad j = \frac{k-1}{2^1} \quad \eta = 2j-1 \\ l &= k-2^1 j \quad j = \frac{l-1}{2^0} \quad \xi = 2j-1 \end{aligned} \tag{5}$$

where all operations in the first two columns are integer and all operations in the last column are real. A similar procedure can also be derived for the midside nodes. Using this type of approach, it is possible to implement the scheme in a form which is dimensionally independent and therefore general.

4. Meshing of subdomains

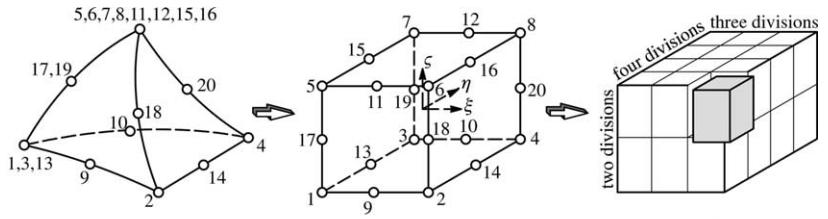
After the domain is defined by a set of quadrilateral (hexahedral) subdomains, the mesh is generated by automatically subdividing each subdomain into a specified number of quadrilateral (hexahedral) blocks which are then split into a set of linear triangular (tetrahedral) elements. These steps are all executed in the transformed curvilinear space, where the subdivision and splitting is performed on square (cubic) blocks. Once the triangles (tetrahedra) have been generated in this space, the resulting local coordinates are transformed to global coordinates in physical space using the transformation (2). The complete sequence of steps for a single subdomain with collapsed edges is shown in Fig. 9.

As shown in Fig. 10, a square (cubic) subdomain can be subdivided into smaller rectangular (hexahedral) blocks in parametric space using either equal or graded subdivisions along each parametric coordinate. The latter are employed if mesh refinement is needed to improve the accuracy of the lower bound analysis. Two different strategies can be used, either separately or in tandem, to produce the desired element proportions in the generated mesh. The first method, which is restricted to quadratic subdomain elements, adjusts the position of the midside nodes from their central location to give a nonuniform distribution of element density. Note that the position of these nodes cannot be chosen arbitrarily, since the one-to-one parametric mapping between the local coordinates and the global coordinates must be preserved. According to Steinmueller [7], all midside nodes must be located in the middle half of the edge to avoid non-uniqueness in the parametric mapping process. Zienkiewicz and Taylor [10] are more conservative and recommend that the 'middle third' rule shown in Fig. 10(b) is safer.

The second, and more flexible, form of mesh grading employs suitable weighting factors to generate unequal subdivisions, as shown in Fig. 10(c). To describe this technique, consider the division of a one-dimensional curved segment AB of length S into n intervals using a geometric progression. Depending on whether the weighting factor w is less than, equal to, or greater than 1, the generated intervals from point A to point B will be, respectively, progressively shorter, equal, or progressively longer. By induction, the sum S of the first n terms of the geometric progression can be written as

$$\begin{aligned} S = S_n &= \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} a_1 w^{i-1} = a_1(1 + w + \cdots + w^{n-1}) \\ &= a_1 \frac{1-w^n}{1-w} \end{aligned} \tag{6}$$

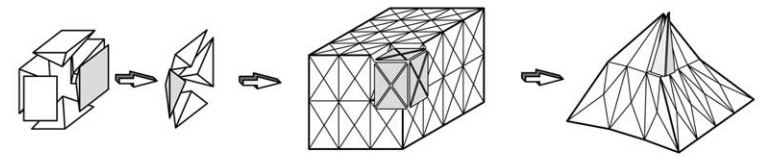
where $w = a_{i+1}/a_i \neq 1$ and a_1 is the first term. Since w and S are known for a given edge, Eq. (6) gives



(a) Subdomain to be meshed (hexahedron with collapsed edges).

(b) Subdomain mapped to cube in parametric space using quadratic shape functions.

(c) Cube subdivided into hexahedral blocks.



(d) Each hexahedron split into tetrahedral elements using midpoint scheme.

(e) Resulting mesh in parametric space.

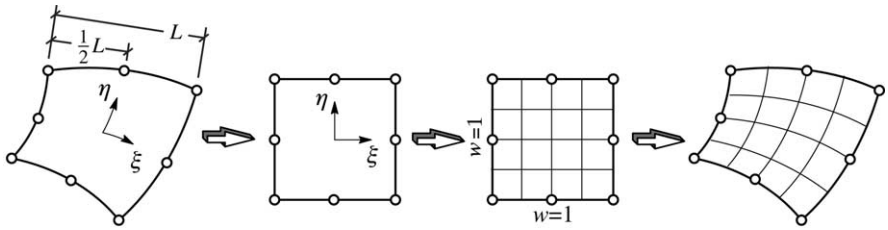
(f) Resulting mesh in physical space*.

* Elements with collapsed edges are not generated.

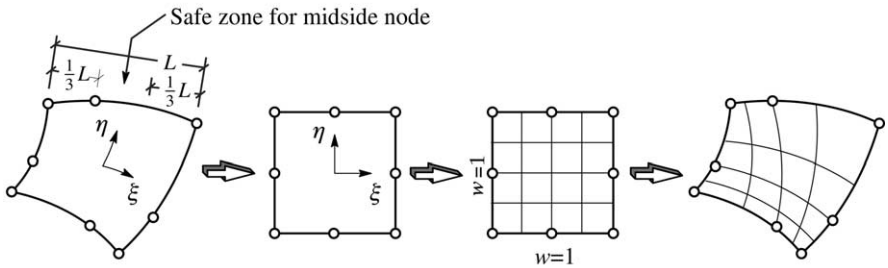
Fig. 9. Filling a volumetric subdomain with tetrahedra.

$a_1 = S(1 - w)/(1 - w^n)$ and the ith term a_i can be found by definition as $a_i = a_1 w^{i-1}$. If X_A and X_B denote the parametric coordinates of the endpoints A and B, respectively, and n denotes the number of subdivisions along the line, then the coordinate of a point i is

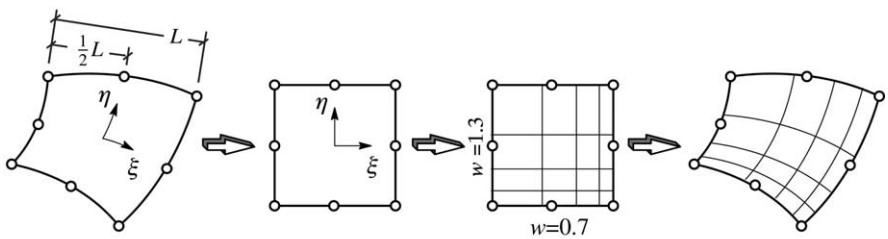
$$X_i = (X_B - X_A)\frac{1 - w^{i-1}}{1 - w^n} + X_A \tag{7}$$



(a) Equal subdivision: Midside nodes in centre of edges with unit weighting factor w.



(b) Graded subdivision: Midside nodes shifted off-centre with unit weighting factor w.



(c) Graded subdivision: Midside nodes in centre of edges with weighting factor w of 0.7.

Fig. 10. Equal and graded subdivision of quadrilateral subdomain.

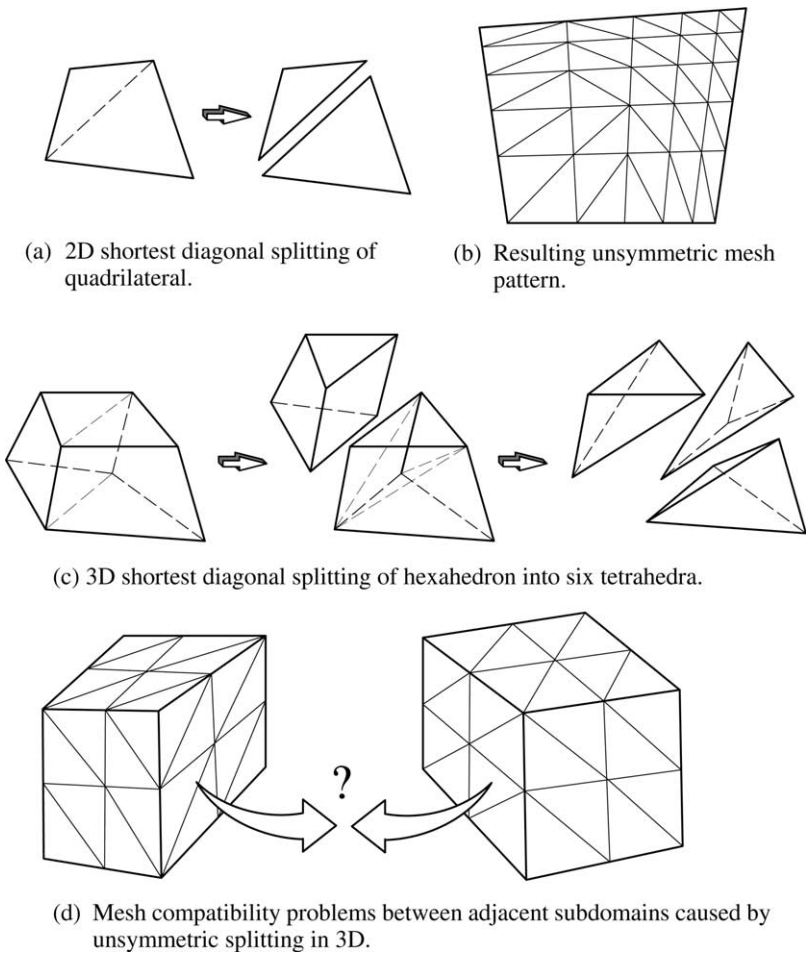
where i is the number of nodes counting from point A. Provided $w \neq 1$, Eq. (7) is employed to evaluate the coordinates of all intermediate nodes when each mapped subdomain is subdivided in parametric space. For convenience, the same weighting factor is used for the two opposite sides of each subdomain and the local coordinates of the endpoints are taken as $X_A = -1$ and $X_B = +1$.

In some applications it is more convenient to define the weighting factor w as the ratio between the lengths of the first and last segments over the interval. For this type of weighting scheme, the expression (7) is replaced by

$$X_i = (X_B - X_A)\frac{1 - w^{(i-1)/n}}{1 - w} + X_A \tag{8}$$

After each mapped subdomain has been subdivided into the required number of quadrilaterals (hexahedra), the next step in the mesh generation procedure is to split these blocks into triangles (tetrahedra). One method for implementing this phase is to split the blocks along their shortest diagonals, as shown in Fig. 11(a) and (c). Although it is convenient and simple, shortest diagonal splitting has a number of significant drawbacks for generating lower bound meshes. Firstly, as shown in Fig. 11(b), there is no symmetry in the generated mesh pattern so that it is not possible to control the precise pattern of the stress discontinuities between the elements. This lack of symmetry in the 3D case also causes mesh compatibility problems between adjacent subdomains, even when the parent mesh is fully compatible (Fig. 11(d)). Secondly, the method needs different logic for splitting triangles and tetrahedra, thus restricting the development of dimensionally independent software.

In view of the above characteristics, the shortest diagonal strategy is generally not well suited for forming lower bound meshes. An alternative splitting scheme, known as the midpoint method, was adopted in the present generator as it creates symmetric meshes, does not need additional compatibility control between adjacent subdomains, and can be generalised to decompose a D-dimensional mapped shape into simplex elements.



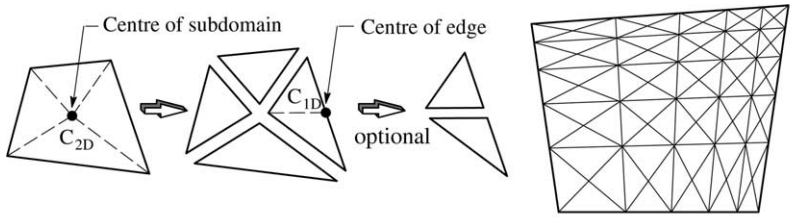
(a) 2D shortest diagonal splitting of quadrilateral.

(b) Resulting unsymmetric mesh pattern.

(c) 3D shortest diagonal splitting of hexahedron into six tetrahedra.

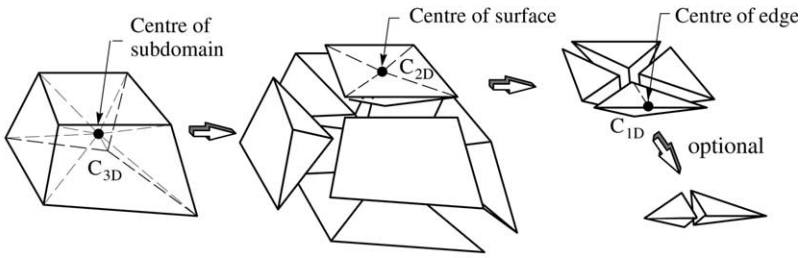
(d) Mesh compatibility problems between adjacent subdomains caused by unsymmetric splitting in 3D.

Fig. 11. Disadvantages of shortest diagonal splitting strategy.

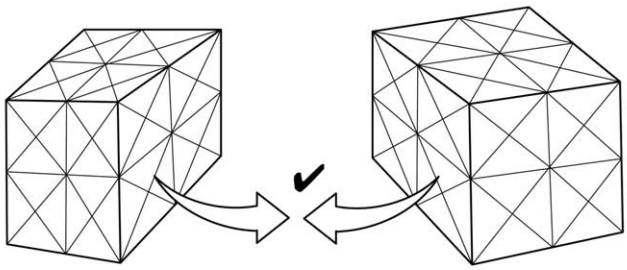


(a) Planar splitting into 4 (or 8) triangles using midpoint strategy.

(b) Uniform distribution of discontinuities.



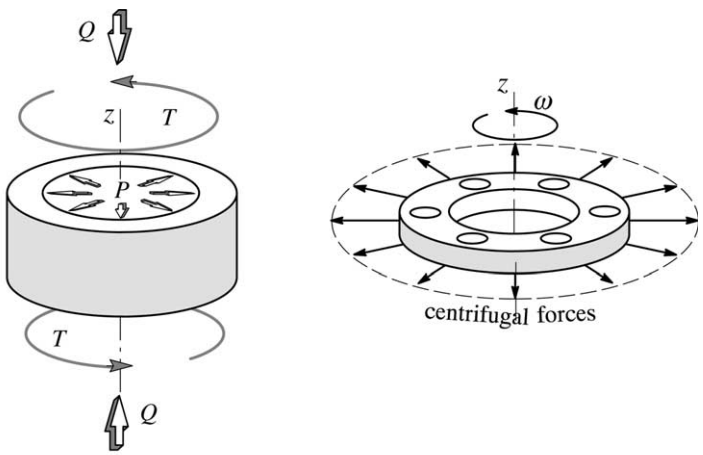
(c) Volumetric splitting into 24 (or 48) tetrahedra using midpoint strategy.



(d) No mesh compatibility problems with midpoint splitting in 3D.

Fig. 12. Midpoint decomposition strategy for planar and volumetric subdomains with optional 'fine' splitting.

The midpoint scheme for subdividing planar and volumetric subdomains is shown, respectively, in Fig. 12(a) and (c). The uniform and predictable pattern of the generated discontinuities and elements, shown in Fig. 12(b), suit the lower bound technique ideally. Moreover, in 3D applications, any compatibility problems between adjacent subdomains are avoided (Fig. 12(d)). If a more refined mesh is required the optional splitting is invoked.



(a) Thick tube under combined torsion, internal pressure, and compression.

(b) Rotation of a wheel drum.

Fig. 13. Examples of problems with loading in local coordinates.

5. Local coordinate systems

It is very common to specify the loading for a lower bound analysis in terms of an auxiliary coordinate system which is different from the usual global Cartesian coordinate system. This is convenient, for example, when normal and shear tractions are applied to surfaces, or when centrifugal body forces are imposed over a volume (Fig. 13). Such a system is generally specific for each point and is, therefore, usually known as a local coordinate system. Let us assume that the local coordinate system $\mathbf{x}'_k$; $k = 1, 2, \ldots, D$ can be obtained by rotating the global Cartesian system $\mathbf{x}_i$; $i = 1, 2, \ldots, D$ about the origin. Under this assumption, the only information we need to perform transformations from one system to the other is the matrix of direction cosines β_{ki} which orient the $\mathbf{x}'_k$-axis with respect to the $\mathbf{x}_i$-axis. Indeed, the coordinates and stresses in the two systems are related by the linear transformation

$$x'_k = \beta_{ki} x_i \tag{9}$$

$$\sigma_{km} = \sigma_{ij} \beta_{ki} \beta_{jm} \tag{10}$$

Depending on the particular situation, the local coordinate system can be constructed according to one of the following options:

1. Using topological information for a single finite element.
2. Using topological information for a subdomain.
3. Using some prescribed rule (e.g. a table or a formula).

The sequence of calculations involved in these variants is shown in Fig. 14. Each option has its own advantages and

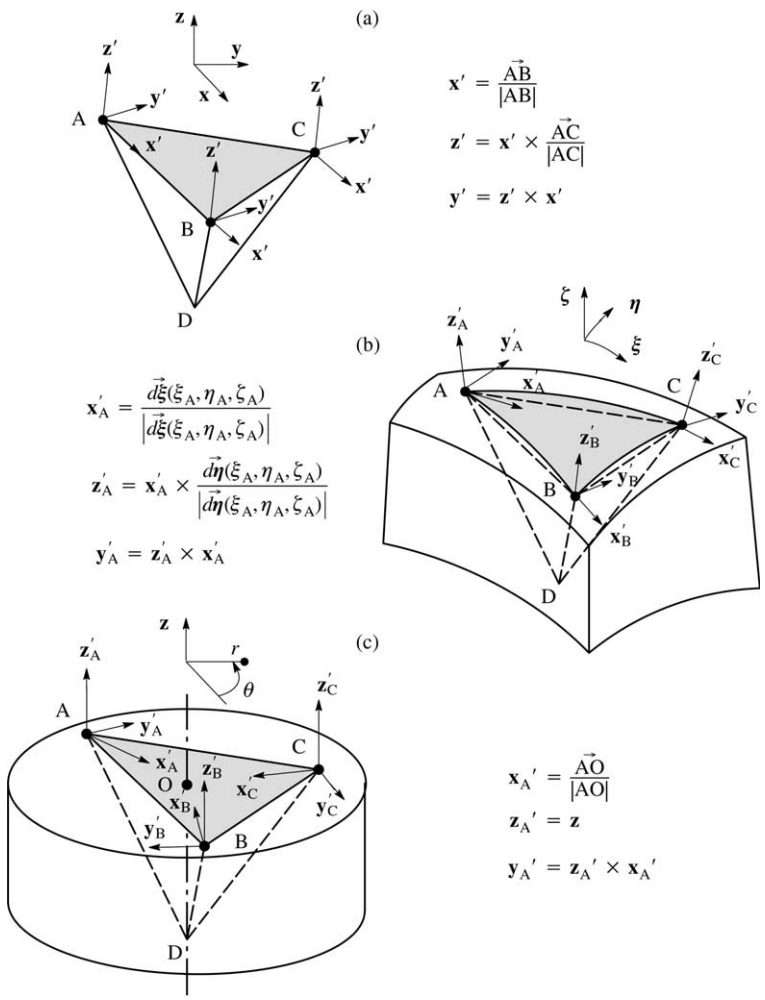


Fig. 14. Local coordinate systems using (a) topology of a single element, (b) topology of subdomain and (c) prescribed formula.

limitations, some of which we will now briefly discuss. Since our numerical formulation is restricted to linear finite elements with linear geometry, the need for a local coordinate system arises only when stress transformations like Eq. (10) are necessary.

Using option 1, we can generate a coordinate system which has one axis normal to the element face and the other axes tangential to it. This arrangement, shown in Fig. 14(a), does not provide much control over the orientation of the tangential components because of the lack of information that can be extracted from the element topology data (apart, for example, from directing one of the tangential axes along one of the edges). A further limitation of this scheme is that the orientation of the local axes is constant over the element surface. These characteristics are a major disadvantage of option 1 when a local coordinate system is required to apply surface tractions along a boundary. Apart from the fact that the scheme cannot specify the tangential tractions properly, it restricts the orientation of the normal traction to being piecewise constant. This limitation can result in weak lower bounds for problems where the loading is applied over curved boundaries. For some tasks, however, such as generating discontinuity constraints, option 1 is sufficiently flexible and provides a simple coordinate system with a minimum of data.

When applied to a parametrically mappable subdomain geometry, option 2 can be used to establish a local coordinate system whose axes are related to the directions of the curvilinear coordinate system at the point of interest. This type of arrangement is useful for applying surface tractions which are normal or tangential to a parametrically approximated boundary. Since the orientation of the coordinate system changes from point to point on the surface, it permits a piecewise linear representation of the tractions over a curved boundary (Fig. 15(b)). This feature means that the tractions can be modelled with good accuracy, even though the geometry of the elements is restricted to being linear. Moreover, since the tangential directions to the surface can be properly oriented with respect to the curvilinear coordinates on the boundary, tangential tractions can be modelled correctly. This type of local coordinate system is also applicable when centrifugal body forces are imposed on a body about an axis of symmetry.

Although it is both effective and general, option 2 is numerically complicated and needs more input information than the other options.

The third and last option is useful when the shape of the domain, or the profile of the applied tractions, can be given in analytical form. In modelling the rotation of the object, for example, the direction of the centrifugal forces at a point can be obtained using the analytical relations between rectangular and polar coordinates, as shown in Fig. 14(c).

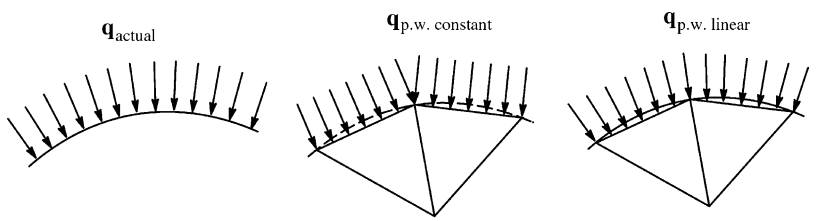
In order to exploit the relative advantages of each of the above options, the following approach is used in the lower bound implementation:

- Use option 1 for interelement discontinuity constraints, where the tangential components of the stress tensor can be oriented in arbitrary, mutually orthogonal directions.
- Use option 2 whenever locally oriented surface tractions, especially tangential surface tractions, need to be applied.
- Use option 3 if the orientation of the applied forces is independent of the geometry of the body (e.g. centrifugal body forces).

This strategy for selecting the local coordinate system has proved to be both effective and accurate for modelling a wide range of practical problems with complex loading.

6. Generation of extension elements

When the lower bound finite element method is applied to problems with semi-infinite domains, only part of the body is discretised. This means that the optimised stress field does not necessarily satisfy equilibrium, the stress boundary conditions and the yield criterion throughout the entire domain and, therefore, cannot be used to infer a rigorous lower bound on the collapse load. Although this type of solution, which is known as a partial stress field, may actually furnish a good estimate of the true collapse load, a fully rigorous lower bound can be obtained only by



Actual tractions acting on curved boundary.

(a) Piece-wise constant approximation of traction orientation.

(b) Piece-wise linear approximation of traction orientation.

Fig. 15. Approximation of applied surface tractions on curved boundary.

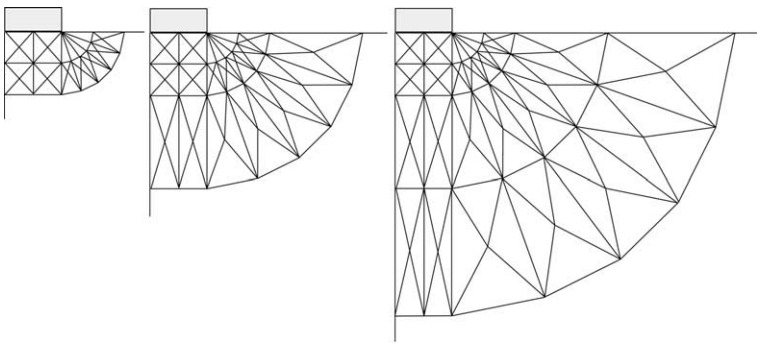


Fig. 16. Extending the stress field by extending the mesh.

extending the stress field over the semi-infinite domain in such a way that all the conditions of the lower bound theorem are fulfilled. This process is often difficult, especially for cases involving irregular boundary shapes, and is frequently omitted in analytical calculations.

To resolve this situation two different approaches may be adopted. In the first approach, a guess is made of the extent of the plastic zone and an initial mesh is chosen to cover this volume. Once the collapse load has been computed, the initial grid is refined by adding an additional layer of elements around its periphery, as shown in Fig. 16, and a new collapse load is found. This process is repeated until the collapse load ceases to change as the mesh penetrates more deeply into the semi-infinite body. Although it does not guarantee that the result obtained is a rigorous lower bound, this 'engineering' approach does give good results in practice. Moreover, it can be used for any type of yield criterion and, provided an automatic mesh generator is available, is not too difficult to apply.

The second approach, which is more complicated than the first, is to develop special extension elements which are deployed around the periphery of the mesh (Fig. 17). These are constructed so they extend the stress field beyond the limits of the grid in such a way that it is statically admissible. To handle an arbitrary geometry in two dimensions, Pastor [4] showed that a maximum of two different types of extension elements are necessary. Here it will be shown that for D-dimensional geometries, a maximum of D different types of extension elements are required. Although they are restricted to certain types of yield criteria, extension elements are attractive because they guarantee that the solution obtained is a rigorous lower bound. Moreover, they remove the need for the trial-and-error process of mesh enlargement described in the previous paragraph.

A D-dimensional extension element is much like a regular lower bound finite element in that the stress field is defined by the stresses at $D+1$ nodes and the body forces are assumed to be constant. Indeed, as with any lower bound element, the stresses must satisfy the equilibrium, stress boundary and yield conditions. Consider the 2D case shown in Fig. 18, where a linear expansion is used to model the stresses across and outside a three-noded extension element. Provided the equilibrium and stress boundary conditions are satisfied within the triangle, then they are automatically satisfied for any point p outside the triangle. This implies that all extension elements are subject to the same equilibrium and stress boundary constraints as regular elements. For the yield condition, however, it is quite clear that additional constraints are necessary, since imposing the condition that $f(\boldsymbol{\sigma}^l) \leq 0$ for the nodes $l = 1, 2, 3$ does not guarantee that $f(\boldsymbol{\sigma}^p) \leq 0$ for any point p outside the element. Deriving appropriate yield constraints for an extension element is the most complicated part of the extension procedure, as special care needs to be taken to ensure the stress field is properly constrained. For the sake of simplicity, our attention will now be limited to those yield surfaces which have no meridional curvature in principal stress space (such as the Tresca, Von Mises, Drucker-Prager and Mohr-Coulomb criteria), as these permit the formulation of elegant and compact extension conditions. It is possible to derive extension conditions for more complex yield criteria with curved meridional sections, but these will not be covered here.

To derive the yield conditions for a two-dimensional extension element, consider σ_{ij} stress space with the yield surface $f(\boldsymbol{\sigma}) \leq 0$ and three nodal stress points $\boldsymbol{\sigma}^1, \boldsymbol{\sigma}^2, \boldsymbol{\sigma}^3$ with coordinates $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3$. Physical reasoning suggests that, once a linear variation of the stresses is assumed between the nodes, then there are only two possible cases to consider when extending the stress field beyond the boundaries of the element. These cases involve either uni- or bi-directional extension and are shown, respectively, in

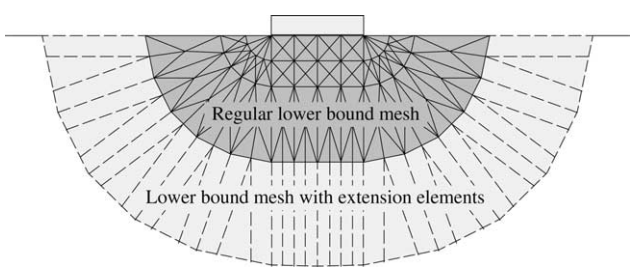


Fig. 17. Extending the stress field using special extension elements.

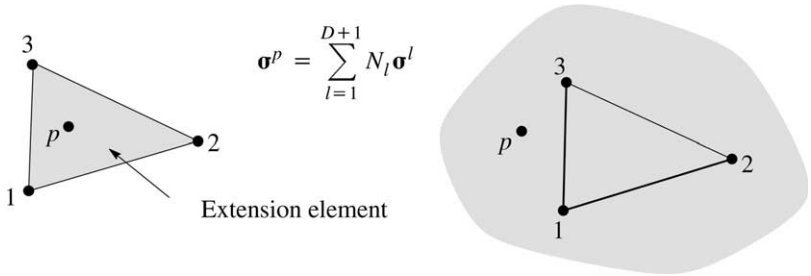


Fig. 18. Approximation of stress field inside and outside the extension element.

Fig. 19(a) and (b). In the first case, extension is possible along the direction $\mathbf{X}^1 - \mathbf{X}^2$ if we apply the yield condition

$$f(\boldsymbol{\sigma}^1 - \boldsymbol{\sigma}^2) \leq 0 \tag{11}$$

to the stresses at node 1. To aid in the description of the extension process, all nodes with this type of modified yield condition are called extension nodes. For a cone-like yield surface, the extension node, node 1, is always positioned further from the apex of the cone $f(\boldsymbol{\sigma}) \leq 0$ than its partner, node 2, and is inside the cone given by the surface $f(\boldsymbol{\sigma} - \boldsymbol{\sigma}^2) \leq 0$. To signify their different roles in the extension process, node 2 is said to be an active node and node 3 is said to be a passive node. Both of these nodes are subject to the same type of yield constraint, so that $f(\boldsymbol{\sigma}^2) \leq 0$ and $f(\boldsymbol{\sigma}^3) \leq 0$, but node 2 is also involved in the constraint (11) which is enforced at node 1. The fourth type of node shown in Fig. 19(a), node 4, is known as a dummy node. The stresses at this node, $\boldsymbol{\sigma}^4$, are not independent, since they can be expressed as a linear combination of the stresses at nodes 1–3. The sole purpose for including node 4 is to accommodate a semi-infinite stress discontinuity along the edge defined by nodes 3 and 4. Indeed, this dummy node is subject to no constraints apart from those that are imposed by discontinuity equilibrium. With reference to Fig. 19(a), the uni-directional extension conditions are summarised by the constraints $f(\boldsymbol{\sigma}^1 - \boldsymbol{\sigma}^2) \leq 0$, $f(\boldsymbol{\sigma}^2) \leq 0$ and $f(\boldsymbol{\sigma}^3) \leq 0$, together with the usual equilibrium and stress boundary constraints that apply to the element defined by nodes $(1, 2, 3)$. These conditions guarantee that any stress point in the semi-infinite area bounded by the vectors

$$\begin{matrix} \alpha(\mathbf{X}^1 - \mathbf{X}^2) \\ \mathbf{X}^2 - \mathbf{X}^3 \qquad \alpha \in (0, \infty) \\ \alpha(\mathbf{X}^4 - \mathbf{X}^3) \end{matrix} \tag{12}$$

will automatically meet the requirements of a statically admissible stress field. The zone of the statically admissible stress field extension is thus the shaded area.

To construct a bi-directional extension element, as shown in Fig. 19(b), we apply the constraint (11) at node 1 and the constraint

$$f(\boldsymbol{\sigma}^3 - \boldsymbol{\sigma}^2) \leq 0 \tag{13}$$

at node 3. Hence the bi-directional extension conditions are $f(\boldsymbol{\sigma}^1 - \boldsymbol{\sigma}^2) \leq 0$, $f(\boldsymbol{\sigma}^2) \leq 0$ and $f(\boldsymbol{\sigma}^3 - \boldsymbol{\sigma}^2) \leq 0$, coupled with the standard equilibrium and stress boundary constraints that apply to the element defined by nodes $(1, 2, 3)$. This gives a semi-infinite statically admissible extension zone which is bounded by the vectors

$$\begin{matrix} \alpha(\mathbf{X}^1 - \mathbf{X}^2) \\ \alpha \in (0, \infty) \\ \alpha(\mathbf{X}^3 - \mathbf{X}^2) \end{matrix} \tag{14}$$

The logic described above for 2D geometries can be generalised to D-dimensions. In this case there are D

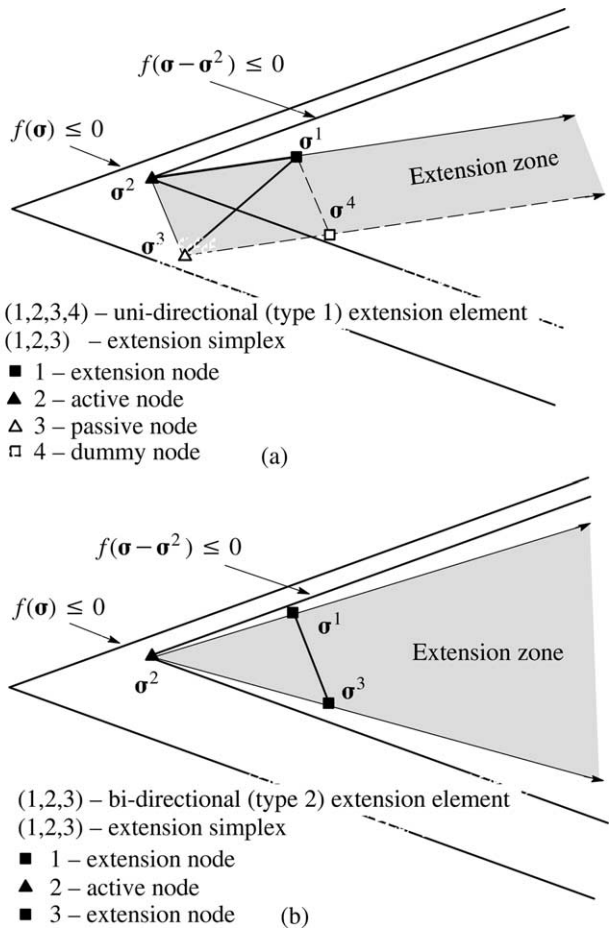
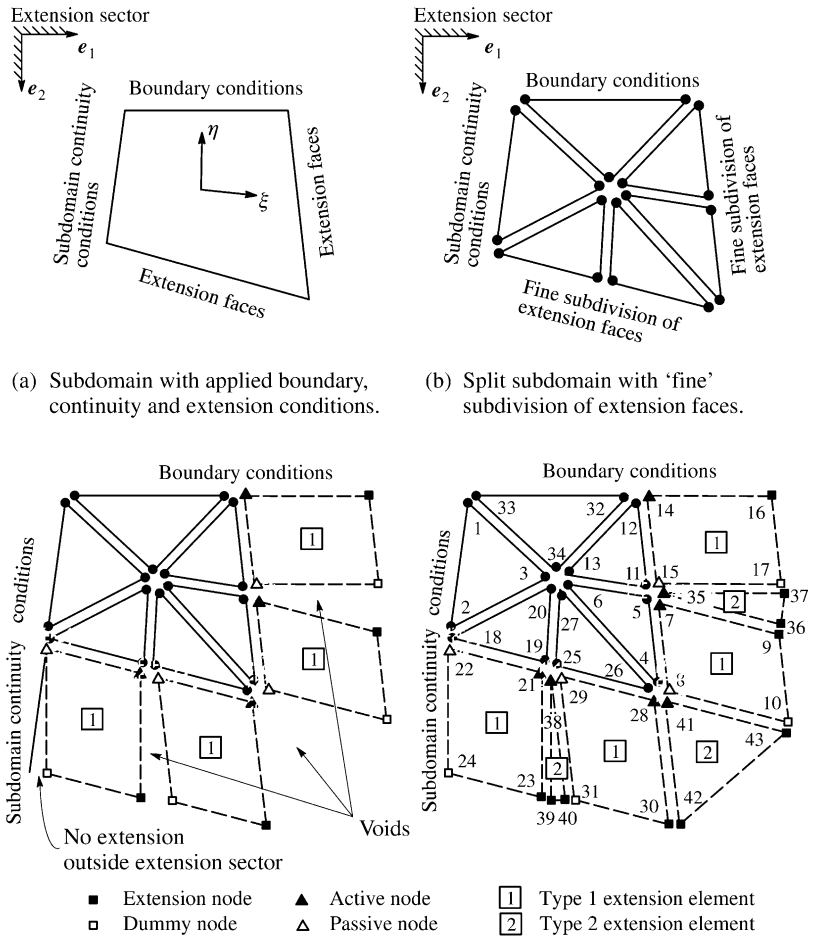


Fig. 19. Extension elements for 2D (a) uni-directional (type 1) and (b) bi-directional (type 2).

different types of extension elements, each of which is distinguished by its number of extension nodes. Each extension node, together with its corresponding active node, determines the direction of the extension. In the case of M multiple extension directions, statically admissible extension of the stress field is possible throughout the semi-infinite M-dimensional volume they define. In D-dimensions, it is convenient to label an extension element as being of type j if it has j extension nodes. For such an element, there are $D - j$ passive nodes, one active node and $2D - (j + 1)(D - j + 1)$ dummy nodes. In the latter, the stresses are not independent and can be expressed as linear combinations of the stresses at the $D + 1$ nodes of the extension simplex. Dummy nodes appear only in extension elements of types up to $D - 1$ and are needed solely for the purpose of enforcing interelement discontinuity equilibrium.

Using the range of extension elements defined above it is possible to generate a statically admissible stress field, and thus obtain a rigorous lower bound on the true collapse load, for any semi-infinite domain. Even though a solution using extension elements is always a strict lower bound, it may considerably underestimate the true collapse value if the regular finite element mesh does not cover the plastic region. This implies that the solutions for all semi-infinite problems should be computed at least twice, once with extension elements and once without them, in order to verify their quality. If these two solutions differ significantly, then the regular mesh should be extended more deeply into the body and the process repeated.

As the theoretical aspects of extending the admissible stress field in a semi-infinite domain have now been discussed, we can proceed with a description of the scheme for constructing the extension elements in a dimensionally independent manner. Using the terminology described earlier, the process starts by generating 'type 1' extension elements on all sides of the current subdomain which are of semi-infinite extent. These sides are part of the domain's



(a) Subdomain with applied boundary, continuity and extension conditions.

(b) Split subdomain with 'fine' subdivision of extension faces.

(c) Mesh after type 1 extension elements have been generated.

(d) Final mesh with type 1 and 2 extension elements and actual node numbering.

Fig. 20. Generation of extension elements in 2D.

exterior boundary and, as shown in Fig. 20(a), are not subject to boundary, loading or subdomain continuity conditions. Because each type 1 element forms a parallelogram, all extension faces need to be split using the 'fine' subdivision process prior to adding these elements. The process for splitting the extension faces is identical to that described in Section 4 and is shown in Fig. 20(b). When forming the type 1 extension elements, each side is extended along the direction of one of the corresponding subdomain edges or, if this lies outside the extension sector, the nearest permissible extension direction (Fig. 20(c)). The boundary nodes on the sides of the newly created extension elements, where appropriate, inherit the boundary conditions from the nearest node on the corresponding side of the parent subdomain. Once all the type 1 extension elements have been generated, voids may still remain in the stress field. These are remedied automatically by looking for gaps between all type 1 extension elements and filling any voids with type i extension elements. During this stage, we use the topology and boundary condition information from the extension elements of type $i-1$, and the index i ranges from 2 to D (Fig. 20(d)). The steps involved in generating extension elements for the general 3D case are shown in Fig. 21.

When generating the extension elements, it is convenient to arrange the node numbers so that the determinant formed by the first $D+1$ lots of nodal coordinates is positive (thus signifying a positive volume). The numbering system shown in Fig. 22 is convenient for controlling the element topology and allows simple logic to be used for the level-by-level void filling procedure described above. In our generation scheme, we can skip consideration of the boundary conditions as these are inherited completely from the nearest regular node. When imposing the discontinuity conditions, option 1 of Section 5 is employed to generate the local coordinate system using the first D nodes on the appropriate face of the extension element.

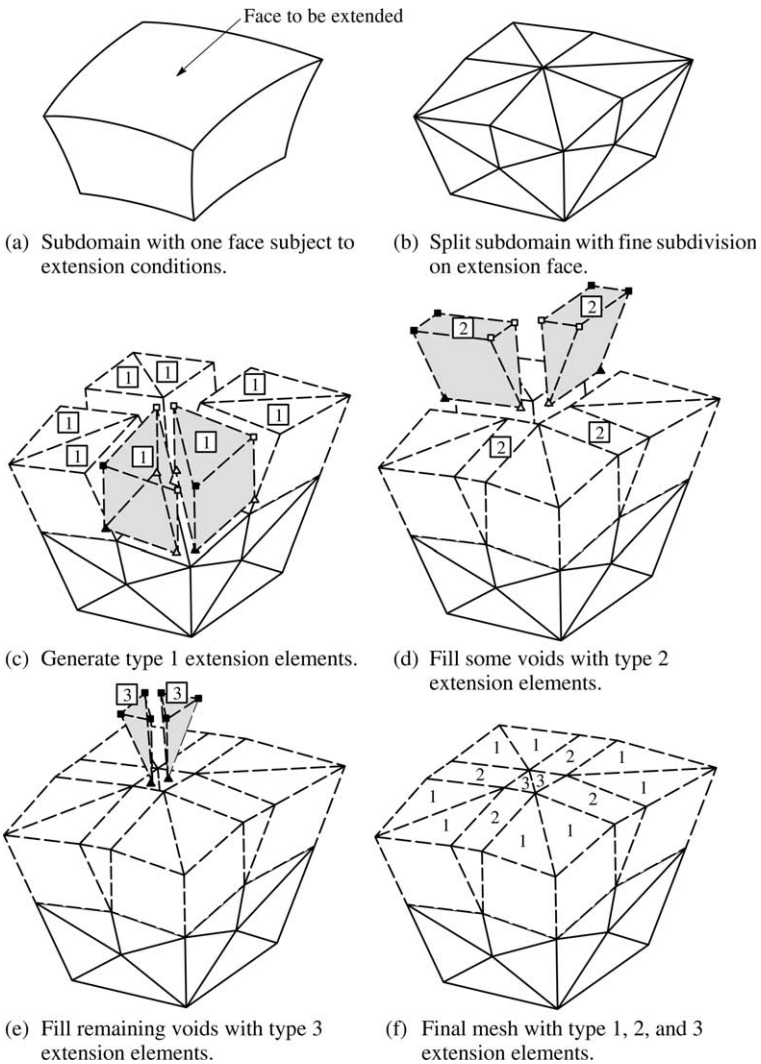


Fig. 21. Generation of extension elements in 3D.

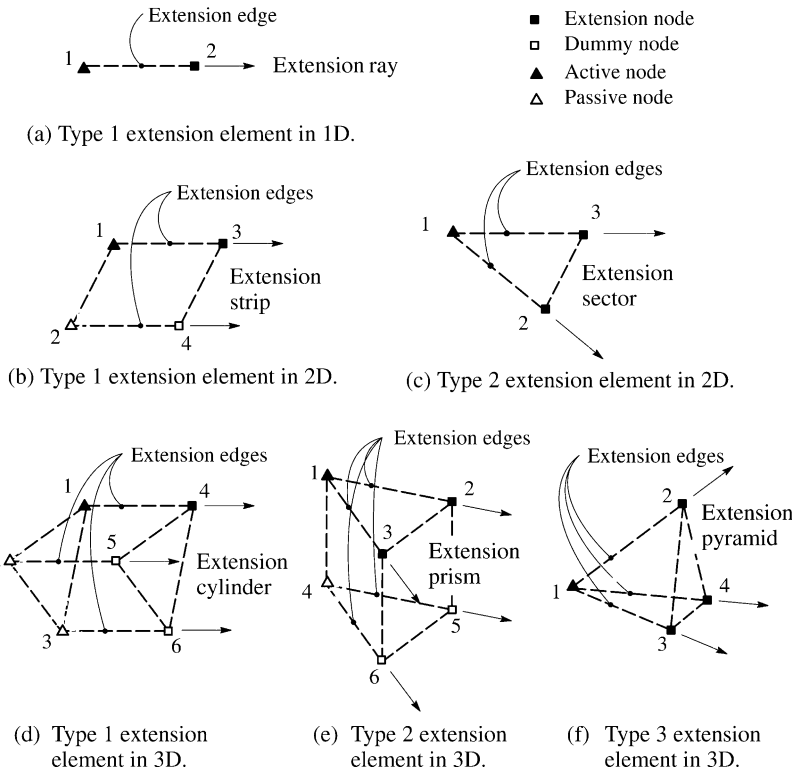


Fig. 22. Extension element numbering scheme: (a) one-dimension, (b) and (c) two dimensions, and (d), (e) and (f) three dimensions.

7. Conclusions

A general strategy for generating lower bound finite element meshes in *D*-dimensions has been described. The procedure is based on a parametric mapping technique, coupled with midpoint splitting of subdomains, and permits the user to control the distribution of the discontinuities and elements precisely. Although it is not fully automatic, the algorithm is fast, generates extension elements, and can be used for one-, two- and three-dimensional geometries.

References

[1] Lyamin AV. Three-dimensional lower bound limit analysis using nonlinear programming. PhD Thesis. Department of Civil, Surveying and Environmental Engineering, University of Newcastle, Australia; 1999

[2] Lyamin AV, Sloan SW. Lower bound limit analysis using nonlinear programming. Int J Numer Meth Engng 2002;55:573–611.

[3] Chen WF. Limit analysis and soil plasticity. Amsterdam: Elsevier; 1975.

[4] Pastor J. Limit analysis: numerical determination of complete statical solutions: application to the vertical cut. J Méch Appl 1978;2:167–96. in French.

[5] Price MA, Armstrong CG. Hexahedral mesh generation by medial surface subdivision. Part I. Solids with convex edges. Int J Numer Meth Engng 1995;38(19):3335–59.

[6] Price MA, Armstrong CG. Hexahedral mesh generation by medial surface subdivision. Part II. Solids with flat and concave edges. Int J Numer Meth Engng 1997;40(1):111–36.

[7] Steinmueller G. Restrictions in the application of automatic mesh generation schemes by isoparametric coordinates. Int J Numer Meth Engng 1974;8:289–94.

[8] Subramanian G, Prasanth A, Raveendra VVS. Algorithm for two- and three-dimensional automatic structured mesh generation. Comput Struct 1996;61(3):471–7.

[9] Zienkiewicz OC, Phillips DV. An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. Int J Numer Meth Engng 1971;3(4):519–28.

[10] Zienkiewicz OC, Taylor RL. The finite element method, 4th ed. New York: McGraw-Hill; 1989.