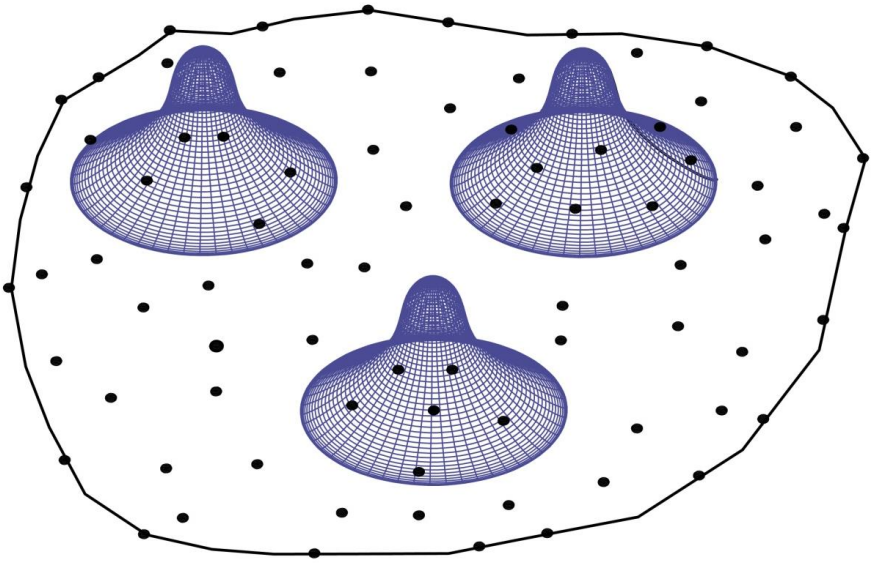
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΣΤΑΤΙΚΗΣ & ΑΝΤΙΣΕΙΣΜΙΚΩΝ ΕΡΕΥΝΩΝ

ΑΝΑΛΥΣΗ ΦΟΡΕΩΝ ΜΕ ΜΗ ΠΛΕΓΜΑΤΙΚΕΣ ΜΕΘΟΔΟΥΣ ΤΥΠΟΥ GALERKIN



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ
ΝΙΚΟΛΑΟΥ ΛΑΝΤΖΟΥΝΗ

ΕΠΙΒΛΕΠΩΝ:
ΜΑΝΟΛΗΣ ΠΑΠΑΔΡΑΚΑΚΗΣ
ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π

ΑΘΗΝΑ
ΟΚΤΩΒΡΙΟΣ 2013

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΣΤΑΤΙΚΗΣ & ΑΝΤΙΣΕΙΣΜΙΚΩΝ ΕΡΕΥΝΩΝ

ΑΝΑΛΥΣΗ ΦΟΡΕΩΝ ΜΕ ΜΗ ΠΛΕΓΜΑΤΙΚΕΣ ΜΕΘΟΔΟΥΣ ΤΥΠΟΥ GALERKIN

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ
ΝΙΚΟΛΑΟΥ ΛΑΝΤΖΟΥΝΗ

ΕΠΙΒΛΕΠΩΝ:
ΜΑΝΟΛΗΣ ΠΑΠΑΔΡΑΚΑΚΗΣ
ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π

ΑΘΗΝΑ
ΟΚΤΩΒΡΙΟΣ 2013

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF CIVIL ENGINEERING
INSTITUTE OF STRUCTURAL ANALYSIS & ANTI-SEISMIC RESEARCH

NUMERICAL SIMULATION OF STRUCTURES USING THE ELEMENT-FREE GALERKIN METHOD

DIPLOMA THESIS

NIKOLAOS LANTZOUNIS

SUPERVISOR:
MANOLIS PAPADRAKAKIS
PROFESSOR, N.T.U.A

ATHENS
OCTOBER 2013

Στους γονείς μου Βάσω και Βασίλη,
που με υπομονή μου έδωσαν τα πρώτα μου εφόδια
για να ξεκινήσω το ταξίδι μου στην γνώση

Ευχαριστίες

Η παρούσα διπλωματική εργασία, αποτέλεσμα πολύμηνης προσπάθειας, έρχεται να σηματοδοτήσει το τέλος των προπτυχιακών μου σπουδών στη Σχολή Πολιτικών Μηχανικών του Ε.Μ.Π. Στο διάστημα αυτό κατακτήθηκαν πολλοί και διαφορετικοί στόχοι, οι οποίοι σίγουρα δεν θα μπορούσαν να γίνουν πράξη χωρίς την συμβολή, καθοδήγηση και υποστήριξη ορισμένων ανθρώπων που βρέθηκαν γύρω μου.

Αρχικά θα ήθελα να ευχαριστήσω τον Καθηγητή και επιβλέποντα της εν λόγω εργασίας κ. Μανόλη Παπαδρακάκη. Η πολυετής πείρα του πάνω σε θέματα Υπολογιστική Μηχανικής, αλλά και η διάθεση να αφιερώσει πολλές ώρες, παρά το απαιτητικό του πρόγραμμα, με βοήθησαν να ξεπεράσω τις όποιες δυσκολίες παρουσιάστηκαν κατά την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας. Εμπνευση και όρεξη για δουλειά μου δημιούργησε όμως και η συνολική του στάση απέναντι στα ερευνητικά ζητήματα, αλλά και στην σχολή γενικότερα. Ανήκει πιστεύω σε μια λαμπρή μερίδα διδασκόντων, η οποία, παρόλα τα προβλήματα που αντιμετωπίζει το ελληνικό πανεπιστήμιο, συνεχίζει να παρέχει ευκαιρίες, όραμα και ελπίδα για κάτι καλύτερο, και κυρίως ήθος στους νέους ανθρώπους. Το πάθος του για το λειτούργημα του, αλλά και ο ιδιάζων τρόπος σκέψης του, επηρέασαν τα επιστημονικά μου ενδιαφέροντα, αλλά και τον τρόπο που τα προσεγγίζω, και αυτό ίσως είναι το μεγαλύτερο όφελος που έχω αποκομίσει από την συνεργασία μαζί του τα τελευταία χρόνια.

Οφείλω επίσης ένα μεγάλο ευχαριστώ στον υποψήφιο Διδάκτορα, και πλέον καλό μου φίλο, Παναγιώτη Μέτση, μέλος της ερευνητικής ομάδας του κ. Παπαδρακάκη. Η ορθή εξέλιξη και η ολοκλήρωση αυτής της εργασίας δεν θα ήταν δυνατή χωρίς το μεγάλο εύρος θεωρητικών γνώσεων που μου προσέφερε. Σπάζοντας την εσωστρέφεια, που πολλές φορές κατατρύχει την επιστημονική κοινότητα, κατάφερε να δομήσει μαζί μου ένα ιδανικό πλαίσιο συνεργασίας, μέσα από το οποίο προέκυψαν καινοτόμες ιδέες και πρακτικές. Σημαντικές επίσης ήταν οι παρατηρήσεις του, από την αρχή της σύνταξης του κώδικα σε γλώσσα προγραμματισμού Matlab για τις ανάγκες της διπλωματικής αυτής εργασίας, μέχρι και την τελική διαμόρφωση του παρόντος κειμένου. Θέλω να ευχαριστήσω τέλος τον κοντινό μου φίλο Δημοσθένη Τσακνιά, για τις γλωσσικές διορθώσεις επί του κειμένου της εργασίας.

Πέραν της βοήθειας κατά την συγγραφή της διπλωματικής εργασίας, θα ήθελα να σταθώ και στα άτομα που με στήριξαν με την εμπιστοσύνη τους αυτά τα χρόνια, και με τα οποία μοιράστηκα την καθημερινότητά μου στη σχολή. Ευχαριστώ τον Αλέξανδρο, τον Λουκά, την Τόνια, την Ανδριάνα, τον Παναγιώτη, τον Βαγγέλη για την συμπαράστασή τους στην κοινή πορεία μας. Ευχαριστίες οφείλω και στους καλούς μου φίλους από την συντονιστική ομάδα του forum της σχολής www.mqn.gr, για τις όμορφες στιγμές που ζήσαμε. Η πολυετής συνδιαμόρφωση και συνδιαχείριση του site -καρπός ενός ομαδικού κόπου-, με έμαθε να μοιράζομαι γνώσεις και εμπειρίες με στόχο την συλλογική αλλά και την ατομική πρόοδο.

Ολοκληρώνοντας θα ήθελα να ευχαριστήσω θερμά τους γονείς μου, Βάσω και Βασίλη, και την αδερφή μου Παναγιώτα για την ηθική υποστήριξη, την αγάπη και την υπομονή τους, όλα αυτά τα χρόνια.

Νίκος Λαντζούνης
Οκτώβριος 2013

Contents

Περίληψη

Στις μέρες μας, θέματα που σχετίζονται με την μοντελοποίηση και την προσομοίωση, διαδραματίζουν ένα σημαντικό ρόλο στην οικοδόμηση και την εξέλιξη των υπολογιστικών συστημάτων στον τομέα του μηχανικού. Η μέθοδος των πεπερασμένων στοιχείων υπήρξε η κυρίαρχη μέθοδος της υπολογιστικής μηχανικής κατά τις τελευταίες δεκαετίες, ενώ σημαντική παραμένει η συνεισφορά της στην επιστήμη, την τεχνολογία και την βιομηχανία. Παρόλα αυτά, και στην μέθοδο των πεπερασμένων στοιχείων υπάρχουν περιορισμοί, οι οποίοι μπορούν να γίνουν ορατοί κατά την επίλυση προβλημάτων μεγάλων μετατοπίσεων, ή κατά την προσομοίωση ασυνεχειών, όπως η διάδοση τυχούσας ρωγμής σε ένα συνεχές μέσο. Συνήθως τέτοια προβλήματα αντιμετωπίζονται από την μέθοδο των πεπερασμένων στοιχείων, με την πύκνωση του δικτύου στις συγκεκριμένες περιοχές του φορέα, μία χρονοβόρα διαδικασία, που θα πρέπει να εκτελείται με ιδιαίτερη προσοχή, ώστε να διατηρηθεί η καλή κατάσταση του αρχικού πλέγματος. Παρόλα αυτά, μερικές φορές φαντάζει πολύ δύσκολο έως αδύνατο, να ξεπεραστούν αυτές οι δυσκολίες που προκύπτουν από την παρουσία δικτύου.

Ετσι λοιπόν η ανάπτυξη μη πλεγματικών μεθόδων προσομοίωσης, έχει προσελκύσει το ενδιαφέρον μεγάλου μέρους της επιστημονικής κοινότητας τα τελευταία χρόνια. Οι μη πλεγματικές μέθοδοι διακριτοποιούν ένα συνεχές μέσο χρησιμοποιώντας έναν πεπερασμένο αριθμό σημείων, ενώ η προσέγγιση που προκύπτει βασίζεται εξ ολοκλήρου στα σημεία αυτά. Κατά την εφαρμογή των μεθόδων αυτών, δεν υπάρχει ανάγκη δημιουργίας δικτύου, ούτε η παρουσία στοιχείων κατά την προσομοίωση ενός φορέα. Μέχρι στιγμής μεγάλη πρόοδος έχει συντελεστεί στον τομέα της ανάπτυξης μη πλεγματικών μεθόδων, ενώ παράλληλα πολλές πτυχές τους επιδέχονται βελτίωσης. Είναι προφανές ότι οι παραπάνω μέθοδοι προσφέρουν ένα ευρύ φάσμα εφαρμογών, καθώς όπως φαίνεται μπορούν να παρέχουν λύσεις σε πολλά σύγχρονα υπολογιστικά προβλήματα.

Στην παρούσα διπλωματική εργασία, παρουσιάζεται μια χαρακτηριστική μη πλεγματική μέθοδος, η μέθοδος Element Free Galerkin (EFG). Το θεωρητικό υπόβαθρο της μεθόδου αναλύεται εκτενώς, ενώ ταυτόχρονα παρέχεται μια λεπτομερής διατύπωση των εξισώσεων που διέπουν την μέθοδο αυτή. Ειδική αναφορά γίνεται και σε θέματα όπως η αριθμητική ολοκλήρωση της εξίσωσης κίνησης και ο τρόπος επιβολής των συνοριακών συνθηκών. Για τις ανάγκες της παρούσας διπλωματικής εργασίας αναπτύχθηκε κώδικας στην γλώσσα προγραμματισμού Matlab. Για να διερευνηθούν ζητήματα όπως η σύγκλιση της μεθόδου και η αποτελεσματικότητα στην εφαρμογή της, επιλέχθηκε ένας τυπικός επίπεδος πρόβολος, σαν πρόβλημα αναφοράς.

Στο έκτο και έβδομο κεφάλαιο αυτής της διατριβής προτείνονται κάποιες ιεραρχικές ιδέες για την μέθοδο EFG. Κάνοντας χρήση αυτών των εννοιών, εισάγεται μια ιεραρχική μέθοδος προσαρμογής η οποία προτείνεται για την βελτίωση

υπολογισθέντων λύσεων και την μετεπεξεργασία αποτελεσμάτων που έχουν προκύψει από την εφαρμογή της μεθόδου EFG. Η μέθοδος EFG παρουσιάστηκε πιο χρονοβόρα σε σύγκριση με την μέθοδο των πεπερασμένων στοιχείων. Μια τέτοια μέθοδος προσαρμογής, λοιπόν, υπόσχεται να δώσει βελτιωμένα αποτελέσματα με χαμηλό υπολογιστικό κόστος. Προκειμένου να διευκρινιστεί αυτή η τεχνική, παρουσιάζεται το θεωρητικό της υπόβαθρο, ενώ επιλύονται και δύο προβλήματα αναφοράς, ο επίπεδος πρόβολος και ένα δισδιάστατο πρόβλημα μορφής L. Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την χρήση της μεθόδου EFG και την ιεραρχική μέθοδο προσαρμογής, ενώ δίνονται και προτάσεις για μελλοντική έρευνα.

Abstract

Topics related to modeling and simulation, play an important role in building and advancing engineering systems, in rapid and cost-effective ways. The Finite Element Method (FEM) has been the dominant technique in computational mechanics in the past decades, and it has made significant contributions to science, engineering and industry. Nevertheless, FEM limitations do exist, when it is applied to large deformations of materials or encountering discontinuities such as a crack propagation along arbitrary and complex paths. Usually FEM faces such problems with remeshing in certain areas of a problem domain, a time consuming procedure, that needs to be handled with special care, in order to maintain the good condition of the initial mesh. Sometimes though, it seems impossible to completely overcome those mesh-related difficulties by a mesh-based method.

Because of those inherent limitations of FEM, mesh free methods started becoming of interest for a broad community of researchers only several years ago. Mesh free methods discretize the continuum body with only a set of nodal points, and the approximation is constructed entirely in terms of nodes. There is no need of mesh or elements in this method. Whereas, great advances of mesh free methods have been achieved, many aspects are still to be further explored and improved, making mesh free methods an interesting research area, with many potential solutions for challenging computational problems.

In this thesis, a typical mesh free method, the Element Free Galerkin Method (EFG) is proposed. The theoretical background of the method is presented while detailed formulation and equations are provided. Technical issues, especially issues related to background integration and boundary conditions imposition, are also examined. Note that the Moving Least Square approximation is applied to form the shape functions of the EFG method, while the constrained Galerkin approach is used to enforce the essential boundary conditions. For the needs of this thesis supporting Matlab code was developed. A typical benchmark problem of a cantilever beam is considered, to illustrate issues such as the rate of convergence and implementation effectiveness.

In the sixth and seventh chapter of the thesis, we propose some hierarchical concepts for the EFG method. Using these concepts, a hierarchical refinement is introduced, which offers an *a posteriori* treatment for the final results of an already calculated problem domain. The EFG method is considered to be a more time consuming process, compared to the FEM. Such a refinement promises to give better results at a lower computational cost. In order to clarify this technique the theoretical background is presented, as well as two typical benchmark problems, a cantilever beam and a L-shaped domain, are solved. Finally, in the last chapter the conclusions drawn from the use of the EFG method, and the hierarchical refinement process are presented, with suggestions for future research.

1 Introduction

1.1 Definition

Designing advanced engineering systems requires the use of computer-aided design (CAD) tools. In such tools, computational simulation techniques are often used to model and investigate physical phenomena in an engineering system. The simulation requires solving the complex differential or partial differential equations that govern these phenomena.
Traditionally, such complex partial differential equations are largely solved using numerical methods, such as the finite element method (FEM) and the finite difference method (FDM). In these methods, the spatial domain where the partial differential governing equations are defined, is often discretized into meshes.
A mesh is defined as any of the open spaces or interstices between the strands of a net that is formed by connecting nodes in a predefined manner. The terminologies of grids, volumes, cells, and elements carry certain physical meanings as they are defined for different physical problems. Given the definition of a mesh, all the aforementioned can be classified as a mesh. The key point here is that a mesh must be predefined to provide a certain relationship between the nodes, which is the base of the formulation of these conventional numerical methods.
After constructing this required mesh and by applying a proper principle governing the problem, complex differential or partial differential equations can be approximated by a set of algebraic equations concerning the mesh.
The finite element method (FEM) has been used with great success in multiple fields with applications in both academia and industry. However one should have in mind that limitations do exist. Due to this mesh-based interpolation referred, distorted or low quality mesh can lead to higher errors, necessitating remeshing , which is a time and human labour consuming task, while it is not always feasible for three-dimensional geometries.
The mesh free method, described in this diploma thesis, is used to establish a system of algebraic equations for the whole problem domain without the use of a predefined mesh. This method uses a set of nodes scattered within the problem domain, as well as a set of nodes scattered on the boundaries of the domain to *represent* (not discretize) the problem domain and its boundaries. These sets of scattered nodes, are sometimes positioned in a quite structured way, as proposed in many references, which does not necessarily mean that the problem has no solution if the nodes were positioned randomly within the problem domain. These distributed nodes, do not form a mesh, which means that no information on the relationship between the nodes is required, at least for field variable interpolation.

There are a number of mesh free methods, such as the element free Galerkin (EFG) method (Belytschko et al., 1994b), the meshless local Petrov–Galerkin (MLPG) method (Atluri and Zhu, 1998), the point interpolation method (PIM) (Liu, G. R. and Gu, 1999), the point assembly method (PAM) (Liu, G. R., 1999), the finite point method (Onate et al., 1996), the finite difference method with arbitrary irregular grids (Liszka and Orkisz, 1980; Jensen, 1980), smooth particle hydrodynamics (SPH) (Lucy, 1977; Gingold and Monaghan, 1977), reproducing kernel particle method (Liu, W. K. et al., 1993), which is an improved version of SPH, and so on. They all share the same feature that predefined meshes are not used, at least for field variable interpolation.
Although these methods involve less time in the preparation of input data they are not really ideal, and may fail in one of the following categories:

- Some of these methods require a set of background cells (Gauss cells) for the integration of system matrices derived from the weak form over the problem domain. EFG method used in this diploma thesis belong in this category. These methods are of practical use in many ways, as the creation of a background mesh is generally feasible and a quite rapid procedure that can be easily automated. Despite the fact that they do not rely on a mesh, these kinds of meshless methods require a preliminary step for the identification of the correlation between nodes and background Gauss points before building the stiffness matrix. This is implicitly performed with the mesh generation in FEM, but must be explicitly done in EFG method and can be time-consuming.
- Furthermore, the resulting matrices from the application of mesh free methods are more densely populated and the computational cost for the formulation and solution of the problem is much higher than the conventional FEM. This is mainly attributed to the vast increase in interactions between nodes and integration points due to their extended domains of influence. For these reasons, computing the stiffness matrix in mesh free methods is a very computationally demanding task which needs special attention in order to be affordable in real-world applications.
- Finally methods that do not require a mesh at all and use the set of scattered nodes, or irregular grids for integration reasons, are less stable and less accurate. Collocation methods and the point interpolation method (PIM), belong to this category. Automation of nodal selection, smoothing and improving the stability of the solution are some of the challenges in these kinds of methods.

1.2 Need for Mesh free Methods

FEM is robust and has been proposed for static and dynamic problems for linear and nonlinear stress analysis of solids, structures, as well as fluid flows, giving stable solutions. Many practical engineering problems are currently solved using a large variety of well-developed FEM packages that are commercially available and easy to be used. However, in the attempt to solve problems with complex geometries, containing large discontinuities, the following limitations of FEM are becoming increasingly significant:

- The creation of a mesh is a prerequisite using the FEM packages. A good-quality mesh that can give a stable solution, may sometimes need a great amount of manpower time. At the same time, nowadays the cost of CPU (central processing unit) time is drastically decreasing. Therefore, it would be ideal for the meshing process to be fully performed by the computer, without any human intervention.
- In stress calculation, the stresses obtained using FEM are discontinuous and less accurate. Using mesh free methods the stresses results become significantly smoother.
- When handling large displacements the element distortion of the FEM results in low accuracy and numerical instabilities.
- Furthermore, the rational of FEM is the continuum assumption. This means that, the elements of the discetization are treated separately but then assembled into the structure. In this way, the elements intrinsically clutch together without separations. Hence, for problems involving fracture, the finite elements might not be the most suitable choice. In fact, for fracture problems, the mesh is always conforming to the crack faces, because the continuum assumption does not allow inter-element cracks. The elements can either be totally "eroded" or stay as a whole piece. This usually leads to a misrepresentation of the breakage path.
- Re-mesh approaches at subsequent steps have been proposed for handling these types of problems in FEM. Besides the obvious increased computational burden, a re-mesh conforming to the crack could also lead to a low quality mesh and degradation in the solution. Moreover, as linear elastic fracture mechanics leads to singular stresses and strains, a very fine mesh might be necessary to capture properly the stress intensity.

A close examination of these difficulties associated with FEM reveals the need of proposing mesh free methods. Because there is no need to create a mesh, the procedure of node creation can become completely automated, saving time. Furthermore, mesh free methods provide flexibility in adding or deleting nodes, making it easy to perform adaptive schemes and hierarchical solutions. For example,

in crack propagation problems referred above, nodes can easily be added around a crack tip to capture the stress concentration, even singularily.

1.3 Outline of the Diploma Thesis

1.3.1 Choice of Mesh Free Method

The present diploma thesis focuses on and uses the element free Galerkin (EFG) method. EFG method is a meshless method developed by Belytschko et al. (1994b) and it is based on the diffuse elements method (DEM) originated by Nayroles et al. (1992). The major features of the EFG method are as follows:

- Moving least square approximation (MLS) is employed for the construction of the shape functions.
- The Galerkin weak form is employed to develop the discretized system equation.
- Background cells (referred as Gauss cells) are employed to carry out the Gauss quadrature scheme, common in the FEM approximations.

Detailed formulation and equations are provided. Applications of the EFG method are presented for solving academic and engineering, one-dimensional and two dimensional linear solid mechanics problems.

1.3.2 Hierarchical Refinement

As we have already remarked by using numerical methods for solving complex differential or partial differential equations we get an approximate solution of the problem.
In finite element method the accuracy of this solution is based on the density of the mesh used, as well as the type of the finite element. To enhance the accuracy in a critical region, .e.g., around a crack tip or a region with high stress density, one can use refinement techniques. Broadly these fall into two categories:

- The *h*-refinement in which the same class of elements continues to be used but it is changed in size, i.e. in some locations made larger and in others made smaller, in order to provide maximum efficiency in reaching the desired solution. Generally, the h-refinement calls for a finer mesh in that critical region.
- The *p*-refinement in which we continue to use the same element size but simply increase, generally hierarchically, the order of the polynomial used in the definition.

On occasion it is possible to combine efficiently the *h*- and *p*-refinements and call it the *hp*-refinement. In this procedure both the size of elements and their degree of polynomial p are altered.
Sometimes the areas of the problem domain, -which need to be refined-, are not *a priori* known. For this reason, a gradual improvement of the solution need to be made, based on previous results. An improvement by an *a posteriori* treatment of the finite element data is feasible using an error estimator. We refer to such processes as recovery. By applying this method, the errors can be reduced once a finite element solution has been obtained.
The essential characteristics of the mesh free methods are that there is no need for high structured meshes as required in finite element methods. Obviously, the major advantages of mesh free methods must be closely related to those characteristics.
In this diploma thesis a refinement technique for the EFG method is proposed. As a predefined mesh is not necessary, the refinement in mesh free methods can be achieved by simply sprinkling an arbitrary number of nodes in the areas of increased stress error. For this refinement process, the extra degrees of freedom are added to the problem domain, without any need to alter the already existing shape functions, concerning the initial node arrangement. This leads to a reduction of the computational cost, as the previous stiffness matrix of the problem solved, stays constant and is not again calculated. This refinement procedure will be called "hierarchical" as it is based in hierarchical concepts for the EFG method that are proposed. In order to clarify this technique the theoretical background is shown, as well as two dimensional linear solid mechanics problems are solved.

2 Theory of Elasticity

The basics of mechanics for solids and structures are outlined in a general manner in this chapter. As the fundamentals of the elastic theory are about to be used for solving linear elastic problems, with the mesh free method proposed, it was considered quite vital, for a brief report of the basics.

All structural materials possess to a certain extent the property of *elasticity*, i.e., if external forces, producing *deformation* of a structure, do not exceed a certain limit, the deformation disappears with the removal of the forces. It is assumed that the mass of an elastic body is *homogeneous* and continuously distributed over its volume so that the smallest element cut from the body possesses the same specific physical properties as the body. It is also assumed that the body is isotropic, i.e., that the elastic properties are the same in all directions.

Solids and structures made of solids are stressed when they are subjected to *loads* or *forces*. The stresses are, in general, not uniform as the forces usually vary spatially. The *stresses* lead to *strains*, which can be observed as *deformation* or *displacement*. Solid mechanics and structural mechanics deal with the relationships between stresses and strains, displacements and forces, stresses (strains) and forces for given boundary conditions of solids and structures. These relationships are critical in modeling, simulating, and designing engineered structural systems.

2.1 Equations for Three-Dimensional Solids

2.1.1 Stress and Strain

Let Figure 2.1 represent a continuum 3D elastic body in equilibrium with volume $\boldsymbol{\Omega}$ and surface boundary Γ. The boundary is further divided into two types of surfaces. The one on which the external forces are prescribed is denoted as Γ_t, and the surface on which the displacements are prescribed is denoted as Γ_u .The solid can also be loaded by body force $\mathbf{b}$ and surface traction (force) $\mathbf{t}$ in any distributed fashion in the volume of the solid.

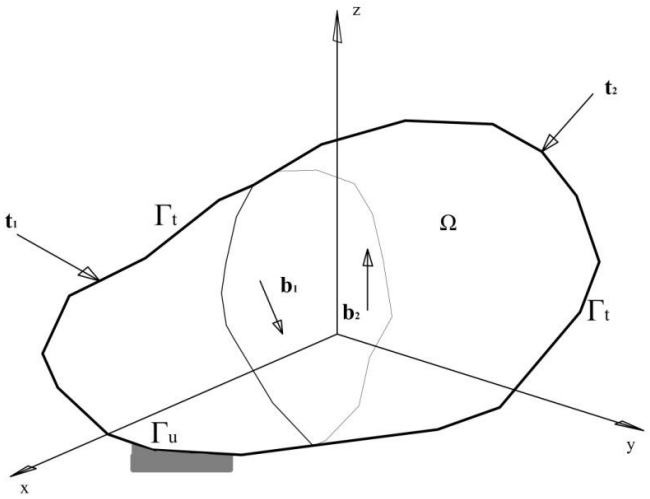


Figure 2.1
A continuum of solids subjected to forces

If we take a very small cubic element at any point in the solid, with sides parallel to the coordinate axes, there are, in general, six components of stress acting on the sides of this element and the directions taken as positive are as indicated in Figure 2.2. On each surface, there is a normal component of stress and two component of shearing stress. The sign convention for the subscript is that the first letter represents the surface on which the stress is acting, and the second letter represents the direction of the stress.

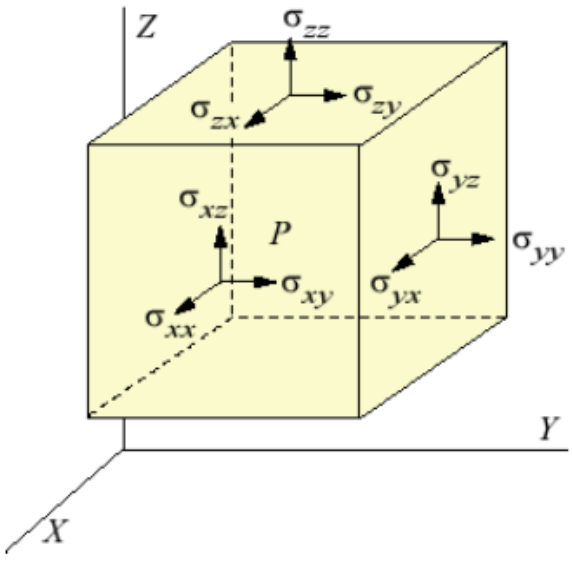


Figure 2.2
Stress components at point P in a solid

In total, there are nine stress components shown on the cubicle. By taking moments of force about the central axes of the cubicle at the state of equilibrium, it is easy to confirm that

$$\sigma_{xy} = \sigma_{yx}; \qquad \sigma_{xz} = \sigma_{zx}; \qquad \sigma_{zy} = \sigma_{yz} \tag{2.1}$$

Therefore, there are in total six independent stress components at a point in solids. These stresses form a *stress tensor*. It is also often written in vector form :

$$\boldsymbol{\sigma}^{\mathrm{T}} = \{ \sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \sigma_{yz}\ \sigma_{xz}\ \sigma_{xy} \} \tag{2.2}$$

At the same time there are six corresponding strain components at any point in solids, as a result of the stress condition, which can also be written in a vector form respectively :

$$\boldsymbol{\varepsilon}^{\mathrm{T}} = \{ \varepsilon_{xx}\ \varepsilon_{yy}\ \varepsilon_{zz}\ \varepsilon_{yz}\ \varepsilon_{xz}\ \varepsilon_{xy} \} \tag{2.3}$$

Concerning the displacement components *u, v, w* in the *x, y,* and *z* direction, respectively, we also need to correlate the components of strain with the displacements in every direction. The strain is the rate of displacements per length, and therefore the components of strain can be obtained by derivatives of the displacement as follows :

$$\begin{aligned}
\varepsilon_{xx} &= \frac{\partial u}{\partial x} \\
\varepsilon_{yy} &= \frac{\partial v}{\partial y} \\
\varepsilon_{zz} &= \frac{\partial w}{\partial z} \\
\varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\
\varepsilon_{xz} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\
\varepsilon_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}
\end{aligned} \tag{2.4}$$

The strain- displacement components in the *x, y, z* direction can be written in the following matrix form :

$$\boldsymbol{\varepsilon} = L\mathbf{u} \tag{2.5}$$

where **u** is the displacement vector of the form :

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{2.6}$$

and ***L*** is a differential operator matrix given by

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \\ \partial/\partial y & \partial/\partial x & 0 \end{bmatrix} \tag{2.7}$$

2.1.2 Generalized Hooke's law

The constitutive equation, which gives the relationship between stress and strain in a solid material, is often termed as the Hooke's law. The generalized Hooke's law for anisotropic materials can be given in matrix form as follows :

$$\boldsymbol{\sigma} = c\boldsymbol{\varepsilon} \tag{2.8}$$

where **c** is a matrix of material constants, which have been obtained experimentally. The exact constitutive equation can be written as :

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & sym. & & & c_{55} & c_{56} \\ & & & & & c_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{yz} \\ \varepsilon_{xz} \\ \varepsilon_{xy} \end{Bmatrix} \tag{2.9}$$

Taking into account that $c_{ij} = c_{ji}$,there are 21 independent material constants c_{ij} . For isotropic materials, **c** can be gradually reduced to :

$$\mathbf{c} = \begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ & c_{11} & c_{12} & 0 & 0 & 0 \\ & & c_{11} & 0 & 0 & 0 \\ & & & \frac{c_{11}-c_{12}}{2} & 0 & 0 \\ & sym. & & & \frac{c_{11}-c_{12}}{2} & 0 \\ & & & & & \frac{c_{11}-c_{12}}{2} \end{bmatrix} \tag{2.10}$$

where

$$c_{11} = \frac{E(1-\nu)}{(1-2\nu)(1+\nu)}; \quad c_{12} = \frac{E\nu}{(1-2\nu)(1+\nu)}; \quad \frac{c_{11}-c_{12}}{2} = G \tag{2.11}$$

in which E, ν and G are Young's modulus, Poisson's ratio, and shear modulus of the material. Taking into consideration that:

$$G = \frac{E}{2(1+\nu)} \tag{2.12}$$

there are finally only two independent constants among these three. Given any two of these constants, the other can be calculated using the above equation.

2.2 Equations for Two Dimensional Solids

2.2.1 Stress and Strain

For a 2D solid let us assume, that all dependent variables are independent of the z coordinate, and all external loads are applied only in x-y axis. There are two basic types of 2D solids. The *plane stress* solid, and the *plain strain* solid.
If a thin plate is loaded by forces applied at the boundary, parallel to the plane of the plate and distributed uniformly over the thickness (Figure 2.3), the stress components σ_z, σ_{xz}, σ_{yz} are zero on both faces of the plate, and it may be assumed, tentatively, that they are zero also within the plate. The state of stress is then specified by σ_x, σ_y, σ_{xy} only, and is called *plane stress.*

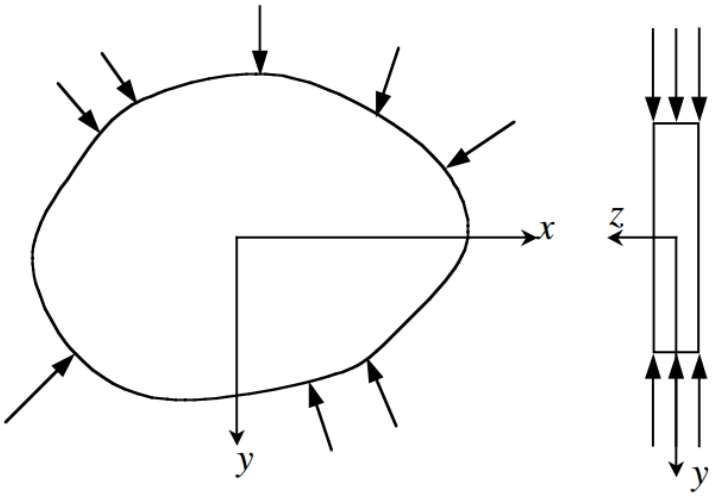


Figure 2.3
The plane stress problem

A similar signification is possible at the other end, when the dimension of the body in the z-direction is much larger compared with the dimensions in the x and y directions. External forces are applied along the z axis and the movement in the z direction at any point is constrained. The strain components in the z direction (ε_{zz}, ε_{xz}, ε_{yz}) are, therefore all zero, and we only have three in-plane stresses to handle. There are many important problems of this kind as the retaining wall with lateral pressure, indicated in Figure 2.4.The state of stress is then called *plane strain*.

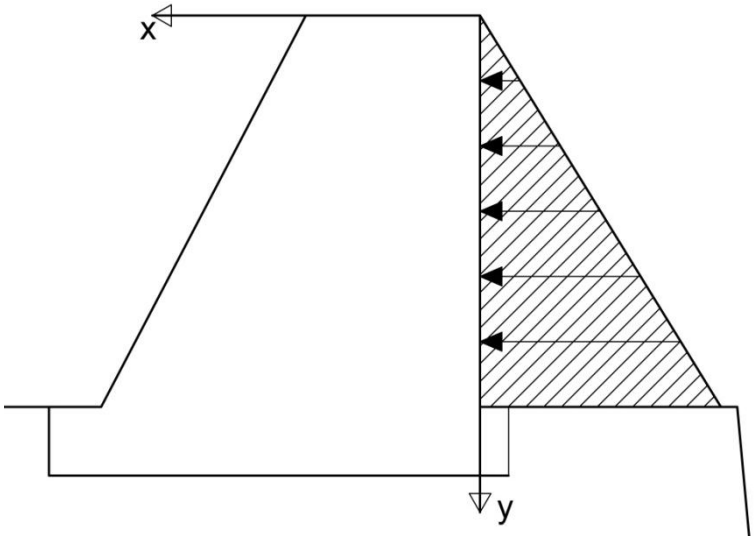


Figure 2.4
The retaining wall. A plane stress problem

The system equations for 2D solids can be obtained by simply omitting terms related with to the z direction in the system equations for 3D solids.

The stress components are:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} \tag{2.13}$$

The three corresponding strain components at any point in 2D solids, can also be written in a similar vector form:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} \tag{2.14}$$

The strain- displacement components in the *x, y* direction are :

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x} \\ \varepsilon_{yy} &= \frac{\partial v}{\partial y} \\ \varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{aligned} \tag{2.15}$$

The strain- displacement components in the *x, y* direction can be written in the following matrix form :

$$\boldsymbol{\varepsilon} = L\mathbf{u} \tag{2.16}$$

where the displacement vector has the form of:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{2.17}$$

and ***L*** is a differential operator matrix given by:

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \tag{2.18}$$

2.2.2 Hooke's law

For 2D solids Hooke's law has the following matrix form :

$$\boldsymbol{\sigma} = c\boldsymbol{\varepsilon} \tag{2.19}$$

where ***c*** is a matrix of material constants, which have been obtained experimentally. For isotropic materials, we have:

$$\mathbf{c} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \qquad \text{(Plane stress)} \tag{2.20}$$

For plane strain problems, the matrix of material constants can be obtained, if we set $E^*=E/(1-\nu^2)$ and $\nu^*=\nu/(1-\nu)$ to the plane stress matrix of material constants, which leads to

$$\mathbf{c} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \qquad \text{(Plane strain)} \tag{2.21}$$

2.3 Equations for Truss Members

A truss member is a solid whose dimension in one direction, is much larger than the other two. The forces are applied only in the basic direction, let it be the x direction. Consequently, a truss member is a 1D solid, and the equations for it, can be obtained by simply omitting terms related with to y and z direction, in the system equations for 3D solids.

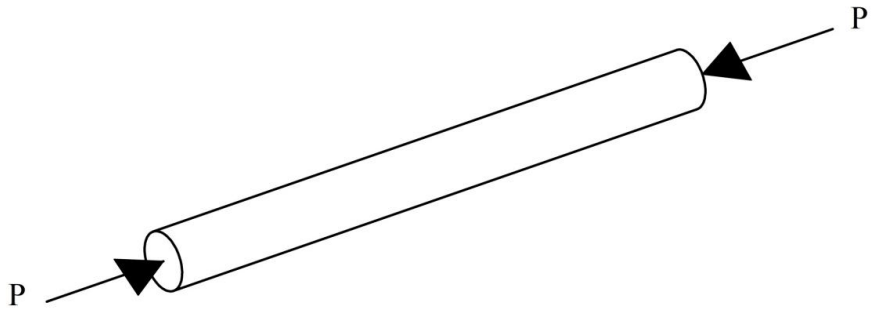


Figure 2.5
A truss member

2.3.1 Stress and Strain

The stress in a truss member is the σ_{xx} one, often simplified as σ_x. The respective strain in a truss member is ε_{xx}, simplified as ε_x. The strain-displacement relationship is simply given by

$$\varepsilon_x = \frac{\partial u}{\partial x} \tag{2.22}$$

2.3.2 Hooke's Law

This is actually the original Hooke's Law, describing the experimental procedure we can get the Young's modulus from, a tensile test of a truss member.

$$\sigma = \mathrm{E}\varepsilon \tag{2.23}$$

3 Moving Least Square (MLS) Approximation

Moving Least Squares (MLS) originated by mathematics for smoothing and interpolating data (Lancaster and Salkauskas, 1981). Currently the MLS method is widely used for constructing mesh free shape functions for approximation. Nayroles et al. (1992) were the first to use MLS approximation to construct shape functions for their diffuse element method (DEM) for mechanics problems. DEM was modified by Belytschko et al. (1994b), who named it Element Free Galerkin method.

3.1 Basic approximations

The field variable u at any point at $\mathbf{x}$ = (x, y, z) within the problem domain is approximated using the displacements at every node within the *support domain* of the point x, i.e.,

$$u(\mathbf{x},t) = \sum_{I \in S} \Phi_I(\mathrm{x})\, \mathrm{u}_I(\mathrm{t}) \tag{3.1}$$

where $\Phi_I : \Omega \rightarrow \mathbb{R}$ are the shape functions and the u_I 's are the nodal values at principle I located at position x_I, and S is set of nodes I for which $\Phi_I(x) \neq 0$ .Note that the above form is identical to a FEM interpolation.

3.2 The concept of the support domain

The support domain of a point **x**, within the problem domain, determines the number of nodes to be used to support or approximate the function value at **x**. A support domain can be (but does not have to be) weighted using a *weight function*, as shown in Figure 3.1. It can have different shapes and its dimension and shape can be different for different point of interest **x**, shown in Figure 3.2. The most often used shapes are either circular or rectangular.

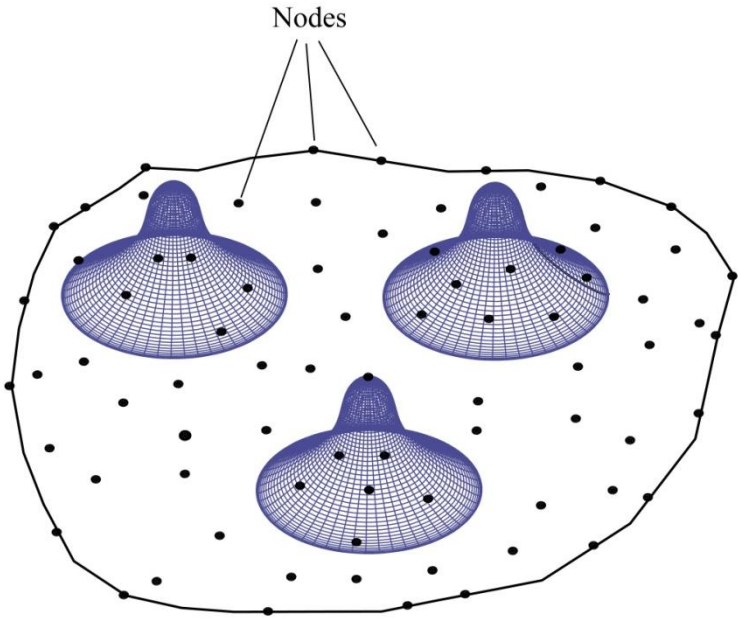


Figure 3.1
Domain representation of a 2D structure with arbitrary nodes in a weighted support domain

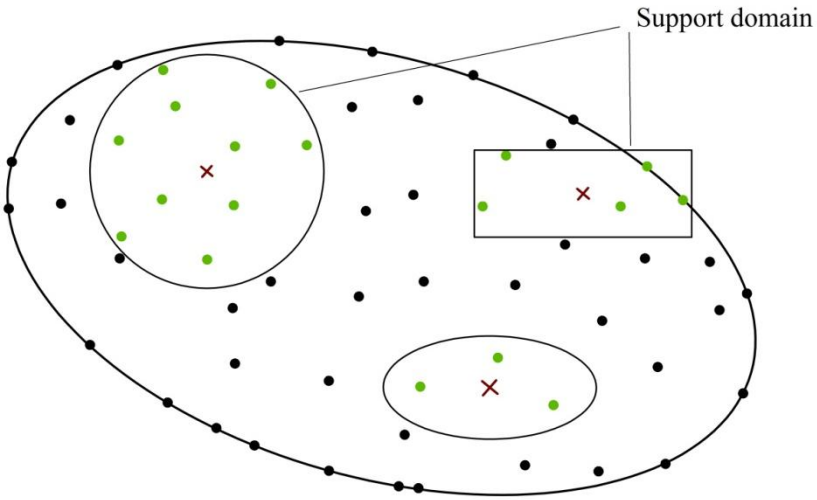


Figure 3.2
Support domain determines nodes (marked with green color) that are used for approximation or interpolation of field variable at point **X**. Support domains are displayed with different shapes. More often used shapes are circular or rectangular

The concept of the support domain works well if the nodes do not vary too drastically in density in the problem domain. In the EFG method that is applied in this diploma thesis, we use the support domain to select the nodes either to construct shape functions or integrate within the problem domain.
The accuracy of the approximation depends on the nodes in the support domain of the point of interest. As a result, a proper support domain should be chosen to ensure

a completeness for interpolation. Generally, to define the dimension of the support domain d_s , for a point **x** we use :

$$d_s = d_m d_c \tag{3.2}$$

where d_m is the dimensionless size of the support domain and d_c is the nodal spacing near the point **x**. If the nodes are uniformly distributed d_c is simply the distance between two neighboring nodes. For non uniformly distributed nodes, d_c can be chosen as an average nodal spacing for the points located in the support domain of the point **x**.

The role of the dimensionless d_m is to relate the support domain with the average nodal spacing. It is therefore the factor of the average nodal spacing. This factor should be predefined before the analysis is carried out. Generally, values of d_m =2.0 to 3.0 lead to good results.

3.3 MLS procedure

We consider a problem domain Ω. Let $u(x)$ be the function of the field variable defined in the domain Ω. The approximation of $u(x)$ at point x is denoted $u^h(x)$.The MLS approximation first writes the field function in the form:

$$u^h(\mathbf{x}) = \sum_i^m p_i(x) a_i(x) = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{3.3}$$

where m is the number of terms of monomials (polynomial basis), and $\mathbf{a(x)}$ is a vector of coefficients given by

$$\mathbf{a}^{\mathbf{T}}(\mathbf{x}) = \{a_0(x)\ a_1(x) \ldots a_m(x)\} \tag{3.4}$$

To ensure minimum completeness, in Eq. (3.3) $\mathbf{p(x)}$ is usually a vector that consists of the lowest order monomials.

- In 1D space a complete polynomial basis of order m is given by:

$$\mathbf{p}^T(x) = \{1, x, x^2, \ldots, \mathrm{x}^m\} \tag{3.5}$$

- In 2D space, we have:

$$\mathbf{p}^{\mathbf{T}}(\mathbf{x}) = \mathbf{p}^{\mathbf{T}}(x, y) = \{1, x, y, xy, x^2, \mathrm{y}^2 \ldots, \mathrm{x}^m, y^m\} \tag{3.6}$$

  which can be obtained by the Pascal triangle of monomials shown in Figure 3.3.

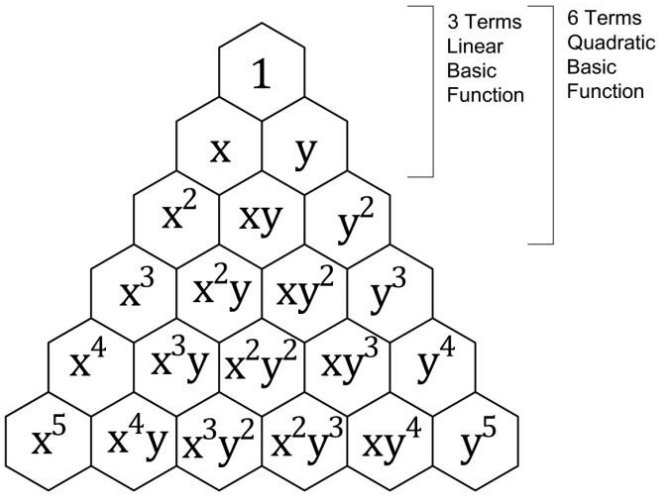


Figure 3.3
Pascal triangle of monomials, 2D case

- Finally in 3D space, we derive:

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(x, y, z) = \left\{1, x, y, z, xy, \mathrm{yz}, \mathrm{zx}, x^2, \mathrm{y}^2, z^2 \ldots, \mathrm{x}^m, y^m, z^m\right\} \quad (3.7)$$

Note that $\mathbf{a(x)}$ is an arbitrary function of $\mathbf{x}$. A functional of weighted residual is constructed using the approximated values of the field function and the nodal parameters, $\mathrm{u_I} = \mathrm{u}(\mathrm{x_I})$. The approximation function $u^h(x)$ and the nodal parameters u_i are shown in Figure 3.4. The weighted, discrete L_2 norm can be written as:

$$\begin{aligned} J &= \sum_{I=1}^{n} w(x - x_I)[\mathrm{u_h}(\mathrm{x}) - \mathrm{u}(\mathrm{x_I})]^2 \\ &= \sum_{I=1}^{n} w(x - x_I)[\mathbf{p}^{\mathrm{T}}(x_I)\mathbf{a(x)} - \mathrm{u}(\mathrm{x_I})]^2 \end{aligned} \quad (3.8)$$

where $w(x - x_I)$ is a weight function with compact support and n is the number of points in the neighborhood of x, for which the weight function $w(x - x_I) \neq 0$, and u_I is the nodal value of u at $x = x_I$. The weight function ensures that nodes leave or enter the support domain in a smooth weighted manner when x moves.

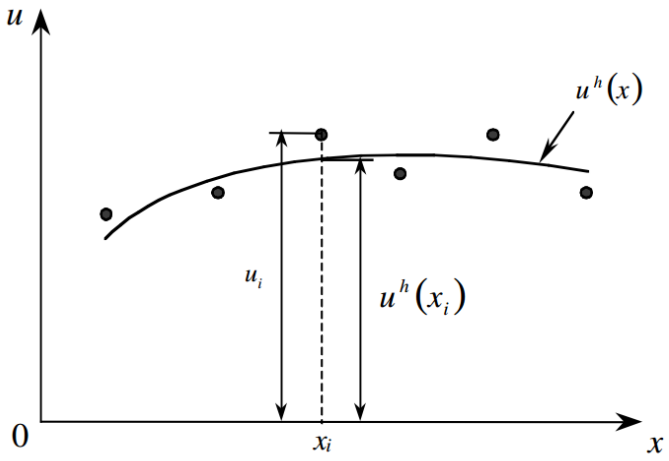


Figure 3.4
The approximation function $u^h(x)$ and the nodal parameters u_i in the MLS approximation

Eq. (3.8) can be rewritten in the form

$$J = \mathbf{(Pa-u)}^{\mathbf{T}}\mathbf{W(x)(Pa-u)} \tag{3.9}$$

where

$$\mathbf{u}^{\mathbf{T}} = (u_1, u_2, \ldots, u_n) \tag{3.10}$$

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix} \tag{3.11}$$

and

$$\mathbf{W(x)} = \begin{bmatrix} w(x-x_1) & 0 & \cdots & 0 \\ 0 & w(x-x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(x-x_n) \end{bmatrix} \tag{3.12}$$

Note that m is the number of terms of polynomial basic $p(x)$,which is usually much smaller than n, which is the number of nodes on the support domain. The requirement $n>>m$ ensures completeness for the functional approximation.

In MLS approximation, at an arbitrary point x, $\mathbf{a(x)}$ is chosen to minimize the functional J. The minimization condition requires :

$$\frac{\partial J}{\partial \mathbf{a}} = 0 \tag{3.13}$$

which leads to a system of *m* equations:

$$\begin{aligned}
&\frac{\partial J}{\partial a_1} = 0 \Leftrightarrow \sum_{I=1}^{n} w(x-x_I)2p_1(x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x)-u_I] = 0 \\
&\frac{\partial J}{\partial a_2} = 0 \Leftrightarrow \sum_{I=1}^{n} w(x-x_I)2p_2(x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x)-u_I] = 0 \\
&\quad\vdots \\
&\frac{\partial J}{\partial a_m} = 0 \Leftrightarrow \sum_{I=1}^{n} w(x-x_I)2p_m(x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x)-u_I] = 0
\end{aligned} \tag{3.14}$$

This is in vector notation

$$\sum_{I=1}^{n} w(x-x_I)2\mathbf{p}(x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x)-u_I] = 0 \tag{3.15}$$

$$2\sum_{I=1}^{n}[w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I)\mathbf{a}(x) - w(x-x_I)\mathbf{p}(x_I)u_I] = 0 \tag{3.16}$$

if we eliminate the constant factor and separate the right hand side

$$\sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I)\mathbf{a}(x) = \sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)u_I \tag{3.17}$$

or

$$\mathbf{A(x)a(x) = B(x)u} \tag{3.18}$$

where **A** is called the *weighted moment matrix* given by

$$\mathbf{A}(x) = \sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I) \tag{3.19}$$

and in Eq.(3.18) matrix $\mathbf{B}(x)$ has the form

$$\mathbf{B}(x) = \left[\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_n\right] \tag{3.20}$$

where

$$\mathbf{B}_I = w(x - x_I)\mathbf{p}(x_I) \tag{3.21}$$

and $\mathbf{u}$ is the vector that collects the nodal parameters of the field variable for all the nodes in the support domain as indicated on Eq. (3.10)

Solving the Eq. (3.18) for $\mathbf{a}(x)$, we obtain

$$\mathbf{a(x) = A(x)^{-1}B(x)u} \tag{3.22}$$

Substituting the above equation back in the Eq. (3.3) leads to

$$u^h(\mathbf{x}) = \mathbf{p^T}(x)\mathbf{a}(x) = \mathbf{p^T}(x)\mathbf{A^{-1}(x)B(x)u} \tag{3.23}$$

or more detailed

$$u^h(\mathbf{x}) = \mathbf{p^T}(x)\left[\sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I)\right]^{-1}\sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)u_I \tag{3.24}$$

This can be written shortly as

$$u^h(\mathbf{x}) = \sum_{I=1}^{n}\varphi_I(\mathbf{x})u_I = \Phi^{\mathrm{T}}(\mathbf{x})\,\mathbf{u} \tag{3.25}$$

where $\mathbf{\Phi(x)}$ is the matrix of MLS shape functions corresponding to n nodes in the support domain:

$$\Phi(\mathbf{x}) = \left[\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_n(\mathbf{x})\right] \tag{3.26}$$

and thus for one certain shape function φ_I at a point x

$$\varphi_I(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})[\mathbf{A}(\mathbf{x})]^{-1} w(x-x_I)\mathbf{p}(\mathrm{x}_1) \tag{3.27}$$

To derive the discrete system equations, it is necessary to derive the spatial derivatives of the MLS shape functions. For reasons of convenience the Eq. (3.25) is first rewritten using Eq. (3.23)

$$\mathbf{\Phi(x)} = \boldsymbol{\gamma}^T(\mathbf{x})\mathbf{B(x)} \tag{3.28}$$

where $\boldsymbol{\gamma}(\mathbf{x})$ is given by

$$\mathbf{A(x)}\boldsymbol{\gamma}(\mathbf{x}) = \mathbf{p(x)} \tag{3.29}$$

The vector $\boldsymbol{\gamma}(\mathbf{x})$ can be determined using an LU decomposition of the matrix $\mathbf{A}$ and followed by back substitution (Belytschko, Lu and Gu (1994), Dolbow and Belytschko (1998) and Liu (2003)).

As a result the partial derivatives of $\boldsymbol{\gamma(x)}$ can be obtained as follows:

$$\mathbf{A}\boldsymbol{\gamma}_{,i} = \mathbf{p}_{,i} - \mathbf{A}_{,i}\boldsymbol{\gamma} \tag{3.30}$$

$$\mathbf{A}\boldsymbol{\gamma}_{,ij} = \mathbf{p}_{,ij} - (\mathbf{A}_{,i}\boldsymbol{\gamma}_{,j} + \mathbf{A}_{,j}\boldsymbol{\gamma}_{,i} + \mathbf{A}_{,ij}\boldsymbol{\gamma}) \tag{3.31}$$

$$\mathbf{A}\boldsymbol{\gamma}_{,ijk} = \mathbf{p}_{,ijk} - (\mathbf{A}_{,i}\boldsymbol{\gamma}_{,jk} + \mathbf{A}_{,j}\boldsymbol{\gamma}_{,ik} + \mathbf{A}_{,k}\boldsymbol{\gamma}_{,ij} + \mathbf{A}_{,ij}\boldsymbol{\gamma}_{,k} + \mathbf{A}_{,ik}\boldsymbol{\gamma}_{,j} + \mathbf{A}_{,jk}\boldsymbol{\gamma}_{,\mathrm{i}} + \mathbf{A}_{,ijk}\boldsymbol{\gamma}) \tag{3.32}$$

where *i, j, k* indicate coordinates *x ,y* and *z*. A comma designates a partial derivative with respect to the indicated spatial variable. Thus the derivatives of shape function $\boldsymbol{\Phi}$ can be obtained in this way:

$$\boldsymbol{\Phi}_{,i} = \mathbf{B}\boldsymbol{\gamma}_{,i} + \mathbf{B}_{,i}\boldsymbol{\gamma} \tag{3.33}$$

$$\boldsymbol{\Phi}_{,ij} = \mathbf{B}\boldsymbol{\gamma}_{,ij} + \mathbf{B}_{,i}\boldsymbol{\gamma}_{,j} + \mathbf{B}_{,j}\boldsymbol{\gamma}_{,i} + \mathbf{B}_{,ij}\boldsymbol{\gamma} \tag{3.34}$$

$$\boldsymbol{\Phi}_{,ijk} = \mathbf{B}\boldsymbol{\gamma}_{,ijk} + \mathbf{B}_{,i}\boldsymbol{\gamma}_{,jk} + \mathbf{B}_{,j}\boldsymbol{\gamma}_{,ik} + \mathbf{B}_{,k}\boldsymbol{\gamma}_{,ij} + \mathbf{B}_{,ij}\boldsymbol{\gamma}_{,k} + \mathbf{B}_{,ik}\boldsymbol{\gamma}_{,j} + \mathbf{B}_{,jk}\boldsymbol{\gamma}_{,\mathrm{i}} + \mathbf{B}_{,ijk}\boldsymbol{\gamma} \tag{3.35}$$

In this section an outline of the basic laws and equations governing the Moving Least Square approximation (MLS) has been held. The theoretical background of computing the EFG method shape functions has also been presented. Special attributes of the MLS shape functions are about to be noted in next section, accompanied with useful depictions.

3.4 Choice of weight function

The weight function $w(x-x_I)$ plays an important role in the performance of the method. The weight function should be non-zero over the support domain of a node **x**. It should also be constructed in a way, so that it is positive and a unique solution **a(x)** is guaranteed; it should be relatively large for $\mathbf{x}_\mathrm{I}$ close to **x**, and relatively small when the distance from $\boldsymbol{x}$ increases. Therefore we will consider weight functions which depend only on the distance between two points as follows:

$$w(\mathbf{x}-\mathbf{x}_I) = w_I(d) \tag{3.36}$$

where $d = \| \mathbf{x}-\mathbf{x}_I \|$ is the distance between the two points $\mathbf{x}_\mathrm{I}$ and **x.**

Some commonly used weight functions, both of which are used in this diploma thesis:

- the cubic spline weight function

$$w(r)=\begin{cases}\frac{2}{3}-4r^2+4r^3, & r\le\frac{1}{2}\\ \frac{4}{3}-4r+4r^2-\frac{4}{3}r^3, & \frac{1}{2}<r<1\\ 0, & r\ge 1\end{cases} \tag{3.37}$$

- the quartic spline weight function

$$w(r)=\begin{cases}1-6r^2+8r^3-3r^4, & r<1\\ 0, & r\ge 1\end{cases} \tag{3.38}$$

with $r=\frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_s}$,where d_s is the size of the support domain of node *I*. Note that the weight function needs to be scaled to satisfy the condition of the partition of unity.

As it is already mentioned, in two dimensions, *circular* and *rectangular* support domains are used more frequently. Therefore for each support domain the weight functions can be written as follows :

- Circular support domain:

$$w(\mathbf{x}-\mathbf{x}_I)=w\left(\frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_s}\right) \tag{3.39}$$

- Rectangular support domain:

$$w(\mathbf{x}-\mathbf{x}_I)=w\left(\frac{|x-x_I|}{d_s^{\,x}}\right)w\left(\frac{|y-y_I|}{d_s^{\,y}}\right) \tag{3.40}$$

The derivatives of the weight function can be computed using the chain rule. For example for circular supports, we have :

$$w_k(r)=w_r(r)r_k=w_r\frac{x_k-x_{Ik}}{rd_s^{\,2}} \tag{3.41}$$

Figure 3.5-3.7 depict the two weight functions, as well as their first and second derivatives for the midpoint of a 1D truss member. The factor d_m of the support domain is chosen to be 2.5.

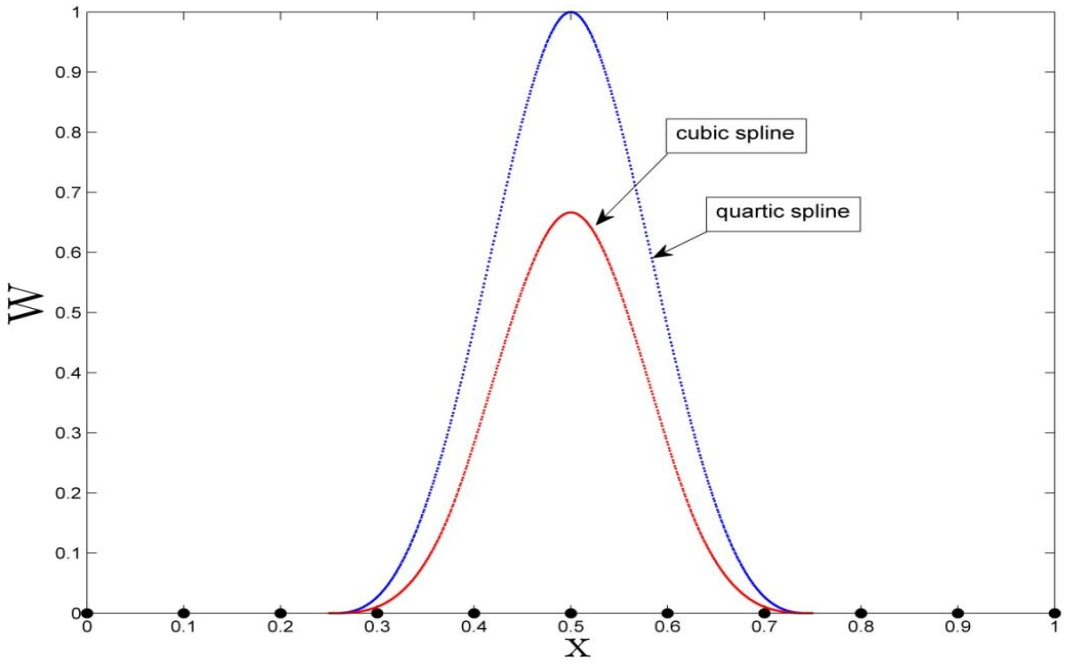


Figure 3.5
The cubic spline and the quartic spline weight functions

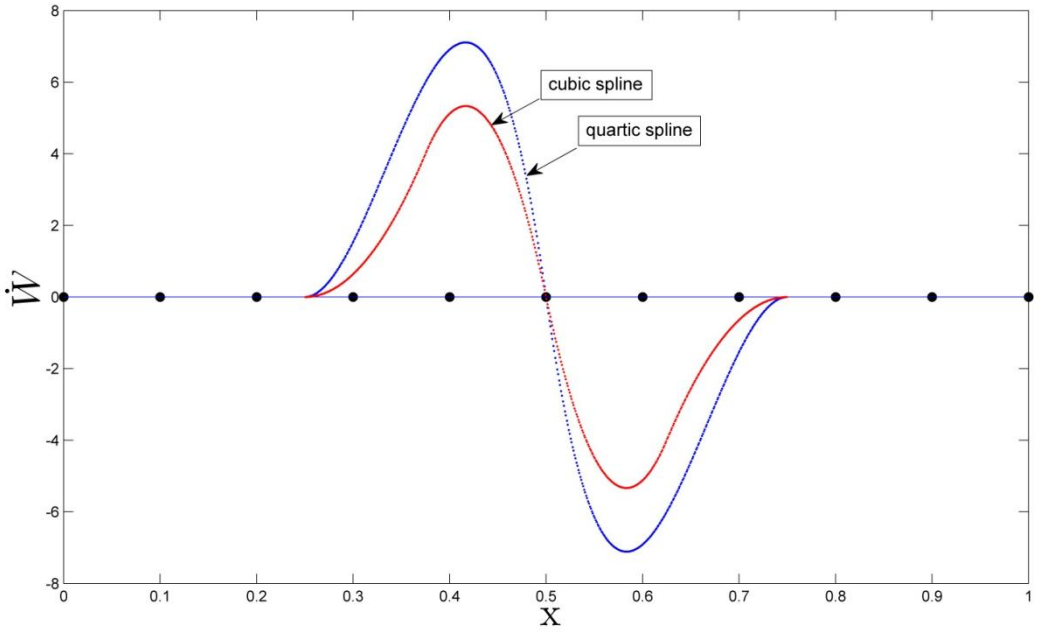


Figure 3.6
The first derivative of the weight functions

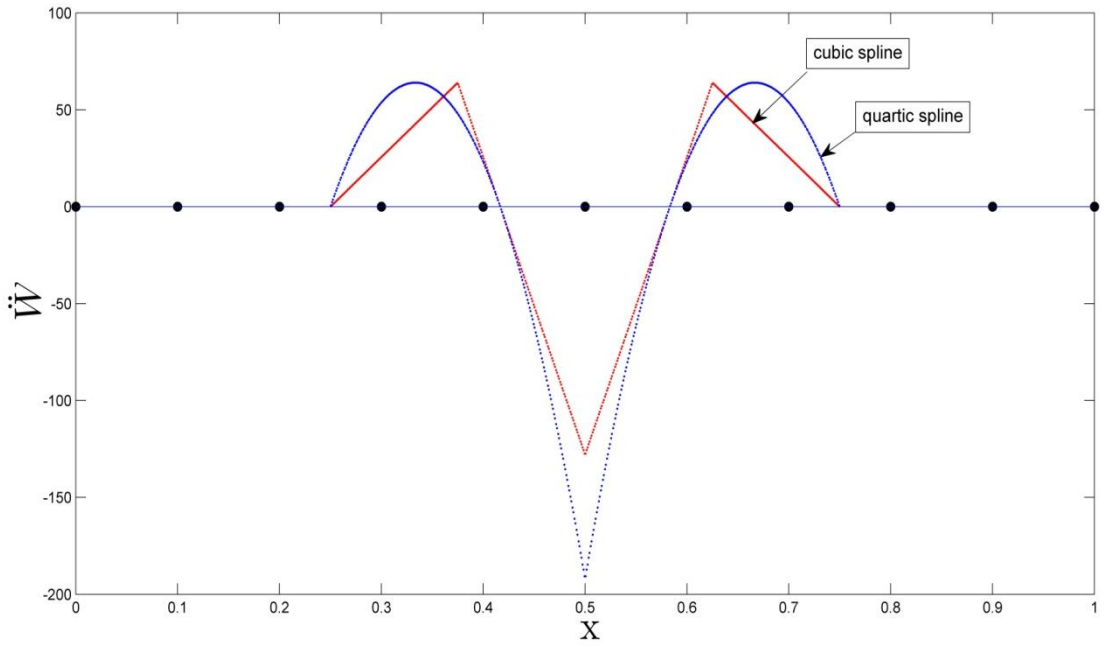


Figure 3.7
The second derivative of the weight functions

We can clearly see, that both weight functions are smooth and bell shaped, quite similar one another. By plotting the first derivatives of the weight functions, it is also shown that they still remain smooth, with piecewise the same behavior.
Figure 3.7 plots the second derivative of the weight functions. It is obvious that the second derivative of the cubic spline (marked with red color) is no longer smooth. This, along with the one piece formulation of the quartic spline, make it quite easier and more popular.

3.5 Moving Least Square (MLS) Shape Functions

A basic characteristic of the MLS shape functions is that they do not satisfy the Kronecker delta property, a criterion which is fulfilled for the FEM shape functions. In mathematics, the Kronecker delta, named after Leopold Kronecker, is a function of two variables, usually integers. The function is 1 if the variables are equal, otherwise it is equal to 0:

$$\delta_{Ij} = \begin{cases} 0 & if\ i \neq j \\ 1 & if\ i = j \end{cases} \tag{3.42}$$

where Kronecker delta δ_{Ij} is a piecewise function of variables *I* and *j*.

Thus $\varphi_I(\mathrm{x_j}) \neq \delta_{Ij}$ that results in $u^h(\mathbf{x}_I) \neq u_I$, i.e., the nodal parameters u_I are not nodal values of $u^h(\mathbf{x}_I)$.The MLS shape functions are not interpolants, but rather

approximations of the function. This leads to the fact, that the approximation of the displacement for node I, $u^h(\mathbf{x}_I)$, depends not only on the nodal displacement u_I but also on the displacements of the nodes located in the support domain of this node. This is illustrated in Eq. (3.25) and is an obvious result of the partition of unity. This property makes the imposition of essential boundary conditions more complicated than that in FEM.

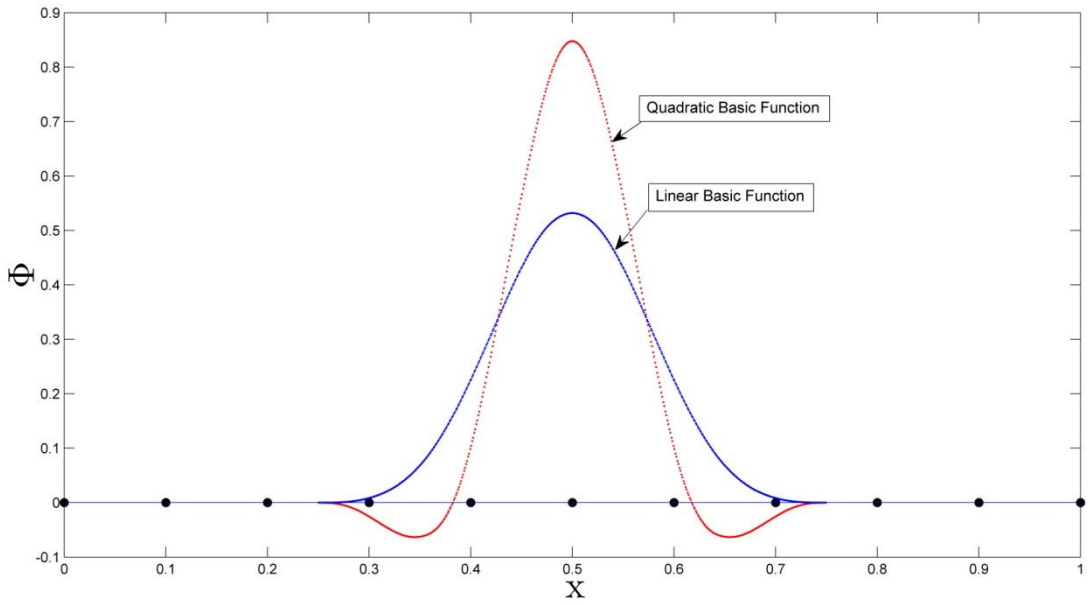


Figure 3.8
MLS shape functions in 1D space

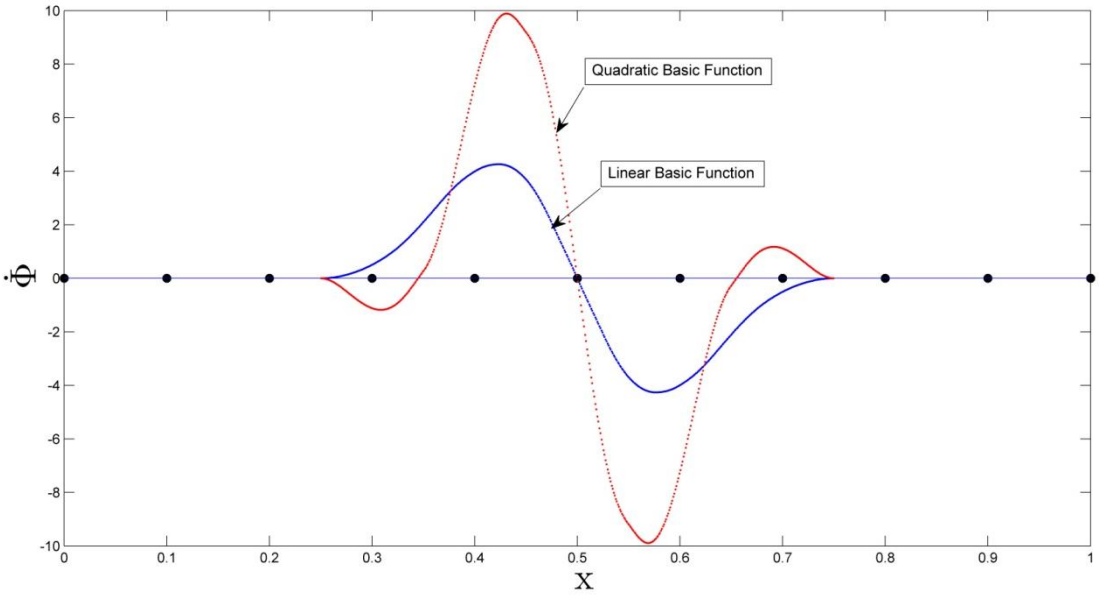


Figure 3.9
The first derivatives of the shape functions in 1D space

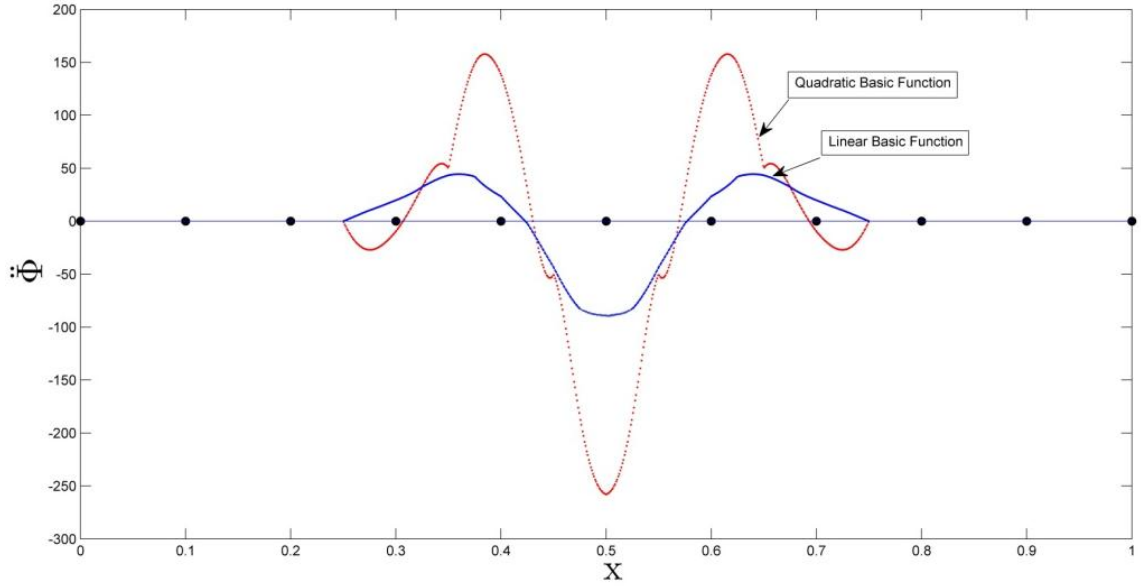


Figure 3.10
The second derivatives of the shape functions in 1D space

A plot of two typical 1D MLS shape functions, as well as their first and second derivatives for the midpoint of a 1D truss member is given in Figure 3.8-3.10. For the first (marked with blue color) the linear polynomial basic function $\mathbf{p}^T(x) = [1 \;\; x]$ has been chosen. For the second (marked with red color) the quadratic polynomial basic function $\mathbf{p}^T(x) = [1 \;\; x \;\; x^2]$ has been used. The factor d_m of the support domain is chosen to be $d_m = 2.5$.The cubic spline weight function has been employed.
It can be seen that the MLS shape function attains a maximum value that is considerably less than 1.To get smooth graphs, we computed the derivatives at 500 sampling points on the interval $0 \le x \le 1$.Note that MLS shape functions are not known in closed form.

Now we consider an interval $0 \le x \le 4$ divided into four equal classes. By plotting the shape functions for each node in the problem domain (Figure 3.11), we can visually understand that the MLS shape functions are indeed a partition of unity. The shape function referred to a node is not exactly equal to one (1) at this node. In addition, the same shape function is not exactly zero (0) at the other nodes in the domain. This remark justifies the special attention that should be paid to enforce essential boundary conditions during the solution of a boundary value problem.

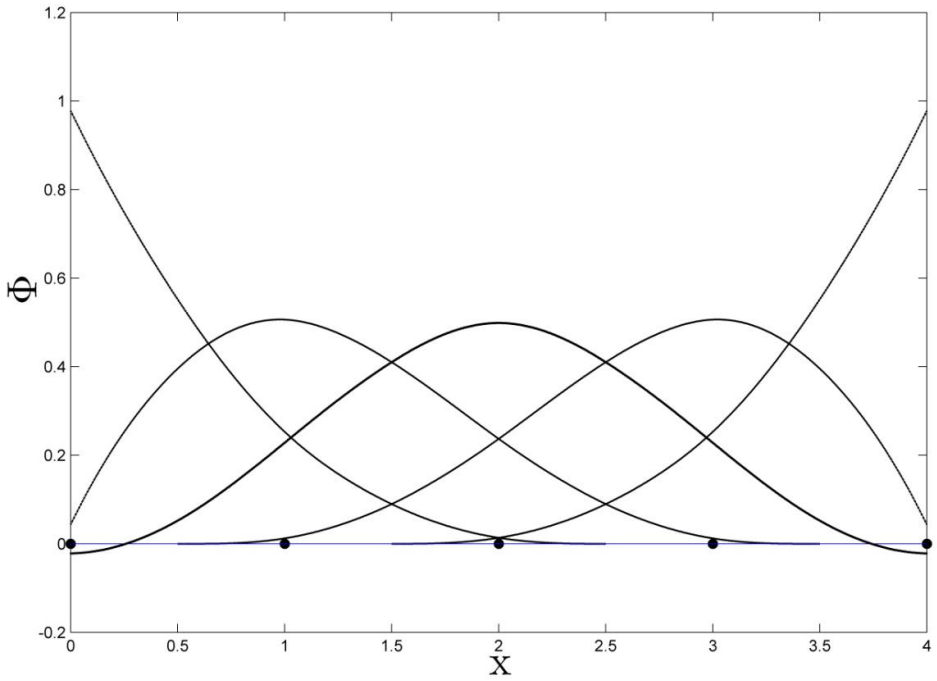


Figure3.11
Shape functions of each node. The overlap of the shape functions at the central node visualizes the partition of unity

For the depiction of the shape functions in two-dimensions, we will examine a 2D problem domain, where $0 \le x \le 1$ and $0 \le y \le 1$, divided into sixteen equal intervals. The cubic spline weight function has been used and the polynomial basic function is the linear one, $\mathbf{p}^T(\mathbf{x}) = [1 \;\; x \;\; y]$. The support domain is rectangular and the factor d_m of the support domain is chosen to be $d_m = 2$. The shape function associated with the centre node, as well as the first partial derivatives of this shape function, are shown in Figures 3.12-3.14.

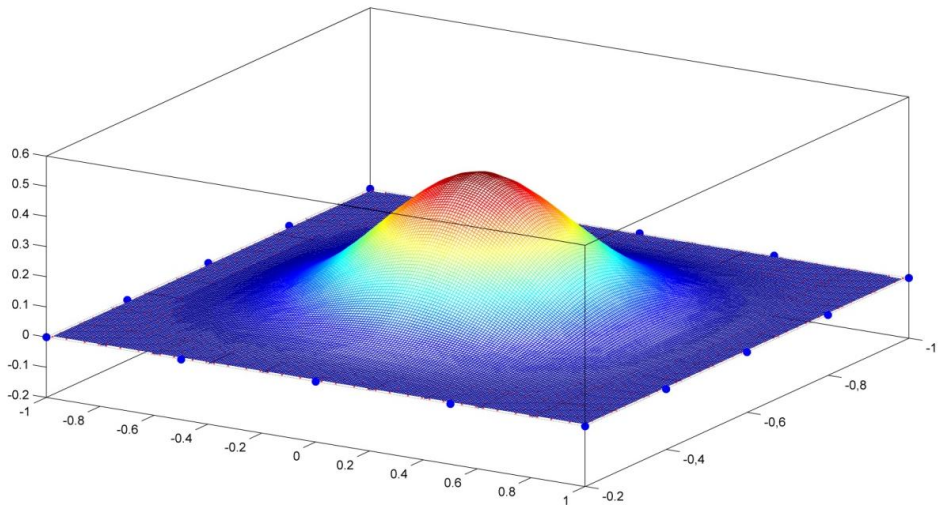


Figure3.12
MLS shape function in 2D space

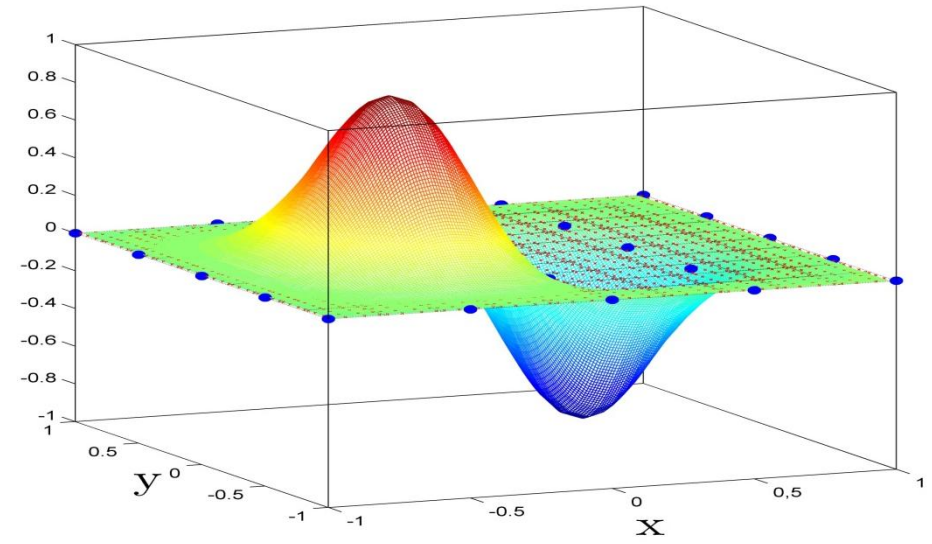


Figure 3.13
MLS shape function first partial derivative Φ_x

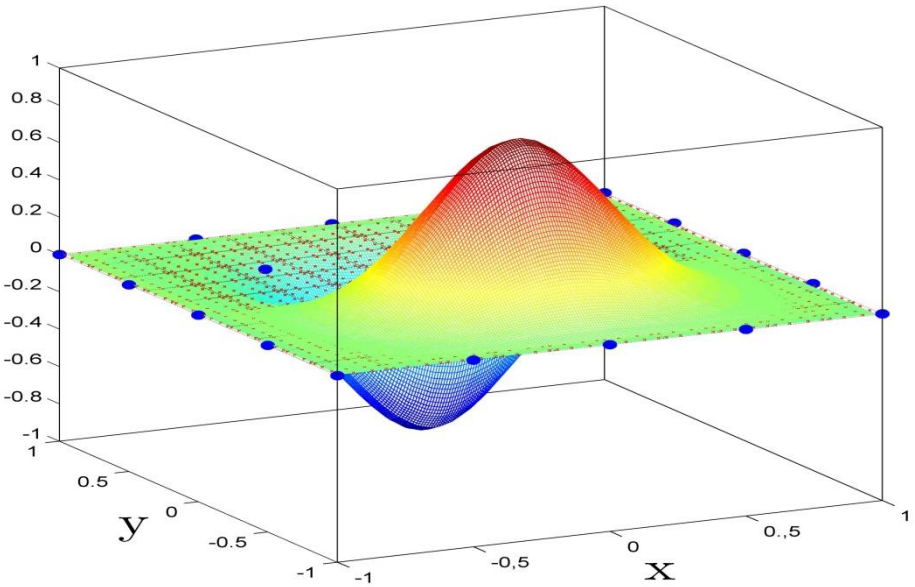


Figure 3.14
MLS shape function first partial derivative Φ_y

As we can notice, the shape functions in 2D shape are also bell shaped, and the first partial derivatives are symmetric. To get smooth graphs we computed the shape function, as well as the derivatives in 400 sampling points. A triangular based interpolation has been used to get color classification.

Finally, we will examine the influence of the dimension of the support domain, in the formulation of the shape functions, and generally the accuracy of the approximation. A proper choice of the size of the support domain is a "hot" issue for the MLS procedure, and it is an important decision to make. The choice of different factors d_m of the support domain, for this diploma thesis, has led to useful results that are presented below.

Let us, first of all, consider an 1D problem domain, $0 \le x \le 5$,divided into ten equal sub domains by eleven nodes. The function $u(x)$ is approximated by $u^h(x)$, using the nodal values at these nodes u_I, $1 \le I \le 11$. In each case, a different factor d_m is used, indicating the size of the support domain for each node. MLS procedure has been used to produce approximate functions, via the nodal values given. The cubic spline weight function has been chosen, and the polynomial basis is the linear one $\mathbf{p}^T(\mathbf{x}) = [1 \;\; x]$. Table 3-1 includes some random nodal values for the eleven nodes of the problem domain.

	Position *(x)*	**Nodal Value *(u_I)***
1	0	0,08
2	0,5	0,03
3	1	0,07
4	1,5	0,11
5	2	0,07
6	2,5	0,05
7	3	0,08
8	3,5	0,09
9	4	0,11
10	4,5	0,085
11	5	0,06

Table 3-1
Random values for the eleven (11) equidistant nodes

Thus, in Figure 3.15 we can see a graphical display for the approximate functions, calculated for different dimensions of the support domain.

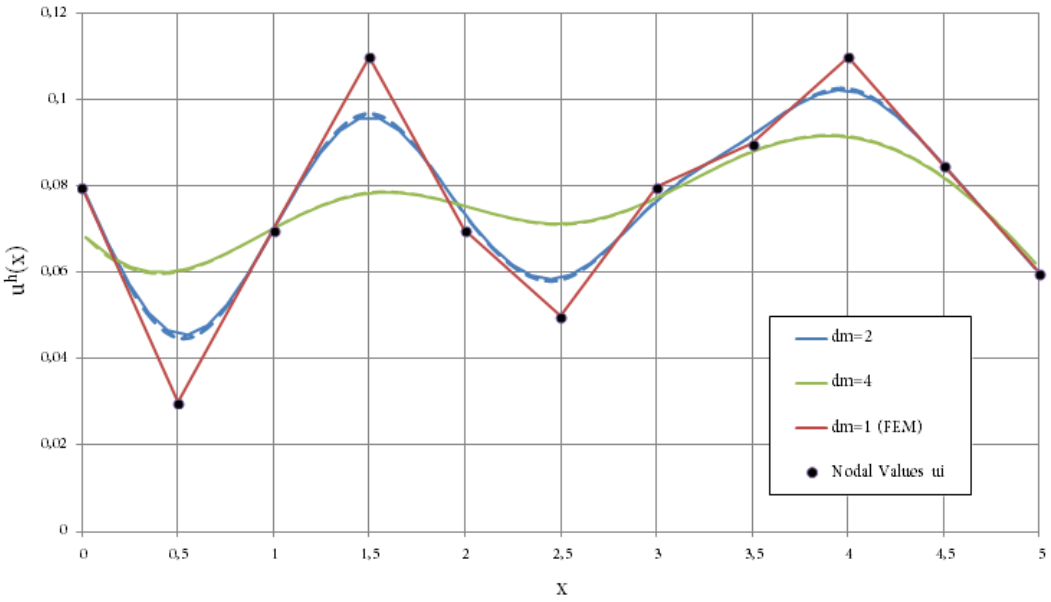


Figure 3.15
Moving Least Square (MLS) approximations for various dimensions of the support domain

Comparing these three graphs, we can make some remarks for the MLS approximations:

- The size d_m should be small enough to maintain the local character for the MLS approximation. If we compare the approximate function calculated for d_m=2, with the one calculated for d_m=4, we will easily notice that the first one is much more accurate than the second. This means that the more we increase the size of the support domain, the more slapstick approximations we get.

- On the other hand, the size d_m of the support domain, should be large enough, to have sufficiently large number of nodes to cover the domain of definition of the MLS approximation, ensuring regularity of **A(x)** matrix. What is more, a very small d_m may result in relatively large numerical error in using Gauss integration for the matrix calculation.

- Last but not least, if $d_m \to 1$, for the linear polynomial basis chosen, the approximate functions tends to become an interpolant rather than an approximation. This interpolant reminds us of the way the FEM approaches the solution.

We will refer extensively to this final remark, expanding it in terms of shape functions, for the MLS procedure. For this reason we consider again an interval $0 \le x \le 4$ divided into four equal domains. If we gradually reduce the dimension of the support domain in this problem we will get the depictions seen on Figure 3.16. In this part, the MLS shape functions are constructed by the linear basis.
We can notice that while we reduce the dimension of the support domain, the values of the shape functions referred to the nodes, at the point of the node, tend to one (1), while for a particular shape function, its value tends to zero at the other nodes in the domain. Therefore, some kind of local Kronecker delta property may be feasible, constituting a direct imposition of the essential boundary condition for the EFG method.

Figure 3.17 also shows the first derivative of the shape functions for the problem refereed. It is obvious, that the first derivative of the shape functions presented in Figure 3.16, tends to become constant as the size of the support domain is decreasing, which means that the approximation between the nodes of the problem domain tends to become linear. This remark reminds us of the linear FEM shape functions, governing the interpolation of two (2) nodes, for the 1D isoparametric problem, where:

$$\Phi_1 = \frac{1}{2}(1-x) \tag{3.43}$$

$$\Phi_2 = \frac{1}{2}(1+x) \tag{3.44}$$

and 1,2 are the interpolated nodes.

However it is already clear, that the use of so small factors for the size of the support domain can lead to numerical instabilities and relatively large numerical errors. Therefore a local alteration of the support domain is suggested, near the boundaries, to help with the direct imposition of boundary conditions. Although a question arises for a smooth transition between the areas with different support domains. Ideas can be found in *methods coupling mesh free and mesh-based* approximations that are already proposed. Note that the mesh free method is still in its early stage of development. Further advances on these procedures will deepen our understanding of mesh free methods.

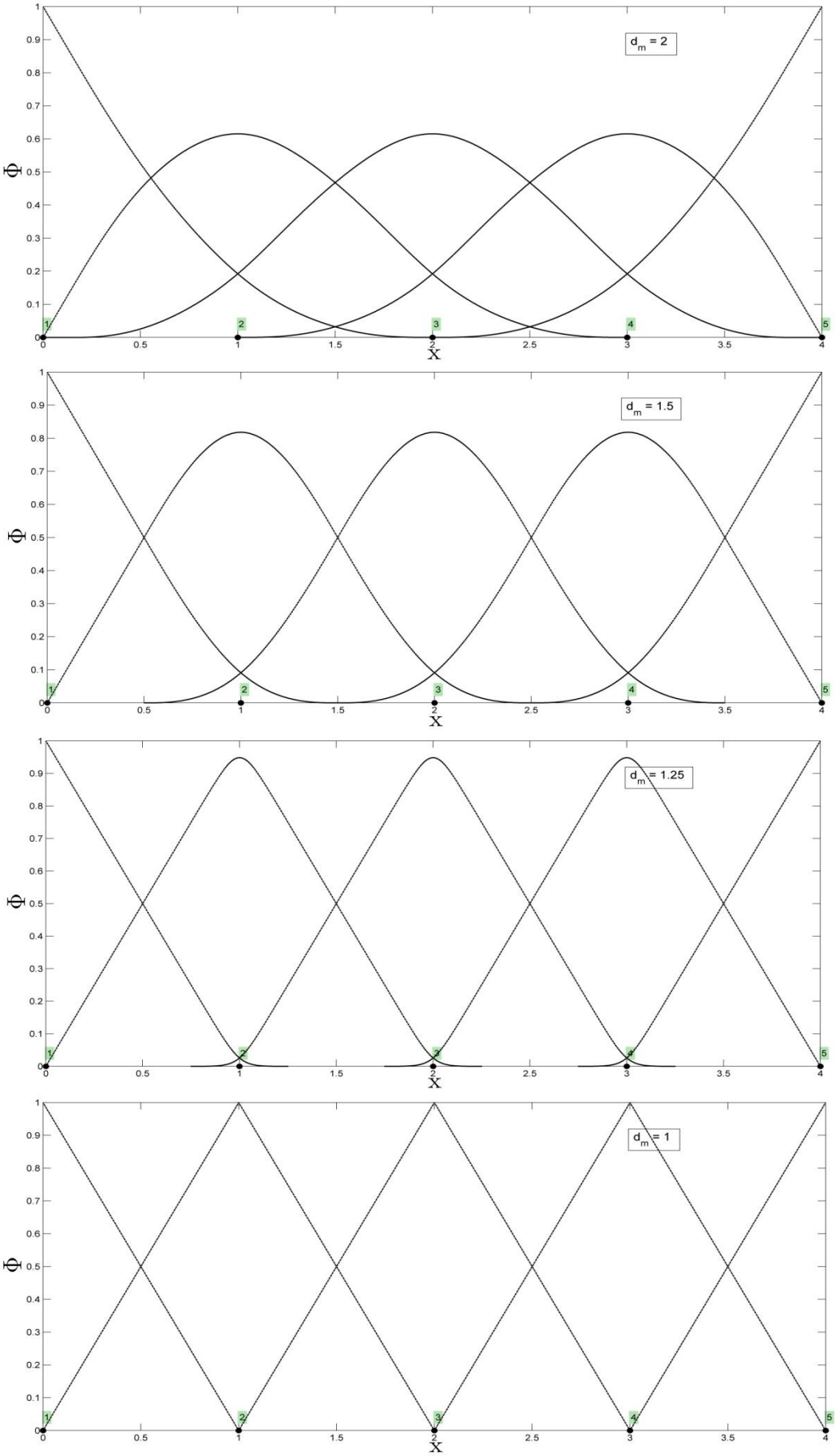


Figure 3.16
Shape functions of each node for different sizes of the support domain

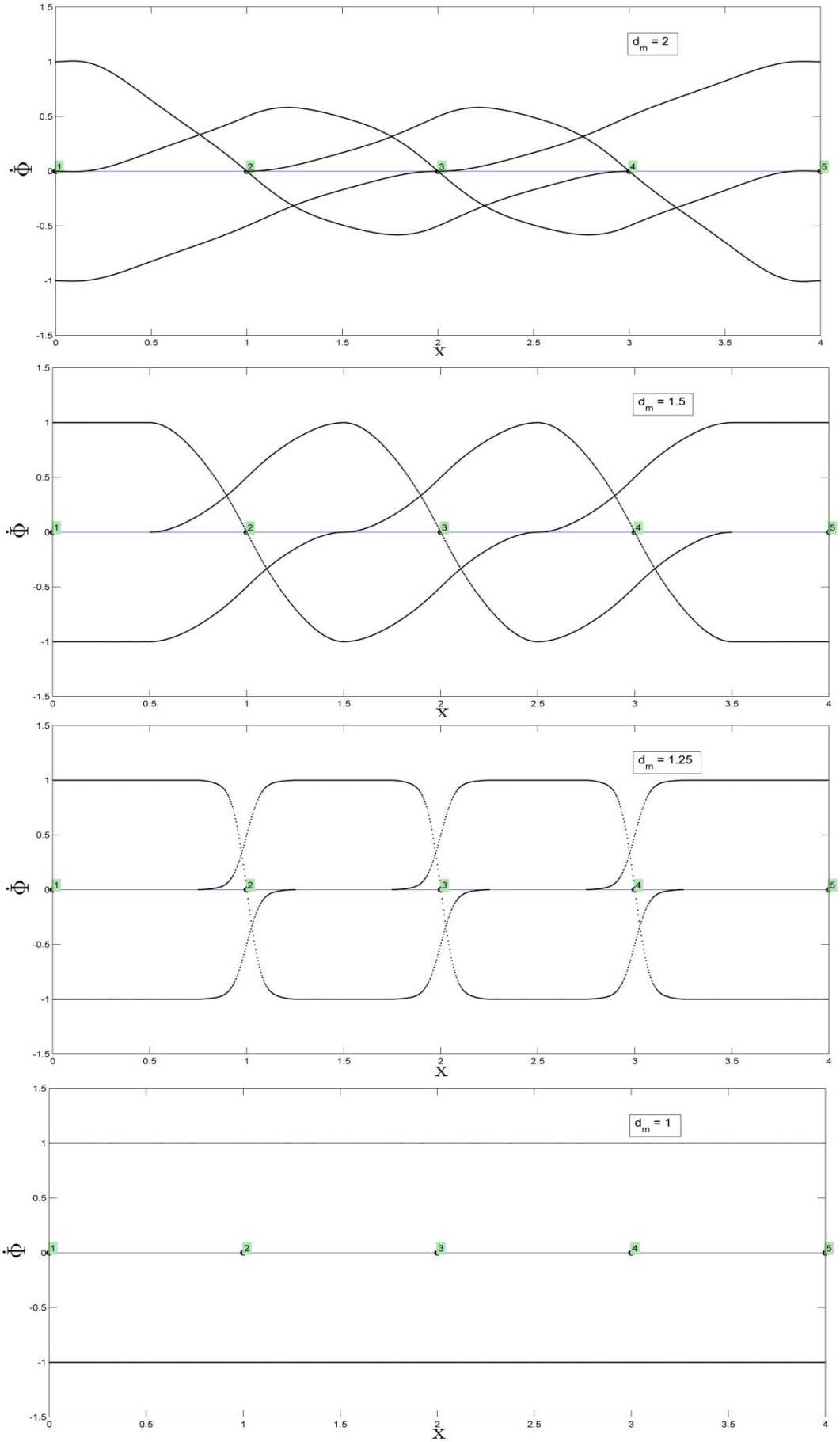


Figure 3.17

First derivative of the shape functions of each node for different sizes of the support domain

To get smooth graphs we computed the shape function, as well as the derivatives in 500 sampling points. Note that dm=1 is referred to these interior sampling points. Hence, only two nodes are located in the support domain of each sampling node at a time, and the MLS approximation is degenerated to interpolation.

3.6 Consistency

Consistency, often referred as reproducibility, in Galerkin methods means the ability of an approximation to reproduce polynomials of a certain order. For FEM the consistency requirement ensures that as the size of the elements tends to zero, the approximation equation will represent the exact differential equation and the boundary conditions for a certain problem. An approximation is called zero order complete (C^0) if it reproduces constant functions exactly. It is called first order complete (C^1) or linear complete if it reproduces exactly linear functions, and so on for higher order of consistency.
In MLS approximation the consistency depends on the complete order of the polynomial basis employed in Eq. (3.5) to Eq.(3.7). If the complete order of the monomials is k, the MLS shape function will possess (C^k) consistency.
The proof of consistency of the MLS approximation is very important, because it ensures that any function that appears in the basis can be exactly reproduced. Making further use of this feature, we can develop shape functions for simulating for example a stress field at a crack tip. This can be easily performed by including singular functions into the basis (Belytschko et al. 1994a).

4 Element Free Galerkin (EFG) Method

The element free Galerkin (EFG) method is a meshless method developed by Belytschko, Lu and Gu (1994). This method only requires a set of nodes and a description of the boundaries to construct an approximate solution. The connectivity between the data point and the shape functions are constructed by the method, without recourse to elements. Although the EFG can be considered meshless with respect to shape function construction or function approximation, a mesh will be required for evaluating partial differential equations by the Galerkin approximation procedure (Background Integration). This is because evaluation of the integrals in the weak form requires a subdivision of the domain unless nodal quadrature is used.
The EFG method employs the *moving least square (MLS) approximation* to fit a cloud of nodes located in the problem domain, which is basically composed of three components as seen in previous chapter: a weight function of compact support associated with each node, a polynomial basis and a set of coefficients that minimize a functional. Weighted *support domains* determine the relative position of each node. The overlap of the nodal support domains defines the nodal connectivity.
In this Chapter principles for creating weak forms for the EFG method are also outlined. For a *2k*th-order differential governing system equation, the Galerkin weak form requires a consistency of only *k*th order, compared with the strong form that calls for a 2*k*th order consistency. This remark gives a good reason for using weak forms. General concepts such as the Hamilton' s principle are presented for completeness, as well as methods of imposing boundary conditions.

4.1 Weak Form

4.1.1 Hamilton's Principle and Minimum Potential Energy Principle

Hamilton's Principle is one of the variational principles that is based on the energy principles; it states "Of possible time histories of consistency displacement states which satisfy

- the compatibility conditions
- the essential (displacement or kinematic) boundary conditions
- the conditions at initial (t_1) and final time (t_2)

the history corresponding to the actual solution makes Lagrangian functional a minimum." Mathematically the Hamilton's principle can be written as :

$$\delta \int_{t_2}^{t_1} L dt = 0 \tag{4.1}$$

where L is the Lagrangian functional. For a system of solids and structures, where Ω is the whole volume of the solid and Γ_t stands for the boundaries of the solid where traction stress is imposed, it can be defined as:

$$L = T - \Pi_s + W_f \tag{4.2}$$

where T is the kinetic energy

$$T = \frac{1}{2} \int_{\Omega} \rho \dot{u}^T \dot{u} \, d\Omega \tag{4.3}$$

Π_s is the strain energy

$$\Pi_s = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \, d\Omega \tag{4.4}$$

and W_f is the work done by the external forces

$$W_f = \int_{\Omega} \mathbf{u}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{u}^T \bar{\mathbf{t}} \, d\Gamma \tag{4.5}$$

For static problems we can simply use the *Principle of Minimum Potential Energy* to simplify obtaining equations of equilibrium; it states "For conservative structural systems, of all the kinematically admissible deformations, those corresponding to the equilibrium state extremize (i.e., minimize or maximize) the total potential energy. If the extremum is a minimum, the equilibrium state is stable."

For this principle the kinetic energy is equal to zero (static problem) and the Eq. (4.1) and Eq.(4.2) are simplified as follows:

$$\delta \Pi = 0 \tag{4.6}$$

where Π is the Lagrangian functional now modified as the total potential energy :

$$\Pi = \Pi_s - W_f \tag{4.7}$$

4.1.2 Galerkin Weak Form

The Galerkin Weak Form can be directly derived for static problems using the Minimum Potential Energy Principle. By using Eq. (4.6) and Eq. (4.7) we take :

$$\delta \int_{\Omega} \frac{1}{2} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \, d\Omega - \delta \left[\int_{\Omega} \mathbf{u}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{u}^T \bar{\mathbf{t}} \, d\Gamma \right] = 0 \tag{4.8}$$

Moving the variation operation into the integral operations, we obtain :

$$\int_{\Omega} \frac{1}{2} \delta \left(\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \right) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} \, d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} \, d\Gamma = 0 \tag{4.9}$$

The integrant in the first integral term can be written as follows using the chain rule of variation:

$$\delta \left(\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \right) = \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} + \boldsymbol{\varepsilon}^T \delta \boldsymbol{\sigma} \tag{4.10}$$

The two terms in the foregoing equation are all scalars, and therefore their transposes are still themselves. For the Eq. (4.10) we have:

$$\boldsymbol{\varepsilon}^T \delta \boldsymbol{\sigma} = \left(\boldsymbol{\varepsilon}^T \delta \boldsymbol{\sigma} \right)^T = \delta \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} \tag{4.11}$$

By using the constitutive Eq. (2.8) and the symmetric property of the matrix of the material constants **c**, we take:

$$\delta \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} = \delta \left(\mathbf{c} \boldsymbol{\varepsilon} \right)^T \boldsymbol{\varepsilon} = \delta \boldsymbol{\varepsilon}^T \mathbf{c}^T \boldsymbol{\varepsilon} = \delta \boldsymbol{\varepsilon}^T \mathbf{c} \boldsymbol{\varepsilon} = \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \tag{4.12}$$

Thus the Eq. (4.10) becomes

$$\delta \left(\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \right) = 2 \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} \tag{4.13}$$

Finally, by replacing the Eq. (4.13), the well known *Galerkin weak form* Eq. (4.9) simply reduces to

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}}\, d\Gamma = 0 \tag{4.14}$$

If we now use the stress-strain relation of Eq. (2.8) and then the strain-displacement relation of Eq. (2.5), the Galerkin weak form Eq. (4.14) for static problems, can be expressed in terms of displacement vector **u** :

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}}\, d\Gamma = 0 \tag{4.15}$$

The above equation is very useful, as it can easily give a set of discretized system equations, if the displacement vector is approximated by a standard method. A numerical method based on the Galerkin weak form looks for a displacement field **u** that satisfies the Eq. (4.15) for any arbitrary **δu**.

4.2 Constrained Weak Form

There are some cases when the field function that approximates the displacement vector, does not satisfy the compatibility and boundary conditions, on parts of the problem domain. To handle this problem, the Lagrangian functional, used in Hamilton's Principle and Minimum Potential Energy Principle, has to be modified. The purpose for this modification is to constrain the Lagrangian functional, and seek for a stationary point, after some certain constraints have been imposed. There are basically two methods used to modify the functional, *the Lagrange multipliers method* and the *penalty method.*

4.2.1 Method of Lagrange Mulripliers

In the method of Lagrange multipliers, the modified Lagrangian is written in the following form:

$$\tilde{L} = L + \int_{\Omega} \boldsymbol{\lambda}^T \mathbf{C}(\mathbf{u})\, d\Omega \tag{4.16}$$

where **λ** is a vector of Lagrange multipliers, that are unknown functions of independent coordinates in the domain Ω, given by:

$$\boldsymbol{\lambda}^T = \{\lambda_1 \;\; \lambda_2 \;\; \dots \;\; \lambda_k\} \tag{4.17}$$

and **C** is a given matrix of coefficients written as:

$$\mathbf{C}(\mathbf{u}) = \begin{Bmatrix} C_1(\mathbf{u}) \\ C_2(\mathbf{u}) \\ \vdots \\ C_k(\mathbf{u}) \end{Bmatrix} \tag{4.18}$$

Using the new functional, and the Eq. (4.1), the modified Hamilton's principle can be written as:

$$\delta \int_{t_2}^{t_1} \tilde{L} dt = 0 \tag{4.19}$$

Respectively the modified Minimum Potential Energy principle for static problems can be written as:

$$\delta \tilde{\Pi} = 0 \tag{4.20}$$

where

$$\tilde{\Pi} = \Pi + \int_{\Omega} \boldsymbol{\lambda}^T \mathbf{C}(\mathbf{u}) \, d\Omega \tag{4.21}$$

As mentioned above, Lagrange multipliers are unknown functions. In the process of seeking discretized system equations, these Lagrange multipliers have to be approximated in a manner similar to field functions. It is obvious therefore, that the Lagrange multiplier method is introducing additional unknowns to the problem. This is the main drawback to this method, however by solving the system of discretized equations, the Lagrange multipliers will rigorously enforce the constrains.

By applying the Lagrange multipliers method to the Galerkin Weak form for static problems seen in Eq (4.15) we can get a constrained weak form written as:

$$\int_{\Omega} \delta(\mathbf{Lu})^T \, \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} \, d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}} \, d\Gamma - \int_{\Omega} \delta \boldsymbol{\lambda}^T \mathbf{C}(\mathbf{u}) \, d\Omega - \int_{\Omega} \boldsymbol{\lambda}^T \delta \mathbf{C}(\mathbf{u}) \, d\Omega = 0 \tag{4.22}$$

4.2.2 Penalty Method

To describe the Penalty method we will consider a functional given as:

$$\mathbf{C}^T\boldsymbol{\alpha}\mathbf{C}=\alpha_1 C_1^{\,2}+\alpha_2 C_2^{\,2}+\ldots+\alpha_\kappa C_\kappa^{\,2} \tag{4.23}$$

where $\boldsymbol{\alpha}$ is a diagonal matrix given by

$$\boldsymbol{\alpha}=\begin{bmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & a_k \end{bmatrix} \tag{4.24}$$

and $\alpha_1, \alpha_2, \ldots, \alpha_k$ are penalty factors, usually assigned positive constant numbers, and $\mathbf{C}$ is a given matrix of coefficients as seen in Eq. (4.18)

The modified Lagrangian is written in the following form:

$$\tilde{L}=L+\frac{1}{2}\int_\Omega \mathbf{C}^T(\mathbf{u})\boldsymbol{\alpha}\mathbf{C}(\mathbf{u})\,d\Omega \tag{4.25}$$

Using the new functional, and the Eq. (4.1), the modified Hamilton's principle can be written as:

$$\delta\int_{t_2}^{t_1}\tilde{L}dt=0 \tag{4.26}$$

Respectively the modified Minimum Potential Energy principle for static problems can be written as:

$$\delta\tilde{\Pi}=0 \tag{4.27}$$

where

$$\tilde{\Pi}=\Pi+\frac{1}{2}\int_\Omega \mathbf{C}^T(\mathbf{u})\boldsymbol{\alpha}\mathbf{C}(\mathbf{u})\,d\Omega \tag{4.28}$$

Because α is a known constant, there will be no increase in the number of unknowns in the system. However, the question is how to choose the penalty factor. The problem

is that if the penalty factor is too small, the constraints will not be properly enforced, but if it is too large, numerical problems will be encountered. In following subsection we will introduce a formula for choosing the penalty factor. What is more in the penalty method, the constraints are only satisfied *approximately*, opposed to Lagrange multiplier method.

By applying the Penalty method to the Galerkin Weak form for static problems seen in Eq (4.15) we can get a constrained weak form written as:

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta\mathbf{u}^T\mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T\bar{\mathbf{t}}\, d\Gamma - \int_{\Omega} \delta\mathbf{C}^T(\mathbf{u})\boldsymbol{\alpha}\mathbf{C}(\mathbf{u})\, d\Omega = 0 \qquad (4.29)$$

4.3 EFG with Penalty Method

4.3.1 Formulation

We consider a two dimensional (2D) linear problem on the domain Ω bounded by Γ. The partial differential equation for this problem can be written in the form :

$$\mathbf{L}^T\boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega \qquad (4.30)$$

while the boundary conditions are given as follows:

$$\boldsymbol{\sigma}\mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \qquad (4.31)$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u \qquad (4.32)$$

where

L = the differential operator defined by Eq. (2.7) for three-dimensional (3D) solids and Eq. (2.18) for 2D solids.
σ = the stress tensor defined by Eq. (2.2) for 3D case and Eq. (2.13) for 2D case.
u = the displacement vector given by Eq. (2.6) for 3D solids and by Eq. (2.17) for 2D solids.
b = the body force vector
$\bar{\mathbf{t}}$ = the traction displacement imposed on the natural stress boundaries Γ_t
$\bar{\mathbf{u}}$ = the displacement imposed on the essential displacement boundaries Γ_u
n = the vector of unit normal on the natural boundary

As refereed in the EFG method, the problem domain Ω is described by a set of nodes scattered in the problem domain and on the boundaries of the domain. The MLS procedure is employed to approximate the displacement field, using the nodal parameters of the displacement at the nodes that fall into the support domain of a point.

We will introduce a penalty factor to penalize the difference between the displacement of MLS approximation and the prescribed displacement on the essential boundary Eq. (4.32). By using Eq. (4.29) the constrained Galerkin weak form for static problems using the penalty method, can therefore be written as :

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta \mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta \mathbf{u}^T \bar{\mathbf{t}}\, d\Gamma - \delta \int_{\Gamma_u} \frac{1}{2} (\mathbf{u} - \bar{\mathbf{u}})^T \boldsymbol{\alpha} (\mathbf{u} - \bar{\mathbf{u}})\, d\Gamma = 0 \quad (4.33)$$

In the fourth term of the Eq. (4.29) the area integral has been modified into curve integral, because the essential boundary conditions are defined only on the boundary.

The MLS approximation described in Chapter 3 is now used to express the trial function at any point of interest **x**. For the displacement component u ,we have :

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \varphi_I(\mathbf{x}) u_I \quad (4.34)$$

where n is the number of nodes used in the support domain of the point, for constructing the shape function $\varphi_I(x)$.

Respectively for the displacement component v, we have :

$$v^h(\mathbf{x}) = \sum_{I=1}^{n} \varphi_I(\mathbf{x}) v_I \quad (4.35)$$

Combining Eq. (4.34) and Eq. (4.35), we obtain

$$\mathbf{u}^h = \begin{Bmatrix} u \\ v \end{Bmatrix}^h = \sum_{I=1}^{n} \underbrace{\begin{bmatrix} \varphi_I & 0 \\ 0 & \varphi_I \end{bmatrix}}_{\boldsymbol{\Phi}_I} \underbrace{\begin{Bmatrix} u_I \\ v_I \end{Bmatrix}}_{\mathbf{u}_I} = \sum_{I=1}^{n} \boldsymbol{\Phi}_I \mathbf{u}_I \quad (4.36)$$

where $\boldsymbol{\Phi}_I$ is the matrix of shape functions.

By using Eq. (4.36) and Eq. (2.18) for the differential operator matrix, $\mathbf{Lu}^h$, that expresses the strains, becomes

$$\mathbf{Lu}^h = L\sum_{I=1}^{n}\mathbf{\Phi}_I\mathbf{u}_I = \sum_{I=1}^{n}\mathbf{L}\mathbf{\Phi}_I\mathbf{u}_I = \sum_{I=1}^{n}\begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x}\end{bmatrix}\begin{bmatrix}\varphi_I & 0\\ 0 & \varphi_I\end{bmatrix}\mathbf{u}_I = \sum_{I=1}^{n}\underbrace{\begin{bmatrix}\varphi_{I,x} & 0\\ 0 & \varphi_{I,y}\\ \varphi_{I,y} & \varphi_{I,x}\end{bmatrix}}_{\mathbf{B}_I}\mathbf{u}_I = \sum_{I=1}^{n}\mathbf{B}_I\mathbf{u}_I \tag{4.37}$$

where $\varphi_{I,x}$ and $\varphi_{I,y}$ represent the partial derivatives of the MLS shape function with respect to x and y. $\mathbf{B}_I$ is called the *strain matrix* for node *I*.

Substituting the Eq. (4.37) into the weak form for static problems of Eq. (4.33) , and after some reductions , we arrive at the final system equation of

$$\left[\mathbf{K}_{total}\right]\mathbf{U} = \mathbf{F}_{total} \tag{4.38}$$

or

$$\left[\mathbf{K} + \mathbf{K}^a\right]\mathbf{U} = \mathbf{F} + \mathbf{F}^a \tag{4.39}$$

Vector **F** is the *global force vector,* which collects force vectors at all the nodes in the problem domain and has the form :

$$\mathbf{F} = \begin{Bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \vdots\\ \mathbf{f}_n\end{Bmatrix} \tag{4.40}$$

where $\mathbf{f}_I$ is the nodal force vector at node I calculated by

$$\mathbf{f}_I = \int_{\Gamma_t}\varphi_I\bar{\mathbf{t}}\,d\Gamma + \int_{\Omega}\varphi_I\mathbf{b}\,d\Omega \tag{4.41}$$

The Matrix **K** of Eq. (4.39) is known as the *global stiffness matrix* which is assembled using the *nodal stiffness matrix* calculated as follows

$$\mathbf{K}_{IJ} = \int_{\Omega}\mathbf{B}_I^T\mathbf{c}\,\mathbf{B}_J\,d\Omega \tag{4.42}$$

The nodal stiffness contains the stiffness coefficients between nodes I and J evaluated at a point in the problem domain. It is a function of coordinates that need to be numerically integrated over the entire problem domain. In the EFG method, we will use the background integration for this purpose. The $\mathbf{K}_{IJ}$ exists as long as the nodes I and J are both covered by the support domain of at least one quadrature point. Otherwise the $\mathbf{K}_{IJ}$ vanishes and there is no need to be computed.

The additional matrix $\mathbf{K}^a$ is the *global penalty matrix* which is assembled using the nodal penalty matrix calculated by

$$\mathbf{K}_{IJ}^{a} = \int_{\Gamma_u} \mathbf{\Phi}_I{}^T \boldsymbol{\alpha} \mathbf{\Phi}_J \, d\Gamma \tag{4.43}$$

where $\mathbf{\Phi}_I$ is the matrix of shape functions Eq. (4.36)

The vector $\mathbf{F}^a$ is a force vector caused by the imposition of the boundary conditions. The nodal force vector has the form of

$$\mathbf{F}_{I}^{a} = \int_{\Gamma_u} \mathbf{\Phi}_I{}^T \boldsymbol{\alpha} \bar{\mathbf{u}} \, d\Gamma \tag{4.44}$$

As we can see from Eq. (4.43), due to the fact that α is a known constant, the degrees of freedom, and as a result the number of unknowns in the system, are not increasing. Furthermore the symmetry and the bandedness of the system matrix are preserved.
These advantages make the penalty method much more efficient and hence much more attractive and easy to use. These are clearly the reasons why, penalty method will be used in this diploma thesis, for the imposition of the essential boundary conditions.

4.3.2 Determination of Penalty Factor

We have already recognized the problem of choosing a proper penalty factor for our problems. Much effort has been made, trying to introduce a universal formula applicable to any problem. However we think that one should always determine the factors that affect the selection of the penalty factor, and are different for each problem, such as Young's modulus or the load case for static problems. Some kind of formula was proposed by Zienkiewicz for the FEM analysis :

$$\alpha = constant\left(1/h\right)^n \tag{4.45}$$

where h is the characteristic length, which can be the ratio of the element size to the dimension of the problem domain, and *n* is the order of the elements. The constant should relate to the material property of the solid structure.

G.R. Liu in his book Mesh Free Methods-Moving Beyond the Finite Element Method(2003), introduces a simple method for determining the penalty factor:

$$\alpha = 1.0 \text{ x } 10^{4-13} \text{ x } \max(\textit{diagonal elements in the stifness matrix}) \tag{4.46}$$

or

$$\alpha = 1.0 \text{ x } 10^{5-8} \text{ x } \text{ Young's modulus} \tag{4.47}$$

This method has been adopted, and works well for most of the examples that are presented in this diploma thesis.

4.3.3 Background Integration

From the above equations Eq. (4.41)-(4.44) it is obvious that there is need to integrate over the problem domain, for both natural and essential boundaries. Due to the fact that the analytical solution for the curve integrals is not feasible, numerical techniques are applied. The Gauss quadrature scheme, common in the FEM approximations, is often used. This method can be described as:

$$\int_{\Omega} f(\mathbf{X}) \, \mathrm{d}\Omega = \sum_{J} f(\xi_J) \, \mathrm{w}_J \det J^{\xi}(\xi) \tag{4.48}$$

where ξ are local coordinates and $\det J^{\xi}(\xi)$ is the determinant of the Jacobian, illustrating the mapping from the parent into the physical domain.

In the use of this numerical quadrature scheme, a mesh of integration shells is required. This mesh of cells is called a *background mesh.* In contrast to FEM, the background mesh in EFG method is used only for integration reasons, and not for field variable interpolation. What is more, the position of the cell structure, and as a result the position of the Gauss points, is independent of the nodes position as shown in Figure 4.1 (b). However, if the nodes of the EFG method are positioned in the problem domain in a structured way, the background mesh may follow this pattern as in conventional FEM-Figure 4.1 (a).

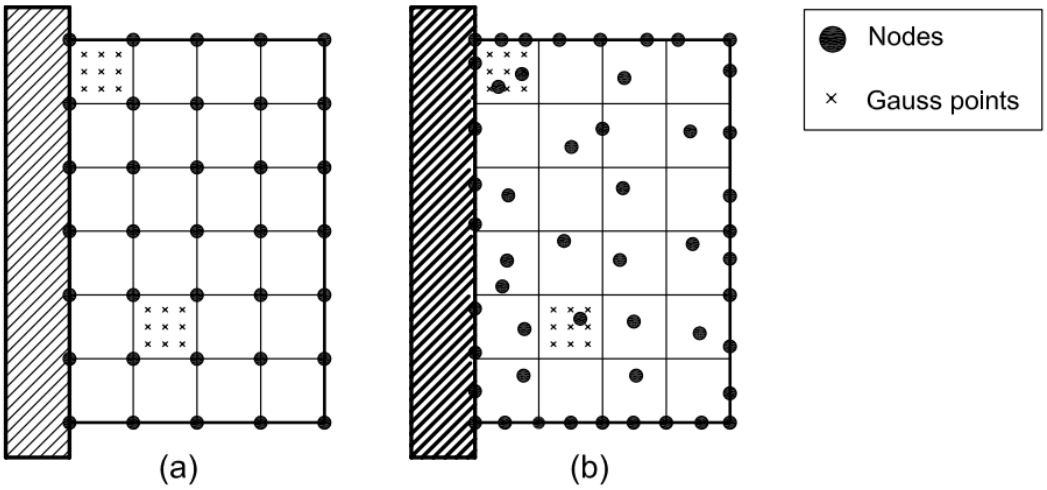


Figure 4.1
Integration in EFG method for a rectangular plate. Background mesh
(a) regular nodes distribution (b) arbitrary nodes distribution.

Finally a question comes for the minimum number of the Gauss points that are needed for a proper integration. For a 2D problem, the number of unknown variables N_u should be

$$N_u = 2 \text{ x } n_t - 2 \text{ x } n_f \tag{4.49}$$

where n_t is the number of nodes in the domain Ω, and n_f is the number of the constrained nodes.
In evaluating the integral at each integration point, three independent strain relations are used. Therefore the number of independent equations used to solve the system matrix is

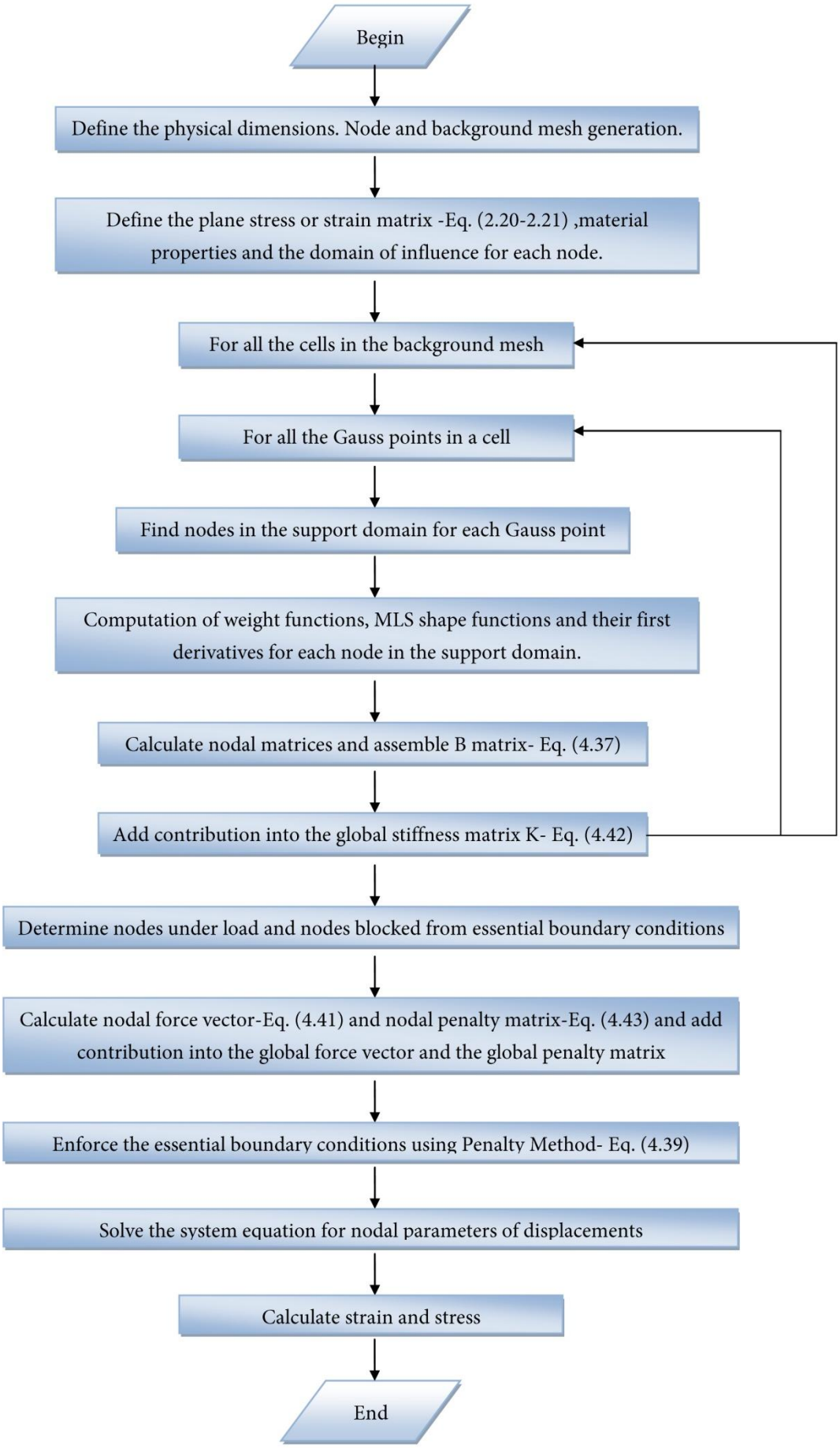
$$N_Q = 3 \text{ x } n_Q \tag{4.50}$$

where n_Q is the total number of Gauss points in domain Ω. Therefore to avoid singularity, the number of independent equations N_Q must be larger than the number of unknown variables N_u which means

$$N_Q > N_u \approx 2n_t \Rightarrow n_Q > \frac{2}{3} n_t \quad , \textit{for 2D problems} \tag{4.51}$$

This requirement is necessary, but not always a sufficient one. In most of the examples solved for this diploma thesis the number of the quadrature points is more than the minimum requirement, to ensure more accurate solutions, and smoother stresses.

4.4 Algorithm Description



5 The Cantilever Beam Problem

In this Chapter, we will introduce a common two dimensional linear solid mechanics problem, solved using the EFG method.
Consider a cantilever beam of dimension *L x D*, subjected to a parabolic load at the free end shown in Figure 5.1. Timoshenko and Goodier (1977) have given an exact analytical solution of this problem that will be compared with the numerical solution calculated by the EFG method.

The parabolic traction is given by

$$t_y(y) = -\frac{P}{2I}\left(\frac{D^2}{4} - y^2\right) \tag{5.1}$$

where $I = D^3/12$ is the moment inertia (second moment of area) .The exact displacement solution for this problem is

$$u_x(x,y) = -\frac{P}{6EI}\left[(6L-3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right] \tag{5.2}$$

$$u_y(x,y) = \frac{P}{6EI}\left[3\nu y^2(L-x) + (4+5\nu)\frac{D^2 x}{4} + (3L-x)x^2\right] \tag{5.3}$$

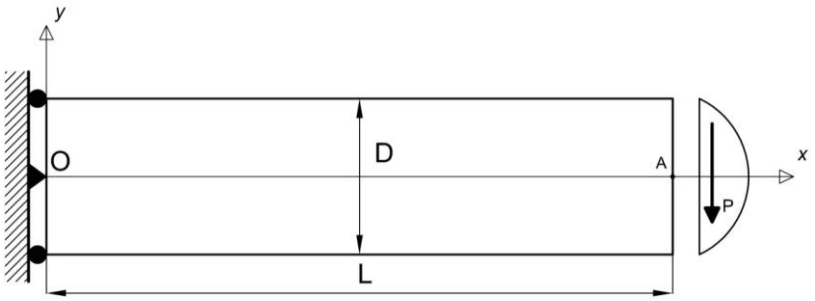


Figure 5.1
Cantilever Beam

The normal stress on the cross section of the beam is

$$\sigma_x = \frac{P(L-x)}{I}y \tag{5.4}$$

The normal stress in the y direction is

$$\sigma_y = 0 \tag{5.5}$$

The shear stress on the cross section of the beam is

$$\sigma_{xy} = -\frac{P}{2I}\left[\frac{D^2}{4} - y^2\right] \tag{5.6}$$

In this example, the parameters for this cantilever beam are taken as follows.

Loading	P=1000 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Height of the beam	D=12m
Length of the beam	L=48m

Table 5-1
Parameters for the cantilever beam

We check the error in the energy and the displacement norm. The energy norm is given by

$$e_{energy} = \left[\frac{1}{2}\int_{\Omega}(\boldsymbol{\varepsilon}_{num} - \boldsymbol{\varepsilon}_{exact})^T \mathbf{D}(\boldsymbol{\varepsilon}_{num} - \boldsymbol{\varepsilon}_{exact})\, d\Omega\right]^{1/2} \tag{5.7}$$

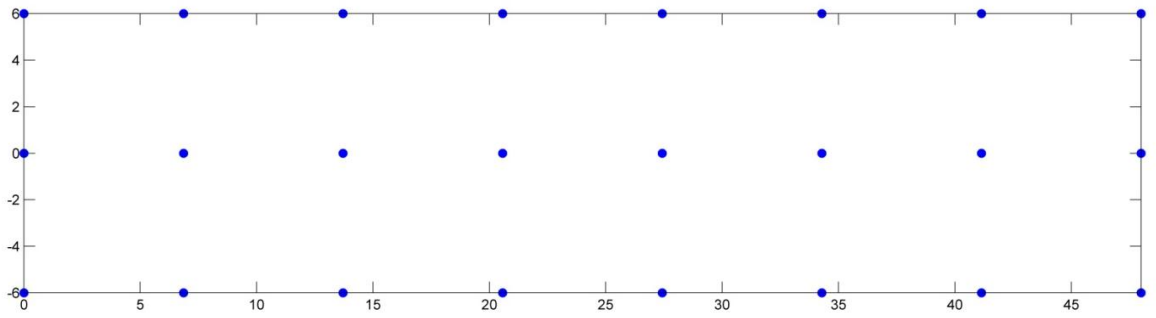
and the displacement norm is given by

$$e_{displacement} = \left[\int_{\Omega}\left[(\mathbf{u}_{num} - \mathbf{u}_{exact})^T(\mathbf{u}_{num} - \mathbf{u}_{exact})d\Omega\right]\right]^{1/2} \tag{5.8}$$

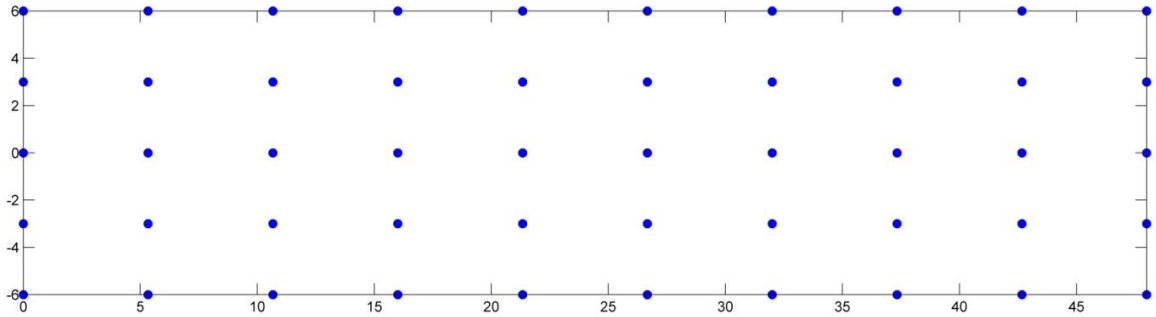
where $\boldsymbol{\varepsilon}_{num}$ and $\boldsymbol{\varepsilon}_{exact}$ are the numerical strain vector and the exact strain vector, respectively. The same symbolism goes for the displacement vectors $\mathbf{u}_{num}$ and $\mathbf{u}_{exact}$.

For showing the rate of convergence in EFG method, the regular node distribution was used. After that, both regular and irregular arrangement of nodes are used, to introduce that there is no need for a structured distribution of the nodes in the problem domain.
Background mesh is used for numerical integration of the weak form. In each Gauss cell 2x2 Gauss quadrature is used. A linear basis and cubic spline weight function are used for the MLS approximation. The penalty factor was chosen to be a=10^{10}.

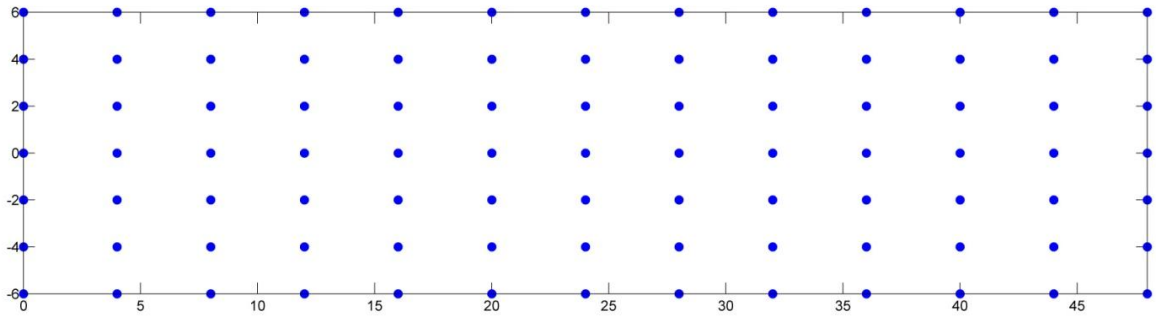
5.1 Regular node distribution



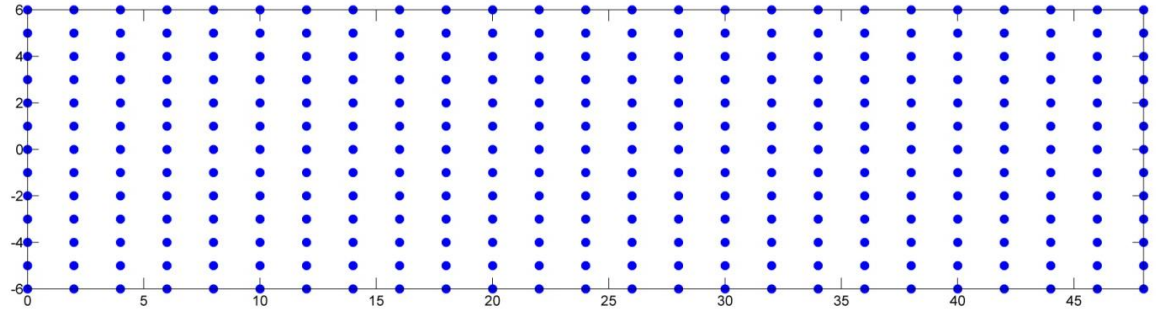
(a) 8x3



(b) 10x5



(c) 13x7



(d) 25x13

Figure 5.2: Regular nodes distribution used for the cantilever beam problem

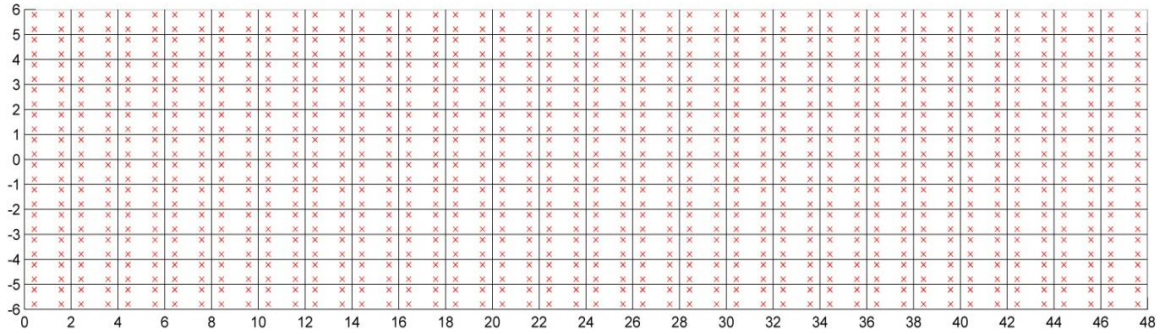


Figure 5.2
Mesh used for integration (24x12) with 2x2 Gauss points in each cell

In Table 5-2 , the vertical displacement at the point A (L,0) on the beam at Figure 5.1 is compared with the exact solution given in Eq. (5.2). The calculation has been done with the same Gauss quadrature as shown in Figure 5.2, and the node distributions are 8x3,10x5, 13x7, 25x13. Rectangular support domain of 2.6 times the nodal spacing is employed.
Figure 5.3 illustrates the comparison of the analytical and the mesh free numerical solutions for the deflection of cantilever beam in each case. A very good agreement is observed. Some minor errors can be explained in the use of the penalty method to impose the boundary conditions.

Nodes	u_y exact	u_y EFG	Error (%)
8 x 3	-0,008900	-0,008115	8,82
10 x 5	-0,008900	-0,008811	1,00
13 x 7	-0,008900	-0,008872	0,32
25 x 13	-0,008900	-0,008897	0,03

Table 5-2
Comparison of vertical displacement at the end of the beam (point A)

Figures 5.4 and 5.5 compare the stresses calculated using the analytical solution and the EFG with penalty method. The normal stress σ_x at the cross section of x= L/2 is shown on Figure 5.4 compared with the corresponding analytical solution of Eq. (5.4). The shear stress σ_{xy} is shown in Figure 5.5. A good agreement is observed as the density of the nodes in the problem domain is increasing. However errors in stress between the exact solution and the numerical results are more evident. As we have already seen, the stresses are obtained using the derivatives of the displacement field. This fact makes the stresses very sensitive to previous errors (propagation of error), and to the way the integration is performed. Finer mesh and more Gauss points have to be used for a more accurate stress field.

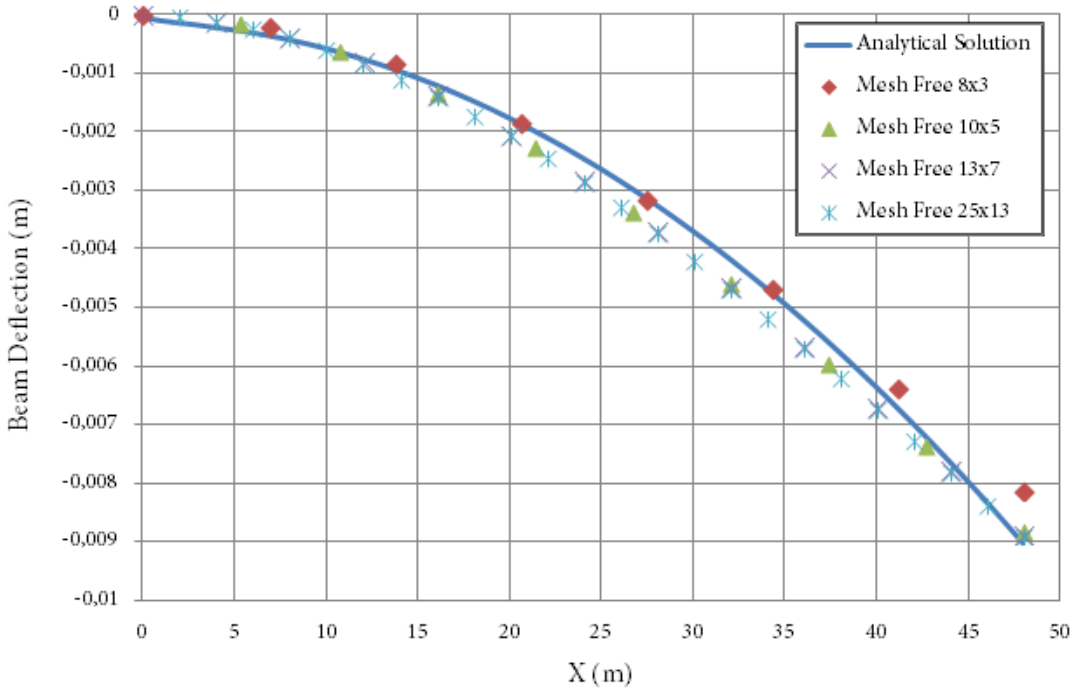


Figure 5.3
Analytical and numerical solutions for the deflection of the cantilever beam (y=0).

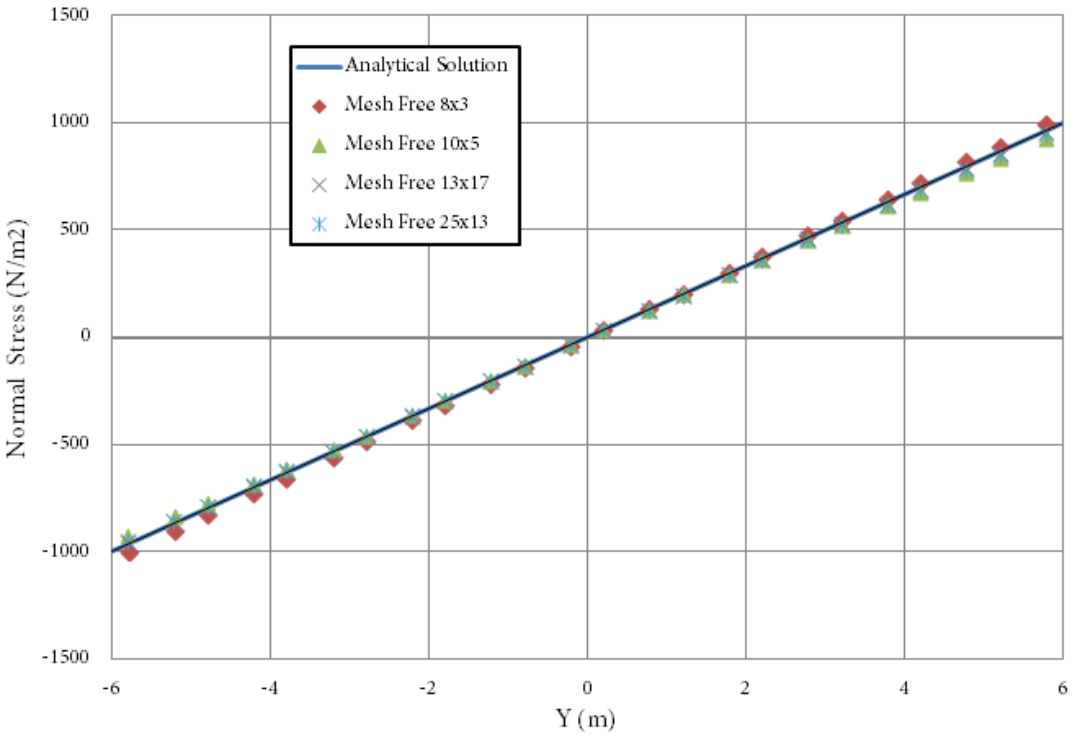


Figure 5.4
Analytical and numerical solutions for the normal stress at the section x= L/2 of the cantilever beam.

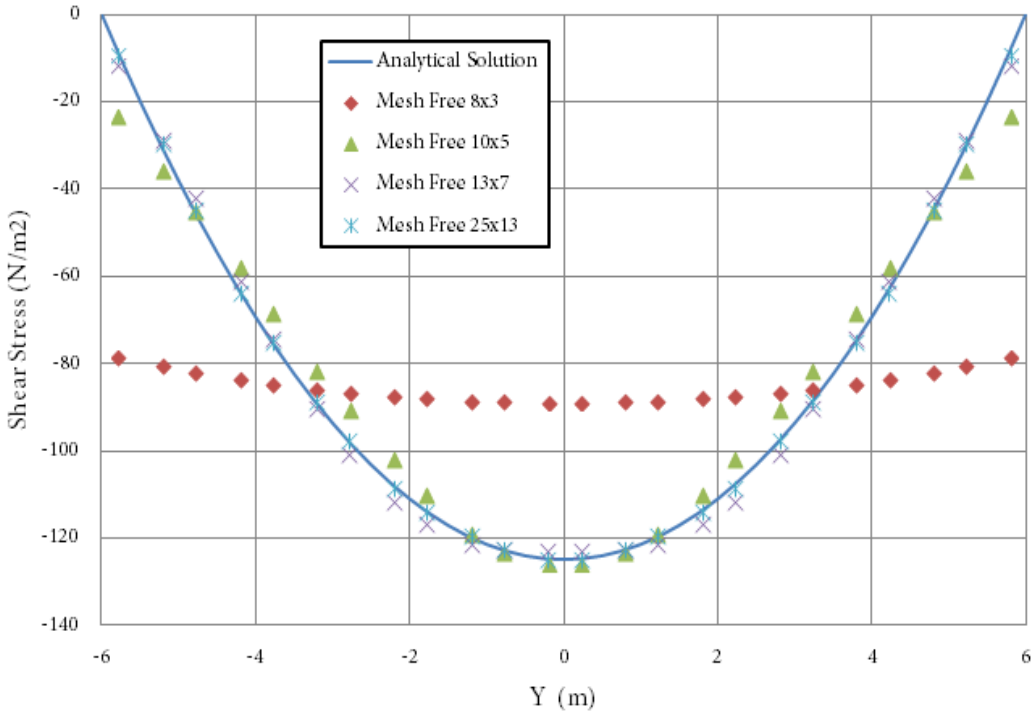


Figure 5.5
Analytical and numerical solutions for the shear stress at the section x= L/2 of the cantilever beam.

To investigate how the node number affects the accuracy of the results, the strain energy error is calculated for the cantilever beam, using different nodal densities. The background mesh is fixed at 24x12 and 2x2 Gauss quadrature is used.

Figure 5.6 is a plot of the rate of convergence in energy error for the beam problem. The value *h* was chosen to be the horizontal nodal spacing in the model. The rate of convergence in energy is calculated using Equation (5.7). The slope of the line plotted in Figure 5.6 is approximately 2.12 which is a greater rate of convergence than the linear finite elements, "which should be around 1.0 (Hughes, 1987) for the same definition of error." [G.R. Liu - Mesh Free Methods-Moving Beyond the Finite Element Method(2003)].

Nodes	DOF	Energy Error (e_{energy})(%)
8 x 3	48	21,4
10 x 5	100	8,4
13 x 7	182	4,0
25 x 13	650	1,6

Table 5-3
Energy errors for different node distributions

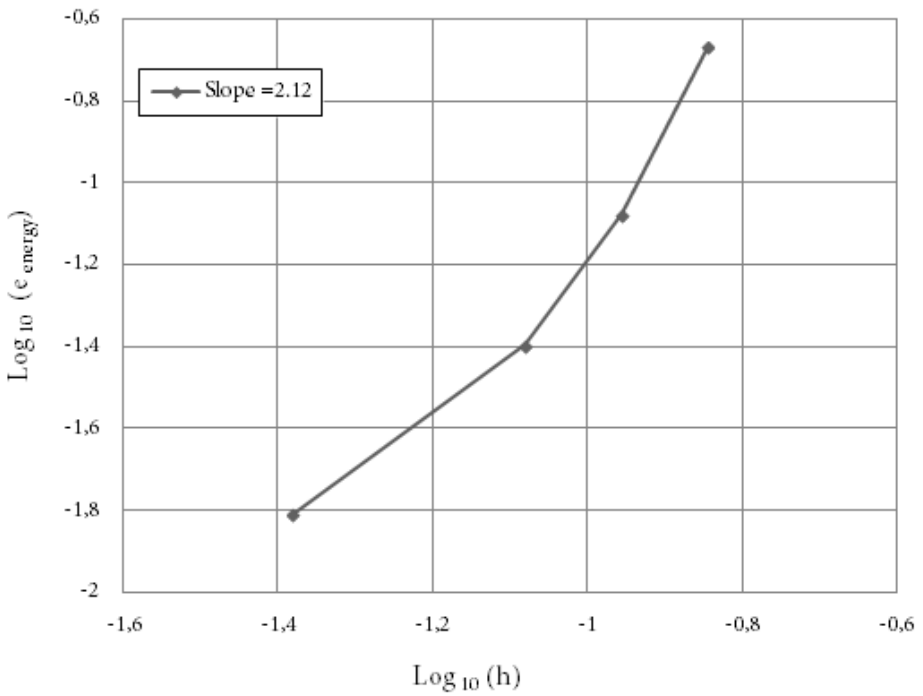


Figure 5.6
Rate of convergence in energy error tested on the cantilever beam

Figure 5-7 also illustrates the rate of convergence in energy error related with the degrees of freedom of the problem. As the degrees of freedom are increasing, the numerical solution becomes more accurate and tends to the analytical solution. The results of this process are also shown in Table 5-3.

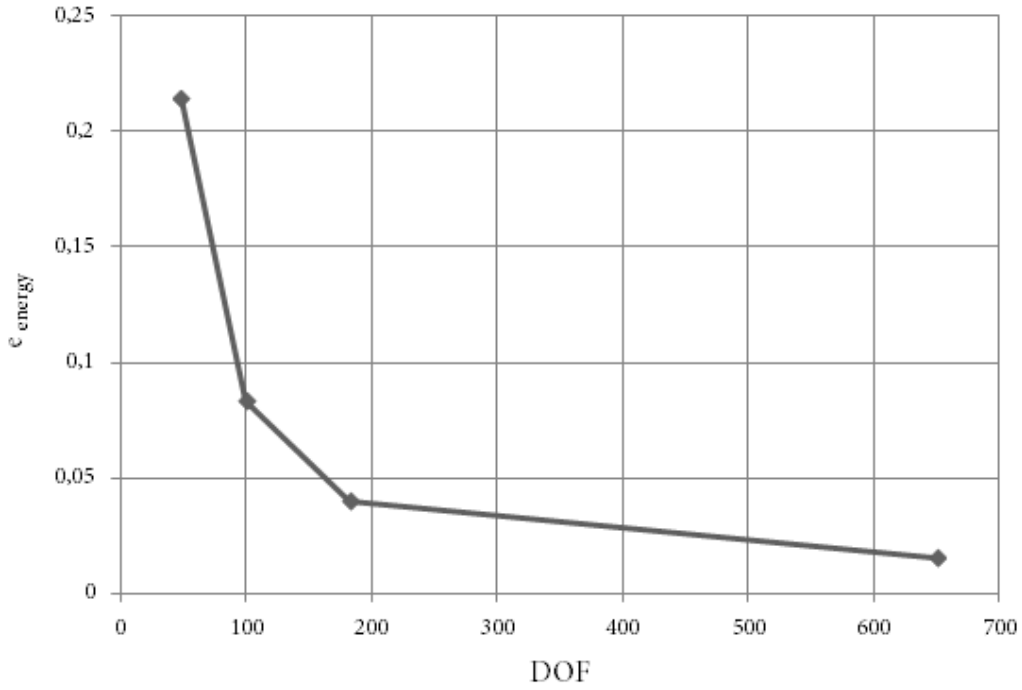


Figure 5.7
Rate of convergence in energy error related with the degrees of freedom

5.2 Irregular node distribution

Now we consider a cantilever beam of the same characteristics as seen in Table 5-1 subjected to a parabolic traction, at the free end. The beam is of unit thickness and the plane stress is considered. Both regular and irregular arrangement of nodes are used and a regular background mesh for numerical integrations, as shown in Figures 5.9-5.11.For this example rectangular support domain of 3.2 times the nodal spacing is employed to ensure enough nodes in the support domain of each Gauss point, and avoid singularity of the moment matrix in constructing MLS shape functions .

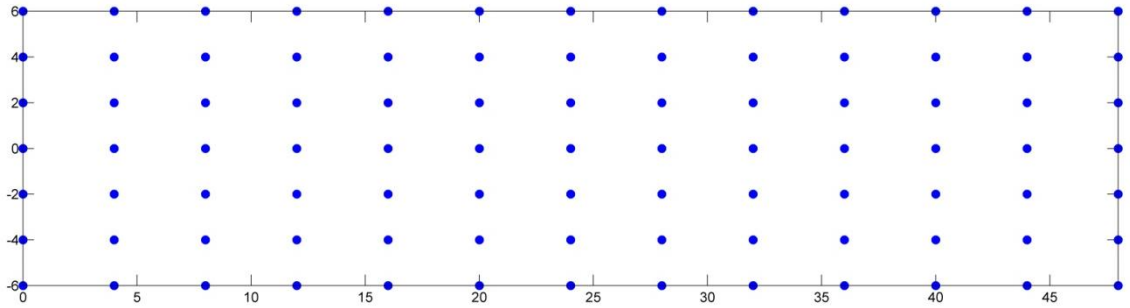


Figure 5.8
Regular node distribution (91 nodes)

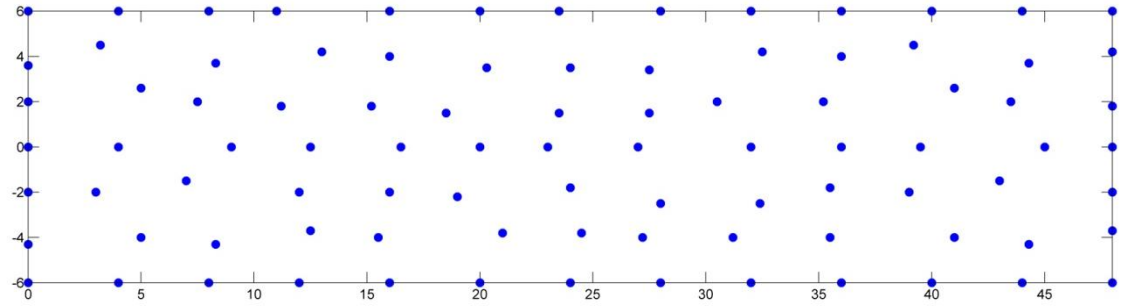


Figure 5.9
Irregular node distribution (91 nodes)

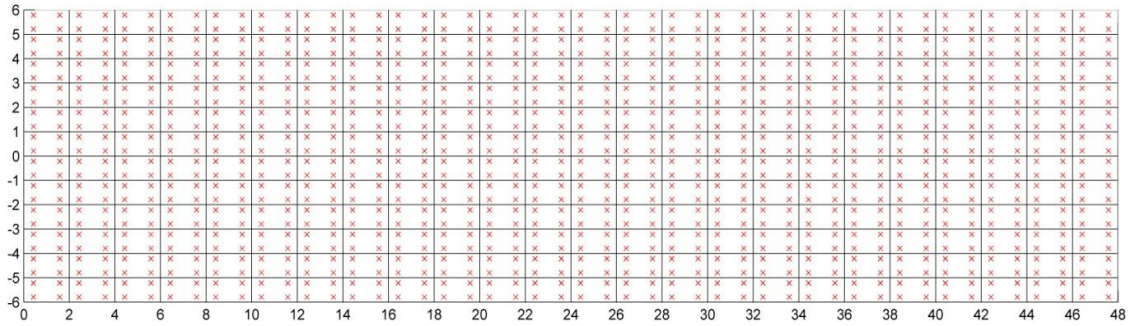


Figure 5.10
Mesh used for integration (24x12) with 2x2 Gauss points in each cell

Figure 5.11 plots again the analytical and the mesh free numerical solutions for the deflection of the cantilever beam. A very good agreement is observed. The exact vertical displacements at the point A (L,0) are also compared in the Table 5-4.

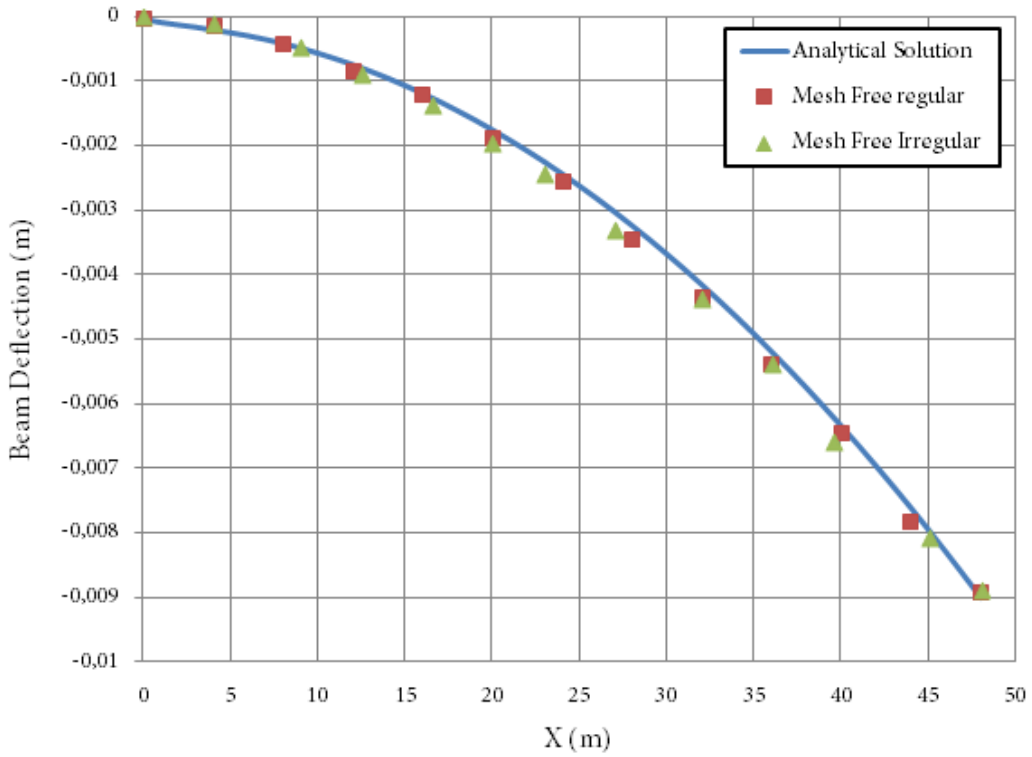


Figure 5.11
Analytical and numerical solutions for the deflection of the cantilever beam (y=0).

Nodes	u_y exact	u_y EFG	Error (%)
91 regular	-0,008900	-0,008889	0,11
91 irregular	-0,008900	-0,008884	0,18

Table 5-4
Comparison of vertical displacement at the end of the beam (point A)

Figures 5.12 to 5.14 compare the stresses calculated using the analytical solution and the EFG with penalty method. The stresses are calculated at the cross section of x= L/2 of the beam.

We can generally notice a good agreement, if we also take into account the fact of the propagation of error in numerical methods for stresses. However we can clearly notice that the accuracy of the shear stress and the normal stress in y direction is lower in the case of the irregular arrangement of nodes, than that in the regular arrangement. This remark makes it clear that a totally arbitrary distribution of nodes can, on the one hand, always give a numerical approximation for the problem domain. On the other

hand a bad dispersion of nodes, can lead to numerical instabilities and relatively large numerical errors.

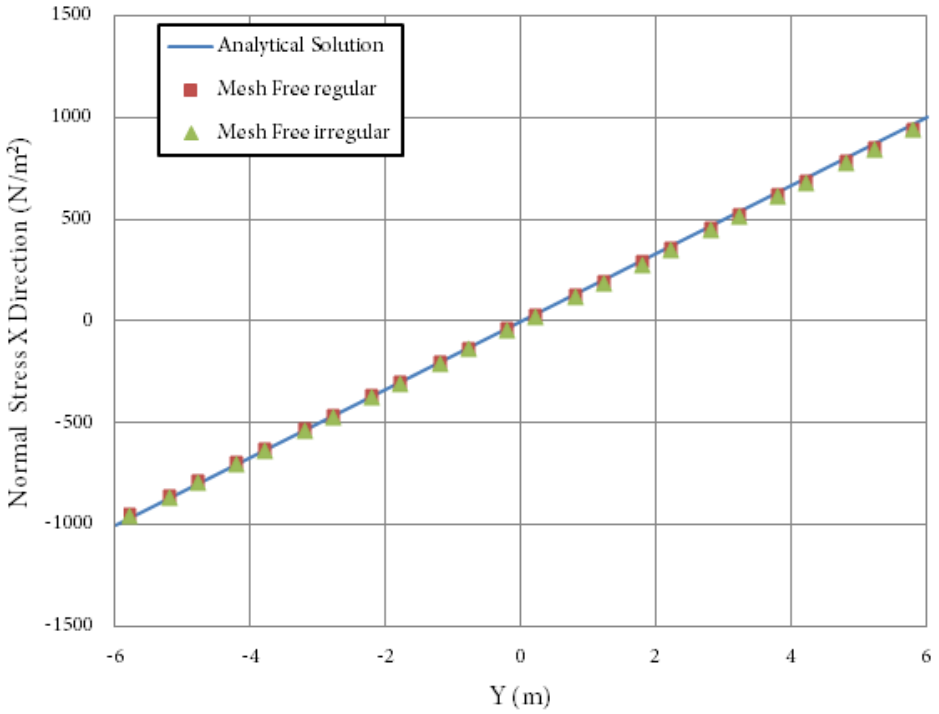


Figure 5.12
Analytical and numerical solutions for the normal stress σ_x at the section x= L/2 of the cantilever beam

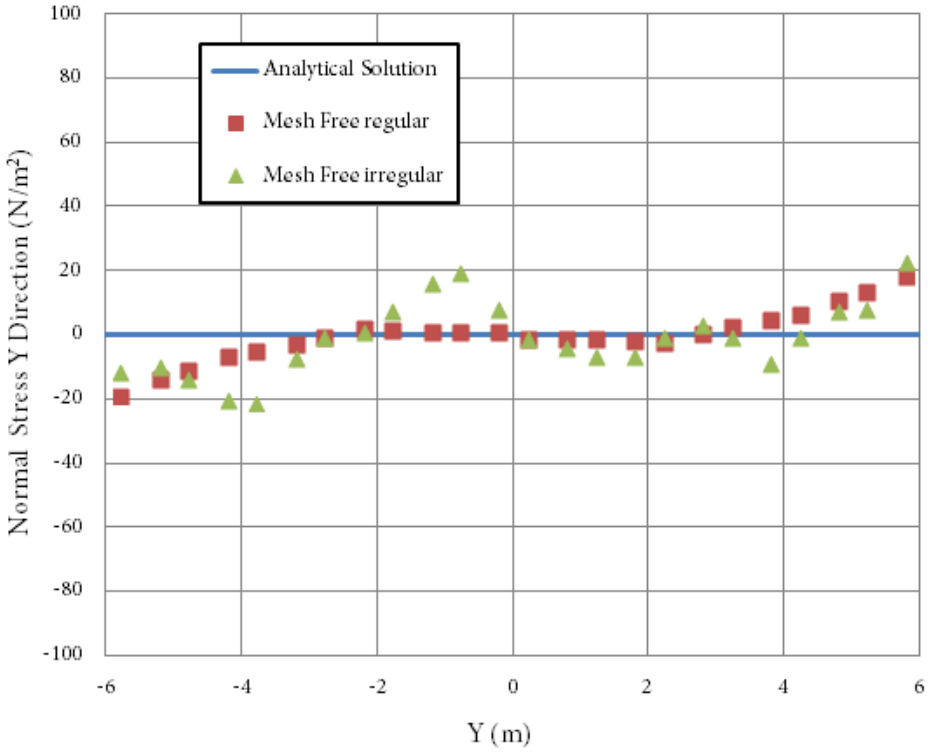


Figure 5.13
Analytical and numerical solutions for the normal stress σ_y at the section x= L/2 of the cantilever beam

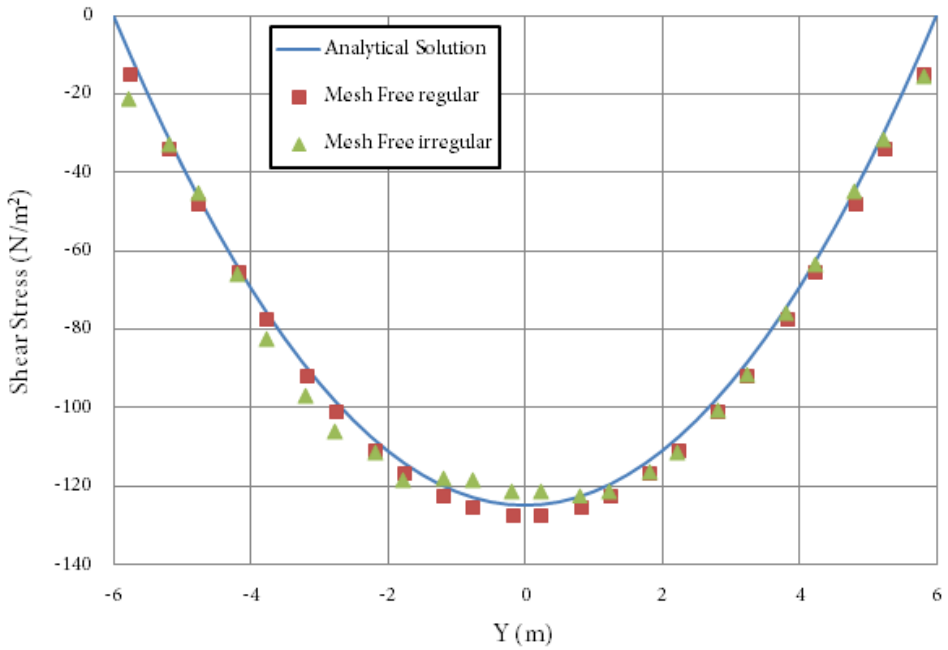


Figure 5.14
Analytical and numerical solutions for the shear stress σ_{xy} at the section x= L/2 of the cantilever beam

6 Hierarchical Concepts for the EFG Method

In previous Chapters we have presented the basic approximations for the EFG method, pointing out the advantages of using this mesh free method for solid mechanics problem calculation. The cantilever beam problem has also been introduced, with a very good agreement of the numerical and analytical results. Therefore the method proposed seems to be a reliable one, with a high rate of convergence. At the same time, as there is no need to create a mesh, the procedure of node creation was completely automated, saving time.
In this Chapter some basic concepts of *hierarchical* forms for the EFG method are proposed. The following approach, carried out for this thesis, shows the flexibility in adding points in mesh free methods, making it easy to perform adaptive schemes and hierarchical solutions. The initial idea of creating hierarchical forms for the EFG method belongs to P.Metsis, Ph.D candidate at Institute of Structural Analysis and Antiseismic Research, School of Civil Engineering, N.T.U.A. An hierarchical refinement method, that can lead to very good results, with low computational cost is introduced in next Chapters.

6.1 Standard and hierarchical forms

As already stated in Chapter 3 the field variable u at any point $\mathbf{x} = (x, y, z)$ within the problem domain is interpolated using the displacements at every node within the *support domain* of the point x, i.e.,

$$u(\mathbf{x},t) = \sum_{I \in S} \Phi_I(x)\, u_I(t) \tag{6.1}$$

where $\Phi_I : \Omega \rightarrow \mathbb{R}$ are the shape functions and the u_I's are the nodal values at principle I located at position x_I, and S is set of nodes I for which $\Phi_I(x) \neq 0$.

The shape functions so defined will be referred to as "standard" ones for a support domain of a point $\mathbf{x} = (x, y, z)$. These shape functions also satisfy the partition of unity, at all nodes of the domain, an important property that can be written as:

$$\sum_{I \in S} \Phi_I(x) = 1 \tag{6.2}$$

A serious drawback however exists, with "standard" shape functions. When nodal arrangement changes, by adding new nodes in the radius of an existing node, new shape functions have to be generated and hence all calculations repeated. This process is time consuming, and should be carried out in order to ensure again the partition of unity.

A more effective domain refinement can be achieved if we hierarchically add new shape functions in the problem domain, while keeping the shape functions of the already existing points constant.

6.1.1 Partition of the shape function

The hierarchical concept is well illustrated by the one-dimensional (elastic bar) problem of Figure 6.1. We consider an interval $0 \le x \le 4$ divided into four classes by five nodes. The expression of the shape functions for each node can be whiten as:

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})[\mathbf{A}(\mathbf{x})]^{-1}\mathbf{B}(x) \tag{6.3}$$

which is evaluated at each Gauss point. The linear basic function $\mathbf{p}^T(x) = [1 \;\; x]$ is used. For a certain Gauss point, the Eq. (6.3) can be written as:

$$\underbrace{\mathbf{\Phi}(\mathbf{x}_{gp})}_{1x5} = \underbrace{\mathbf{p}^T(\mathbf{x}_{gp})}_{[1x2]}\underbrace{[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}}_{[2x2]}\underbrace{\mathbf{B}(x_{gp})}_{[2x5]} \tag{6.4}$$

where

$$\mathbf{p}^T(\mathbf{x}_{gp}) = [1 \quad x_{gp}] \tag{6.5}$$

$$\mathbf{A}_{1:5}(\mathbf{x}_{gp}) = \sum_{i=1}^{5} w(\mathrm{x_{gp}} - \mathrm{x}_i)\mathbf{p}(x_i)\mathbf{p}^T(x_i) = \sum_{i=1}^{5} w(\mathrm{x_{gp}} - \mathrm{x}_i)\begin{bmatrix}1\\ x_i\end{bmatrix}[1 \quad x_i] \tag{6.6}$$

$$\mathbf{B}(x_{gp}) = \mathbf{p}_{gp}\begin{bmatrix} w(x_{gp}-x_1) & w(x_{gp}-x_2) & w(x_{gp}-x_3) & w(x_{gp}-x_4) & w(x_{gp}-x_5)\end{bmatrix} \tag{6.7}$$

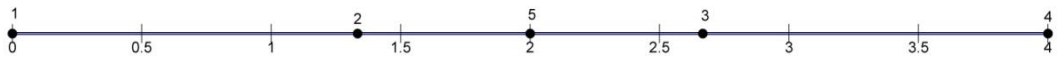


Figure 6.1
1D elastic bar problem

The Eq. (6.4) can be separated into two parts. The first term includes the value of the shape functions for the first four nodes, at a certain Gauss point, and the second one the value of the shape function for the fifth node.

$$\mathbf{\Phi}_{1:5}(\mathbf{x}_{gp}) = \mathbf{\Phi}_{1:4}(\mathbf{x}_{gp}) + \mathbf{\Phi}_{5}(\mathbf{x}_{gp}) = \mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{1:4}(x) + \mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{5}(x) \tag{6.8}$$

The weighted moment matrix **A** can also be separated into two parts, in respective of the nodes in the problem domain, as follows:

$$\mathbf{A}_{1:5}(\mathbf{x}_{gp}) = \mathbf{A}_{1:4}(\mathbf{x}_{gp}) + \mathbf{A}_{5}(\mathbf{x}_{gp}) = \sum_{i=1}^{4} w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_{i})\mathbf{p}(x_i)\mathbf{p}^{T}(x_i) + w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_{5})\mathbf{p}(x_5)\mathbf{p}^{T}(x_5) \tag{6.9}$$

If we now use the *Binomial inverse theorem,* that can help us inverse the weighted moment matrix in parts [Eq. (6.9)], we obtain

$$\left[\mathbf{A}_{1:5}(\mathbf{x}_{gp})\right]^{-1} = \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp}) + \mathbf{A}_{5}(\mathbf{x}_{gp})\right]^{-1} = \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1} + \mathbf{X} \tag{6.10}$$

where

$$\mathbf{X} = -\left\{\mathbf{I} + \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1}\mathbf{A}_{5}(\mathbf{x}_{gp})\right\}^{-1}\left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1}\mathbf{A}_{5}(\mathbf{x}_{gp})\left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1} \tag{6.11}$$

Finally by combining the Eq. (6.10) and (6.8) the values for the shape functions of the five nodes, at a certain Gauss, point can be written in parts:

$$\mathbf{\Phi}_{1:4}(\mathbf{x}_{gp}) = \underbrace{\mathbf{p}^{T}(\mathbf{x}_{gp})[\mathbf{A}_{1:4}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{1:4}(x)}_{\mathbf{\Phi}_{1:4}:4\ nodes\ in\ the\ problem\ domain} + \underbrace{\mathbf{p}^{T}(\mathbf{x}_{gp})\,\mathbf{X}\,\mathbf{B}_{1:4}(x)}_{\delta\mathbf{\Phi}_{1:4}:add\ fifth\ node\ in\ the\ problem\ domain} \tag{6.12}$$

$$\mathbf{\Phi}_{5}(\mathbf{x}_{gp}) = \mathbf{p}^{T}(\mathbf{x}_{gp})[\mathbf{A}_{1:4}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{5}(x) + \mathbf{p}^{T}(\mathbf{x}_{gp})\,\mathbf{X}\,\mathbf{B}_{5}(x) \tag{6.13}$$

The first term of the Eq. (6.12) concerns the first four nodes and describes the values for the shape functions, when the fifth node is not located in the problem domain, and the interval is divided into three sub domains.
The second part of the Eq. (6.12) modifies the initial part, in order to retain the partition of unity, when the fifth node is added into the problem domain.

Figures 6.2 and 6.3 plot the shape functions of the first four nodes, before and after the addition of the fifth node in the problem domain respectively. To make this process more clear the second term of the Eq. (6.12), which expresses the difference between the shape functions, after the transition from four to five nodes, has also been plotted in Figure 6.4.
Figure 6.5 illustrates the shape function of the fifth node. The factor d_m of the support domain is chosen to be d_m = 2.2.The cubic spline weight function has been employed.

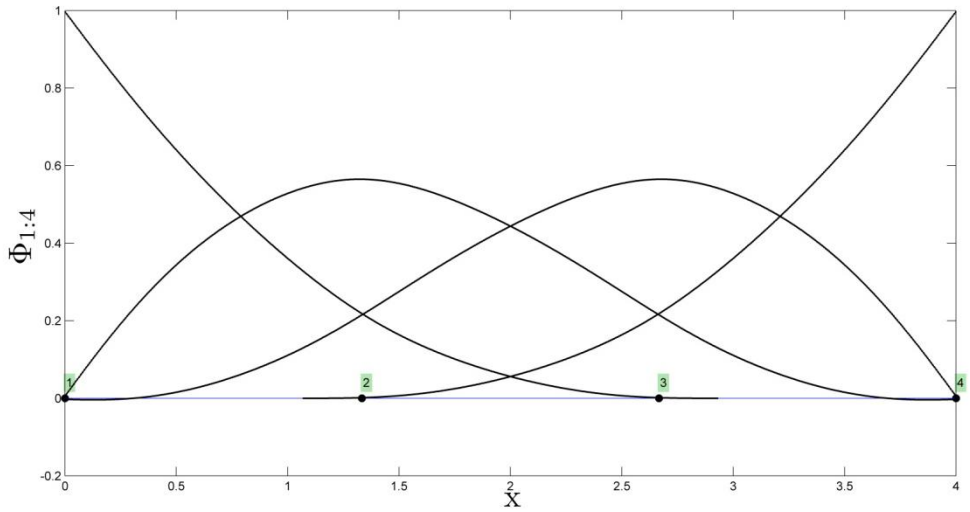


Figure 6.2
Shape functions of each node. Four nodes in the problem domain.

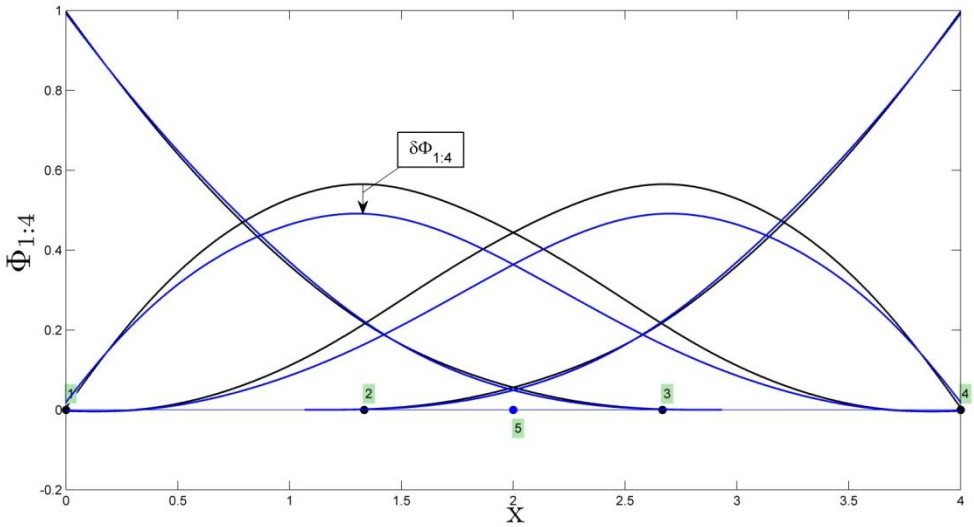


Figure 6.3
Shape functions of the first four nodes. Transition from four to five nodes.

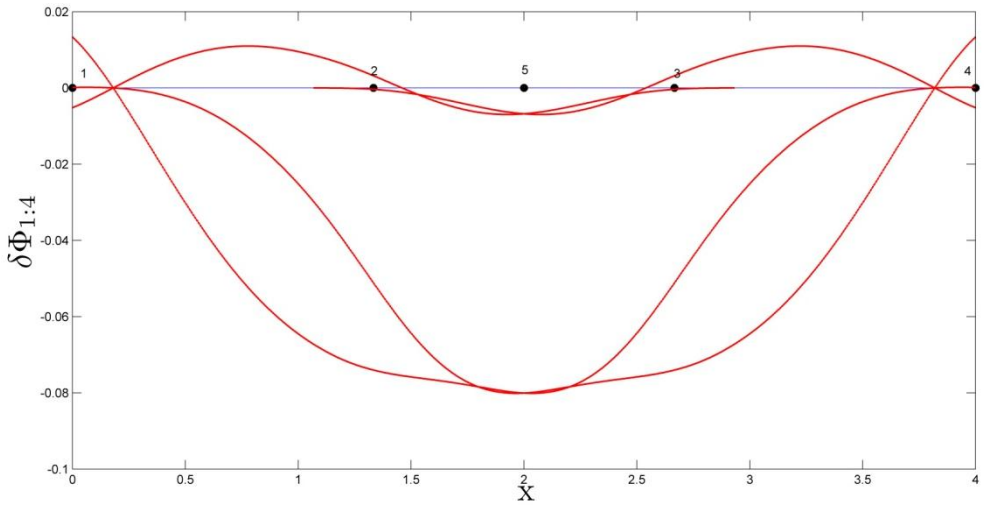


Figure 6.4
Difference between the shape functions of the first four nodes, after the transition from four to five nodes.

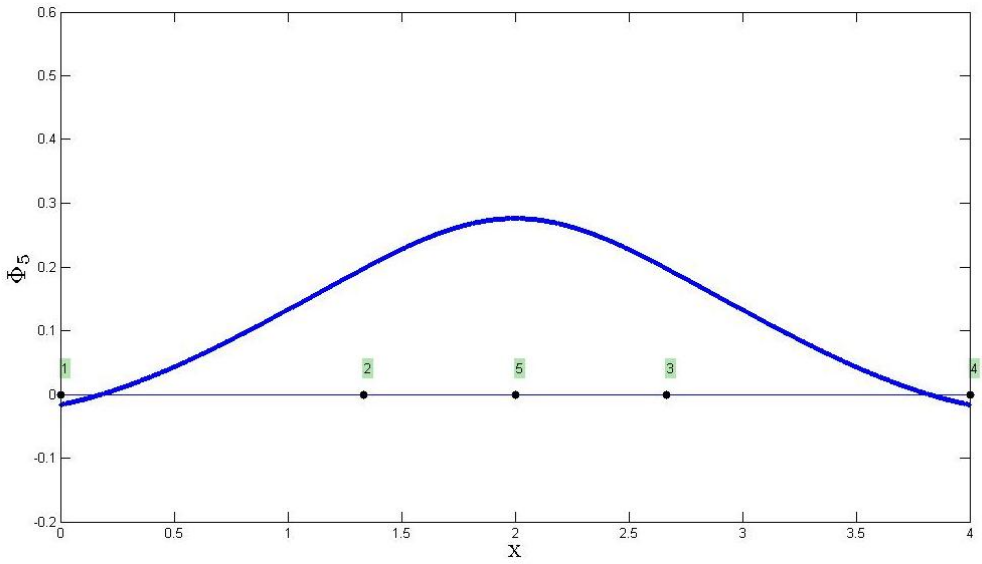


Figure 6.5
Shape function of the fifth node.

The above process, has shown the need for modification of the already existing values of the shape functions at a problem domain, when an extra node is added. This process is necessary, if we want to keep the partition of unity.
However if we only add the values of the shape function of the extra nodes (in this case the fifth node), the shape functions that are already computed for the rest of the nodes can remain constant, saving computational time. This procedure is able to offer a very good refinement to the problem domain, as we will see in the next Chapters. We will call this type of refinement, *hierarchical*, as it reminds us of the hierarchical forms of the FEM shape functions, where higher order forms are added to the elements, without modification of the already existing shape functions.
Furthermore this *numerical decomposition* of the shape functions that was carried out, can contribute to the calculation of the shape functions in parts. As we will notice in the next section, this process is extended to the first derivative of the shape functions, and therefore to the partial calculation of the stiffness matrix.

6.2 Total and hierarchical restoration of the stiffness matrix

6.2.1 Partition of the first derivative of the shape function

In the previous section we have managed to calculate the shape functions in parts. However in order to make this process useful for solid structure problems, it is necessary to extend it to the first derivative of the shape functions as indicated in Eq. (4.37).
We again consider the problem shown in Figure 6.1. If we do the calculations for the first derivative of the shape functions, we will notice that they can also be written in two parts. The first term concerns the first four nodes and describes the values for the

first derivative of the shape functions, when the fifth node is not located in the problem domain, and the interval is divided into three sub domains.
The second term modifies the initial part, when the fifth node is added into the problem domain. The analytical results of the calculations are presented into the following Matlab code. The code is solving the problem of adding only one node in the domain. It can easily be generalized for adding more nodes.

```
function [phi,dphix] =
Shapeh (gpos,x,v,dm,phi_prev,aa_prev,ainv_prev,daax_prev,dphix_prev)

%Import previous values of the shape functions at a particular Gauss point
%Import the previous weighted moment matrix aa
%Import their first derivatives
%Lv :How many nodes in the support domain
%--------------------------------------------------
%       HIERARCHICAL FORMULATION
%--------------------------------------------------
   aa_add=w(Lv)*[1 x(v(Lv))]'*[1 x(v(Lv))];
   daax_add=dwdx(Lv)*[1 x(v(Lv))]'*[1 x(v(Lv))];

   X=-inv(eye(2)+ainv_prev*aa_add)*ainv_prev*aa_add*ainv_prev;

%Calculation of the new shape functions
   phi(1:(Lv-1))=phi_prev+[1 gpos]*X*B(:,1:(Lv-1));
   phi(Lv)=[1 gpos]*ainv_prev*B(:,Lv)+[1 gpos]*X*B(:,Lv);

%Calculation of the first derivative of the shape functions
   dphix_plus(1:(Lv-1))=[0 1]*X*B(:,1:(Lv-1))+[1 gpos]*X*dbx(:,1:(Lv-1))...
   -[1 gpos]*(ainv_prev*daax_add*ainv_prev...
   +X*daax_prev*X+X*daax_add*X+ainv_prev*daax_prev*X...
   +X*daax_prev*ainv_prev+ainv_prev*daax_add*X...
   +X*daax_add*ainv_prev)*B(:,1:(Lv-1));

    dphix(1:(Lv-1))=dphix_prev+dphix_plus(1:(Lv-1));

    dphix(Lv) = [0 1]*((ainv_prev+X)*B(:,Lv))...
    +[1 gpos]*(-(ainv_prev+X)*(daax_prev+daax_add))*((ainv_prev+X)*B(:,Lv))...
    +[1 gpos]*((ainv_prev+X)*dbx(:,Lv));

end
```

Listing 6.1
Calculation of the shape functions and their first derivatives in parts, using the Binomial inverse theorem

The results of this procedure are also plotted in Figures 6.6 to 6.7.These Figures illustrate the first derivative of the shape functions of the first four nodes, before and after the addition of the fifth node in the problem domain, respectively.

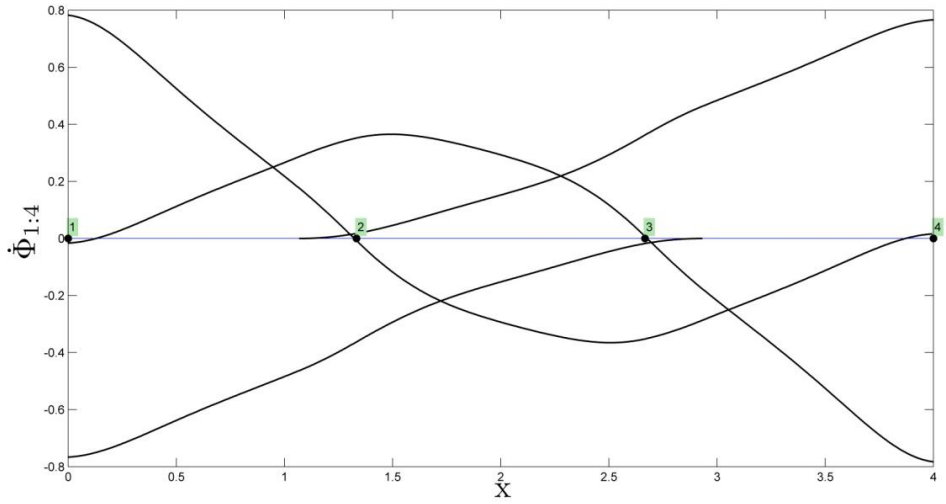


Figure 6.6
First derivatives of the shape functions of each node. Four nodes in the problem domain.

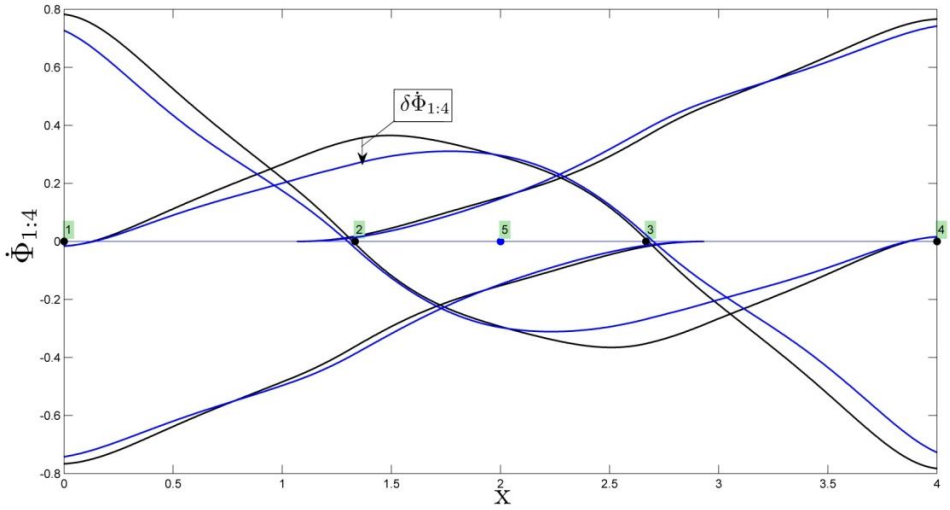


Figure 6.7
First derivatives of the shape functions of the first four nodes. Transition from four to five nodes.

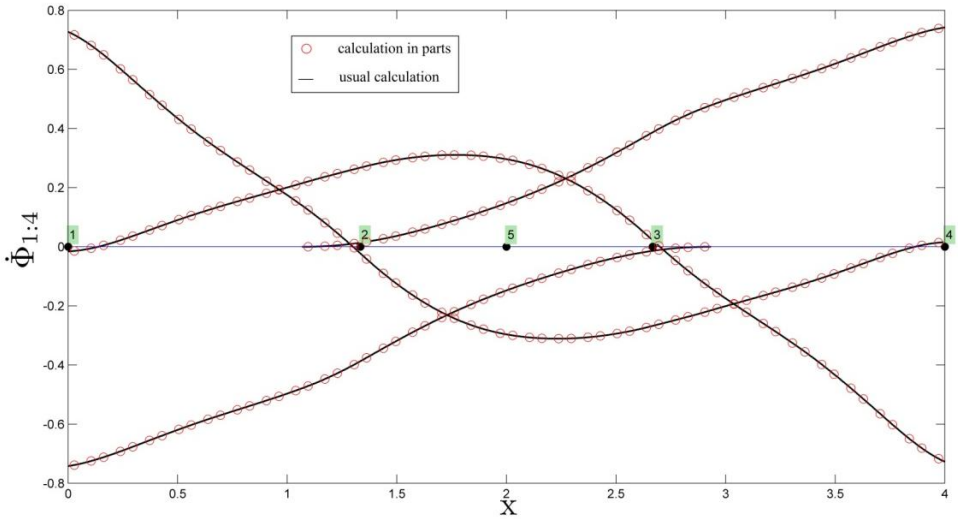


Figure 6.8
Total agreement between the calculation of the first derivatives in parts and the usual calculation

Figure 6.8 compares the first derivatives of the shape functions for the first four nodes, calculated by the usual process, with these calculated with the one that was proposed above. A total agreement is observed.

6.2.2 Total restoration of the stiffness matrix

We consider a two dimensional (2D) linear problem on the domain Ω bounded by Γ. As we have already proved, by using the Binomial inverse theorem, the EFG shape functions, as well as their first derivatives, can be written in two parts. This property is very useful, and can help us calculate these values, when extra points are added in a problem domain. Therefore for the shape functions we have :

$$\mathbf{\Phi}_{1:\mathrm{m}}^{\mathrm{new}}(\mathbf{x}) = \mathbf{\Phi}_{1:\mathrm{m}}^{\mathrm{prev}}(\mathbf{x}) + \delta\mathbf{\Phi}_{1:\mathrm{m}}(\mathbf{x}) \tag{6.14}$$

$$\mathbf{\Phi}_{(m+1):(m+\mathrm{n})}^{\mathrm{new}}(\mathbf{x}) = \mathbf{\Phi}_{(m+1):(m+\mathrm{n})}(\mathbf{x}) \tag{6.15}$$

where n is the number of the additional nodes in the problem domain, and m is the initial number of nodes. Respectively, for the partial derivatives of the shape functions we take:

$$\dot{\mathbf{\Phi}}_{1:\mathrm{m}}^{\mathrm{new}}(\mathbf{x}) = \dot{\mathbf{\Phi}}_{1:\mathrm{m}}^{\mathrm{prev}}(\mathbf{x}) + \delta\dot{\mathbf{\Phi}}_{1:\mathrm{m}}(\mathbf{x}) \tag{6.16}$$

$$\dot{\mathbf{\Phi}}_{(m+1):(m+\mathrm{n})}^{\mathrm{new}}(\mathbf{x}) = \dot{\mathbf{\Phi}}_{(m+1):(m+\mathrm{n})}(\mathbf{x}) \tag{6.17}$$

To assemble the global matrix **K** we will use the Eq. (4.42) for the nodal stiffness matrix:

$$\mathbf{K}_{IJ}^{new} = \int_{\Omega} \left[\mathbf{B}_I^{new}\right]^T \mathbf{c} \left[\mathbf{B}_J^{new}\right] d\Omega \tag{6.18}$$

where

$$\left[\mathbf{B}_I^{new}\right] = \begin{bmatrix} \varphi_{I,x}^{new} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{new} \\ \varphi_{I,\mathrm{y}}^{new} & \varphi_{I,x}^{new} \end{bmatrix} = \begin{cases} \begin{bmatrix} \varphi_{I,x}^{prev} + \delta\varphi_{I,x} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{prev} + \delta\varphi_{I,\mathrm{y}} \\ \varphi_{I,\mathrm{y}}^{prev} + \delta\varphi_{I,\mathrm{y}} & \varphi_{I,x}^{prev} + \delta\varphi_{I,x} \end{bmatrix}, & I \le m \\ \begin{bmatrix} \varphi_{I,x}^{new} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{new} \\ \varphi_{I,\mathrm{y}}^{new} & \varphi_{I,x}^{new} \end{bmatrix}, & m < I \le (m+n) \end{cases} \tag{6.19}$$

We will try to simplify the expression of the strain matrix $\mathbf{B}_I^{new}$ in order to replace it in the Eq. (6.18):

$$= \begin{cases} \underbrace{\begin{bmatrix} \varphi_{I,x}^{prev} & 0 \\ 0 & \varphi_{I,y}^{prev} \\ \varphi_{I,y}^{prev} & \varphi_{I,x}^{prev} \end{bmatrix}}_{\mathbf{B}_I^{prev}} + \underbrace{\begin{bmatrix} \delta\varphi_{I,x} & 0 \\ 0 & \delta\varphi_{I,y} \\ \delta\varphi_{I,y} & \delta\varphi_{I,x} \end{bmatrix}}_{\delta\mathbf{B}_I}, & I \le m \\ & = \left[\mathbf{B}_I^{prev}\right] + \left[\delta\mathbf{B}_I\right] \\ \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{B}_I^{prev}} + \underbrace{\begin{bmatrix} \varphi_{I,x}^{new} & 0 \\ 0 & \varphi_{I,y}^{new} \\ \varphi_{I,y}^{new} & \varphi_{I,x}^{new} \end{bmatrix}}_{\delta\mathbf{B}_I}, & m < I \le (m+n) \end{cases} \quad (6.20)$$

We are now ready to substitute the Eq. (6.20) into the expression of the nodal stiffness matrix Eq. (6.18).Therefore the nodal stiffness matrix can be written as follows :

$$\begin{aligned} \mathbf{K}_{IJ}^{new} &= \int_\Omega \left[\mathbf{B}_I^{new}\right]^T \mathbf{c} \left[\mathbf{B}_J^{new}\right] d\Omega = \int_\Omega \left\{\left[\mathbf{B}_I^{prev}\right] + \left[\delta\mathbf{B}_I\right]\right\}^T \mathbf{c} \left\{\left[\mathbf{B}_J^{prev}\right] + \left[\delta\mathbf{B}_J\right]\right\} d\Omega \\ &= \underbrace{\int_\Omega \left[\mathbf{B}_I^{prev}\right]^T \mathbf{c} \left[\delta\mathbf{B}_J\right] d\Omega + \int_\Omega \left[\delta\mathbf{B}_I\right]^T \mathbf{c} \left[\mathbf{B}_J^{prev}\right] d\Omega + \int_\Omega \left[\delta\mathbf{B}_I\right]^T \mathbf{c} \left[\delta\mathbf{B}_J\right] d\Omega}_{\delta\mathbf{K}_{IJ}} \\ &+ \underbrace{\int_\Omega \left[\mathbf{B}_I^{prev}\right]^T \mathbf{c} \left[\mathbf{B}_J^{prev}\right] d\Omega}_{\mathbf{K}_{IJ}^{prev}} = \mathbf{K}_{IJ}^{prev} + \delta\mathbf{K}_{IJ} \end{aligned} \quad (6.21)$$

We have thus managed to separate the global stiffness matrix in two parts. The first part, $\mathbf{K}^{\text{prev}}$, concerns the first m nodes, that were initially located in the problem domain. The second term $\delta\mathbf{K}$, modifies the already existing stiffness matrix, in order to take the additional points into consideration. This is a very important result as, when extra nodes are added into a problem domain, only $\delta\mathbf{K}$ matrix has to be calculated, a process that saves much computational time. What is more, this numerical decomposition of the stiffness matrix, can probably be of use for the parallelization of the calculation of the stiffness matrix for the EFG method. This diploma thesis proposes this idea for future research.

Numerical example

We consider again the cantilever beam problem, subjected to a parabolic load at the free end. In this example, the parameters for this cantilever beam are taken as follows.

Loading	P=2000 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Height of the beam	D=6m
Length of the beam	L=24m

For this problem, the regular node distribution was used. Firstly the beam is solved with a 11x4 node distribution. The stiffness matrix is calculated, as well as the displacements for every node and the stress field. The set of nodes is plotted in Figure 6.9.
Background mesh is employed for numerical integration of the weak form. In each Gauss cell 2x2 Gauss quadrature is used. A linear basis and a cubic spline weight function are used for the MLS approximation. The support domain is rectangular with a dimension of 2.2 times the nodal spacing. The penalty factor was chosen to be $a = 10^{10}$.

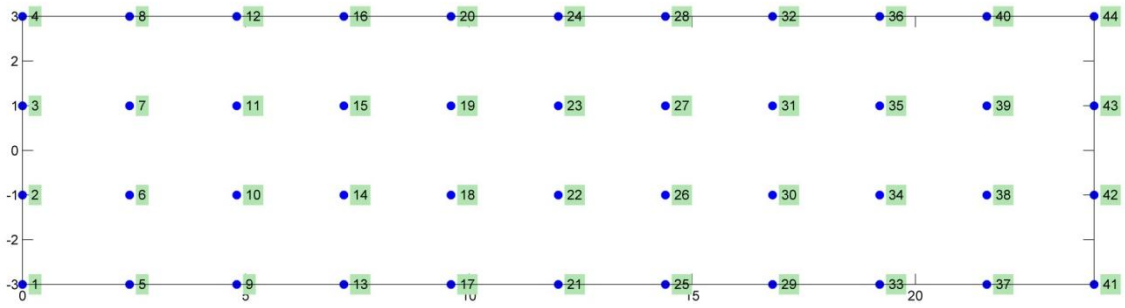


Figure 6.9
Initial arrangement of nodes

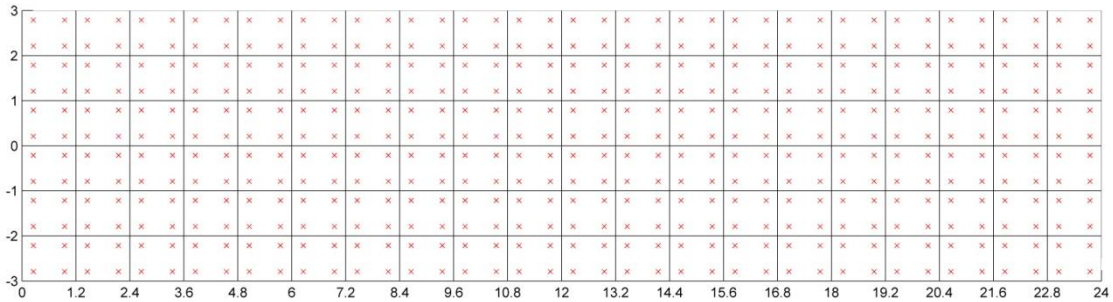


Figure 6.10
Mesh used for integration (20x6)

Then eight (8) new nodes are added in the problem domain. The new nodes are plotted with black color in Figure 6.11.
The calculations were carried out both with the usual method, and the partition of the stiffness matrix method. For the second process, only the additional matrix $\delta \boldsymbol{K}$ was computed, and then added to the already existing stiffness matrix, that was calculated before the addition of the extra nodes. The same mesh, as indicated in Figure 6.10, was used for integration.

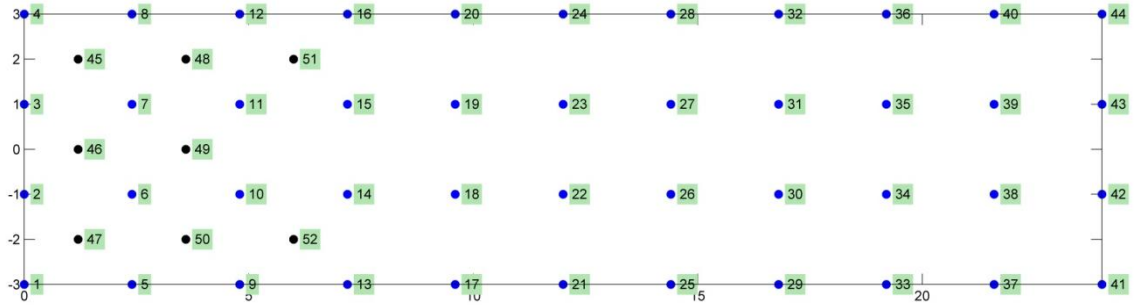
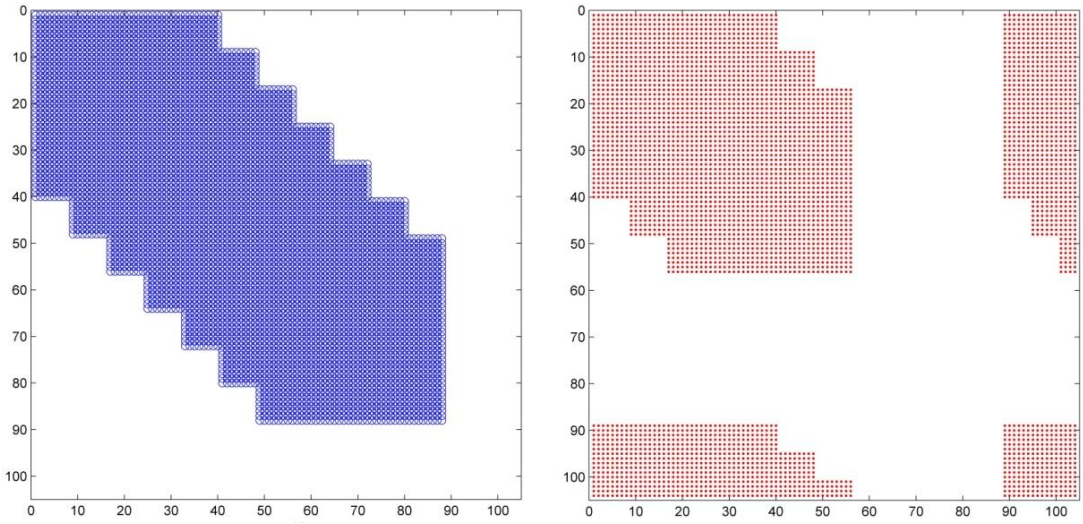
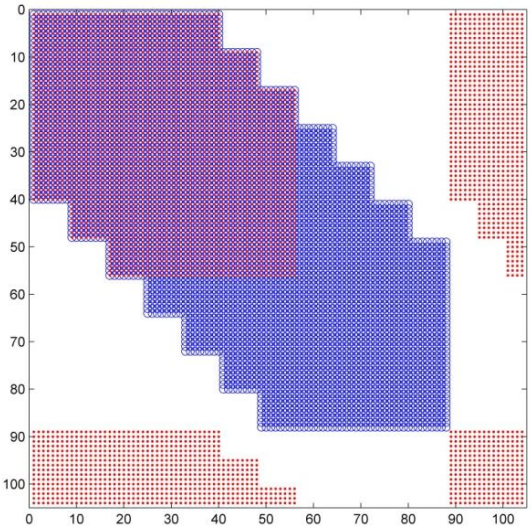


Figure 6.11
New arrangement of nodes

Figure 6.12 illustrates the sparsity pattern of the matrices, used to assemble the final stiffness matrix of the beam.
As we can see, the stiffness matrix of the cantilever beam, computed before the addition of the extra nodes, is a symmetrical one [Figure 6.12(a)] . As long as the support domain is compact and does not cover too wide the problem domain, many $\mathbf{K}_{IJ}^{prev}$ in Eq. (4.42) are zero, and the global stiffness matrix is sparse. Furthermore, the nodes are properly numbered, and as a result $\mathbf{K}^{prev}$ is also banded. An interesting notification is also that the bandwidth of the EFG stiffness matrix is wider than in FEM, for a domain discretized with EFG and FEM having equal number of nodes and Gauss points. Since the domains of influence of Gauss points are much larger than the corresponding domains in FEM, the amount of the interactions between Gauss points and nodes is higher.
Figure 6.12(b) plots $\delta \boldsymbol{K}$, the additional matrix, calculated to modify the existing stiffness matrix. As we will notice the $\delta \boldsymbol{K}$ is also a symmetric matrix. The non zero $\delta \mathbf{K}_{IJ}$ expresses the influence of the attachment of new points. The support domain of the extra nodes does not cover the whole problem domain, therefore $\mathbf{K}^{prev}$ needs to be altered only for some degrees of freedom (i.e. 1 to 56).Finally in Figure 6.12 (c) the two matrixes are added to give $\mathbf{K}^{new}$.
After the computation of the new stiffness matrix, the displacement field for every node as well as the stress field for the cantilever beam are calculated. The same calculations are also carried out for the final node arrangement, using the usual EFG method.



(a) $\mathbf{K}^{prev}$ [88x88] **(b)** $\delta\mathbf{K}$ [104x104]



(c) $\mathbf{K}^{new}$ [104x104]

Figure 6.12
Transition of the stiffness matrix from $\mathbf{K}^{prev}$ to $\mathbf{K}^{new}$.

To compare the two procedures, and ensure total coincidence of the results, the shear stress fields for each case are plotted in Figure 6.13.
The plot shows excellent agreement between the solutions of the two procedures. This fact reveals that the solution in EFG method can be partitioned with great accuracy, and gives us a good step for future research.

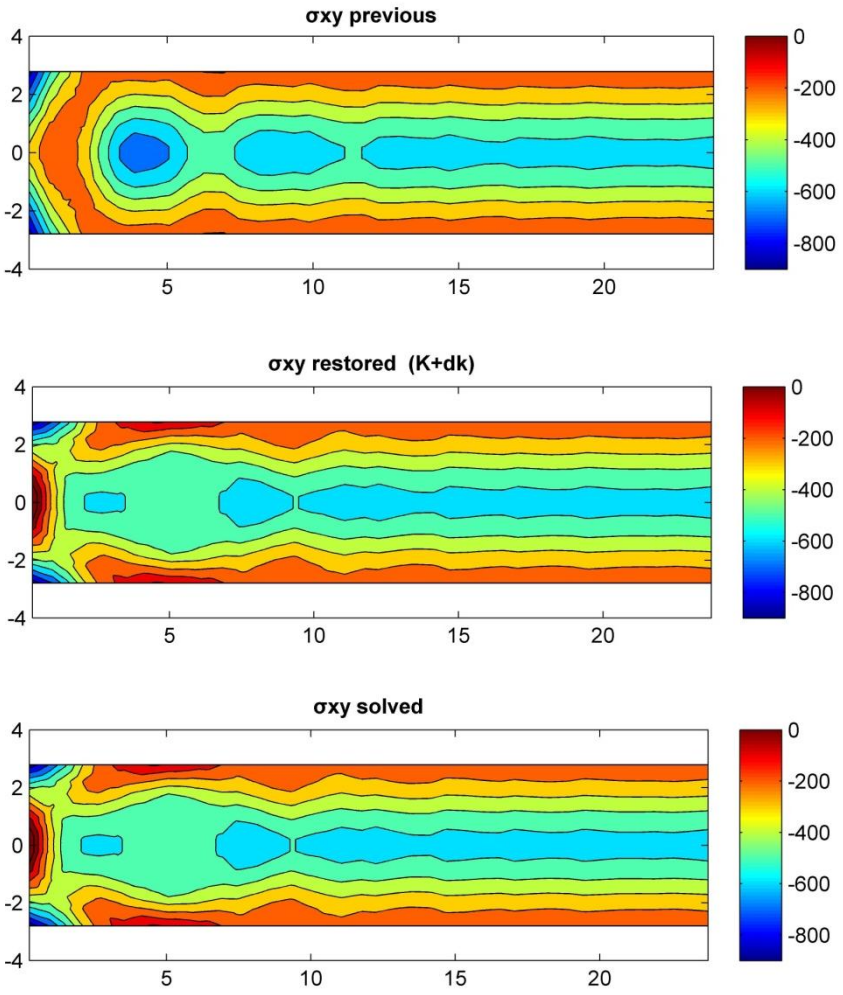


Figure 6.13
Shear stress field for the cantilever beam before and after the addition of extra nodes

6.2.3 Hierarchical restoration of the stiffness matrix

We consider again the same two dimensional (2D) linear problem on the domain Ω bounded by Γ. The total restoration of the stiffness matrix, as we have already seen, is an accurate way to add new nodes in a problem domain. Now we propose the addition of the values of the shape functions, only for the extra nodes, without the modification of the existing shape functions that cover the problem domain. Therefore for the shape functions we take:

$$\mathbf{\Phi}_{1:\mathrm{m}}{}^{\mathrm{new}}(\mathbf{x}) = \mathbf{\Phi}_{1:\mathrm{m}}{}^{\mathrm{prev}}(\mathbf{x}) \tag{6.22}$$

$$\mathbf{\Phi}_{(m+1):(m+\mathrm{n})}{}^{\mathrm{new}}(\mathbf{x}) = \mathbf{\Phi}_{(m+1):(m+\mathrm{n})}{}^{hier}(\mathbf{x}) \tag{6.23}$$

where n is the number of the additional nodes in the problem domain, and m is the initial number of nodes. $\mathbf{\Phi}^{hier}(\mathbf{x})$ indicates the values of the shape functions of the extra nodes, that are added hierarchically and can be computed as seen in Listing 1. Respectively, for the partial derivatives of the shape functions we take:

$$\dot{\mathbf{\Phi}}_{1:\mathrm{m}}{}^{\mathrm{new}}(\mathbf{x}) = \dot{\mathbf{\Phi}}_{1:\mathrm{m}}{}^{\mathrm{prev}}(\mathbf{x}) \tag{6.24}$$

$$\dot{\mathbf{\Phi}}_{(m+1):(m+\mathrm{n})}{}^{\mathrm{new}}(\mathbf{x}) = \dot{\mathbf{\Phi}}_{(m+1):(m+\mathrm{n})}{}^{hier}(\mathbf{x}) \tag{6.25}$$

To assemble the global matrix **K** we will use the Eq. (4.42) for the nodal stiffness matrix:

$$\mathbf{K}_{IJ}^{new} = \int_{\Omega} \left[\mathbf{B}_I^{new}\right]^T \mathbf{c} \left[\mathbf{B}_J^{new}\right] d\Omega \tag{6.26}$$

where

$$\left[\mathbf{B}_I^{new}\right] = \begin{bmatrix} \varphi_{I,x}^{new} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{new} \\ \varphi_{I,\mathrm{y}}^{new} & \varphi_{I,x}^{new} \end{bmatrix} = \begin{cases} \begin{bmatrix} \varphi_{I,x}^{prev} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{prev} \\ \varphi_{I,\mathrm{y}}^{prev} & \varphi_{I,x}^{prev} \end{bmatrix} = \left[\mathbf{B}_I^{prev}\right], & I \le m \\ \begin{bmatrix} \varphi_{I,x}^{hier} & 0 \\ 0 & \varphi_{I,\mathrm{y}}^{hier} \\ \varphi_{I,\mathrm{y}}^{hier} & \varphi_{I,x}^{hier} \end{bmatrix} = \left[\mathbf{B}_I^{hier}\right], & m < I \le (m+n) \end{cases} \tag{6.27}$$

If we now substitute the Eq. (6.27) into the expression of the nodal stiffness matrix [Eq. (6.26)] for every case, the hierarchical stiffness matrix can be written in parts as follows :

$$\left[\mathbf{K}^{hier}\right] = \begin{bmatrix} \underbrace{\left[\mathbf{K}^{pp}\right]}_{[2m\mathbf{x}2m]} & \left[\mathbf{K}^{ph}\right] \\ \left[\mathbf{K}^{hp}\right] & \left[\mathbf{K}^{hh}\right]_{[2n\mathbf{x}2n]} \end{bmatrix} \tag{6.28}$$

where

$$\mathbf{K}_{IJ}^{pp} = \mathbf{K}_{IJ}^{prev} = \int_{\Omega} \left[\mathbf{B}_I^{prev}\right]^T \mathbf{c} \left[\mathbf{B}_J^{prev}\right] d\Omega \tag{6.29}$$

$$\mathbf{K}_{IJ}^{hp} = \int_{\Omega} \left[\mathbf{B}_I^{hier}\right]^T \mathbf{c} \left[\mathbf{B}_J^{prev}\right] d\Omega \tag{6.30}$$

$$\mathbf{K}_{IJ}^{ph} = \int_{\Omega} \left[\mathbf{B}_I^{prev}\right]^T \mathbf{c} \left[\mathbf{B}_J^{hier}\right] d\Omega \tag{6.31}$$

$$\mathbf{K}_{IJ}^{hh} = \int_{\Omega} \left[\mathbf{B}_I^{hier}\right]^T \mathbf{c} \left[\mathbf{B}_J^{hier}\right] d\Omega \tag{6.32}$$

Through the procedure described above, we have managed to take the addition of extra nodes in the problem domain into consideration, without modifying the existing stiffness matrix. In Eq. (6.32) $\mathbf{K}^{hh}$ is the stiffness matrix of the additional forms, while $\mathbf{K}^{hp}$ and $\mathbf{K}^{ph}$ express the interaction between the previous nodes and the extra ones. Using the Eq. (6.30) and Eq. (6.31) it is easy to write :

$$\mathbf{K}^{hp} = \left[\mathbf{K}^{ph}\right]^T \tag{6.33}$$

Due to this property, only one of the above matrixes needs to be computed, making the calculations easier. The dimension of these matrixes may vary, depending on the number of nodes that are affected.

This technique does not give the exact results for the addition of new points in a problem domain, because the partition of unity is no longer fulfilled. However it can be used as part of a refinement process, that can improve the already calculated results for a problem domain, *a posteriori.* The convenience of calculating the extra matrices, makes the hierarchical restoration of the stiffness matrix very promising, at it provides significantly improved results, with a low computational cost.

Numerical example

This time we deal with the cantilever beam problem, subjected to a parabolic load at the free end. In this example, the parameters for the cantilever beam are chosen to be the same as in the total restoration of the stiffness matrix.

Loading	P=2000 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Height of the beam	D=6m
Length of the beam	L=24m

The beam is analyzed with a 11x4 node distribution. The stiffness matrix is calculated, as well as the displacements for every node and the stress field. The set of nodes is illustrated in Figure 6.14.
Background mesh is employed for numerical integration of the weak form. In each Gauss cell 2x2 Gauss quadrature is used. A linear basis and a cubic spline weight function are used for the MLS approximation. The support domain is rectangular with a dimension of 2.2 times the nodal spacing. The penalty factor was chosen to be $a = 10^{10}$.
Then eight (8) new nodes are added in the problem domain. The new nodes are plotted with black color in Figure 6.15.

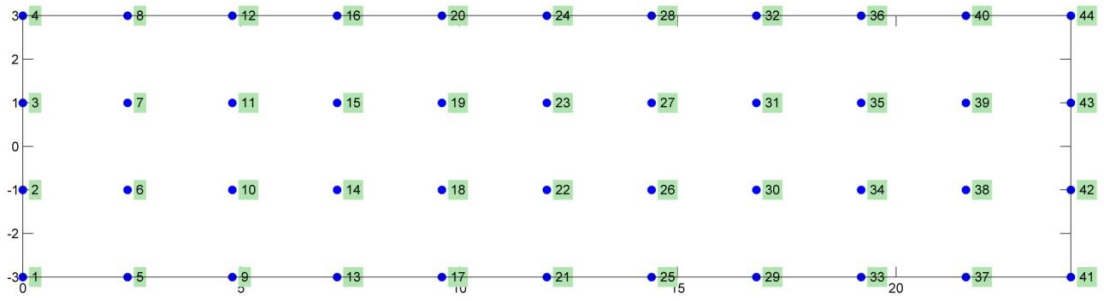


Figure 6.14
Initial arrangement of nodes

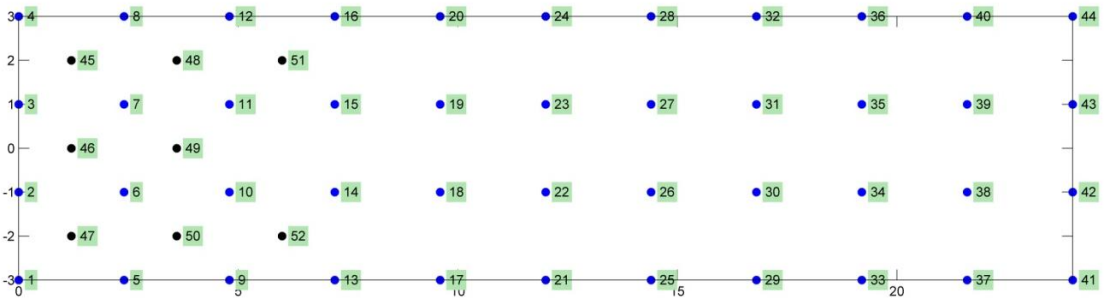


Figure 6.15
New arrangement of nodes

The addition of the extra nodes is performed with the 'hierarchical' way described above. The already calculated values of the shape functions of the first 44 nodes, are kept constant, and only eight new forms are attached in the problem domain. Some of them can be found plotted in the following figures. Figure 6.16 and 6.17 illustrate the additional shape functions of nodes 49 and 50 respectively.

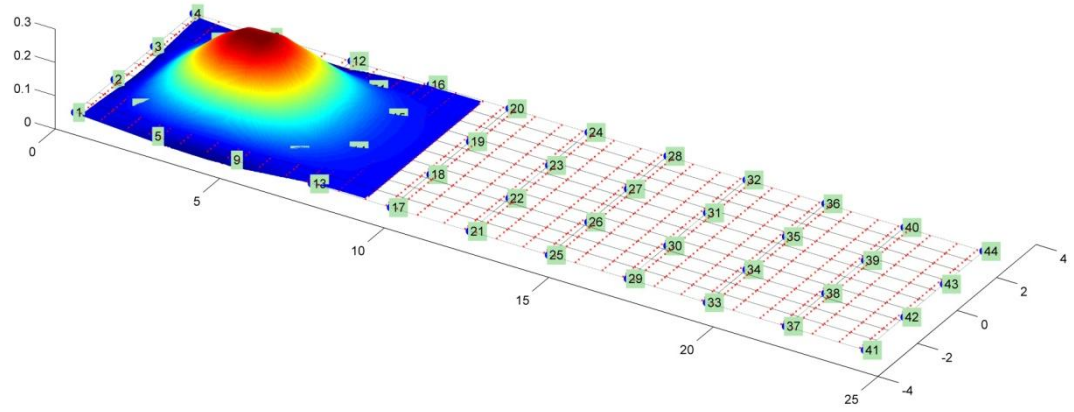


Figure 6.16
Additional shape function of the 49th node

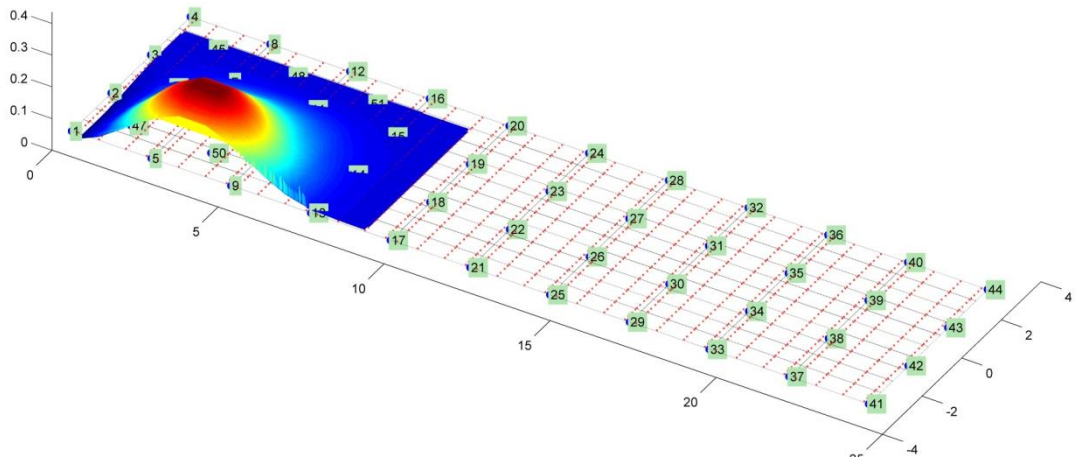
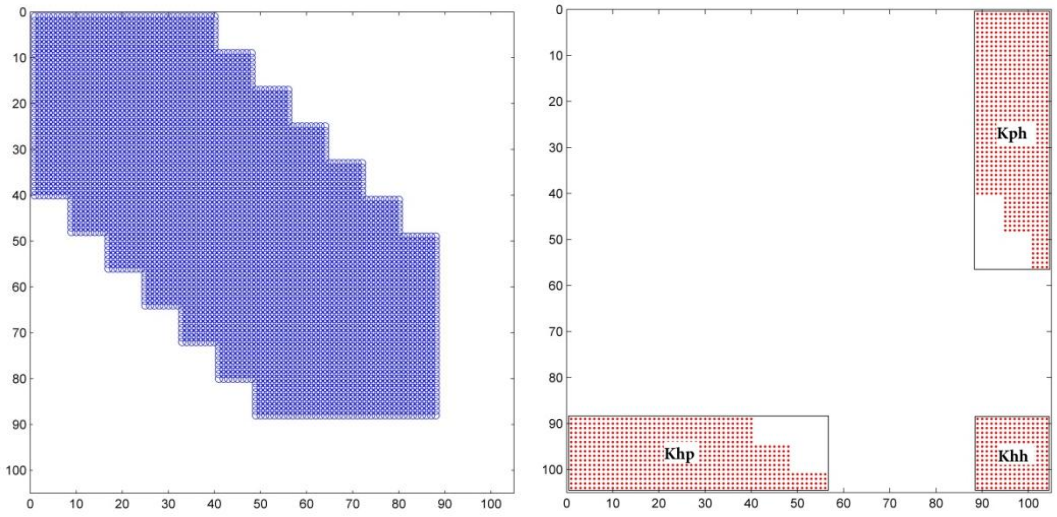
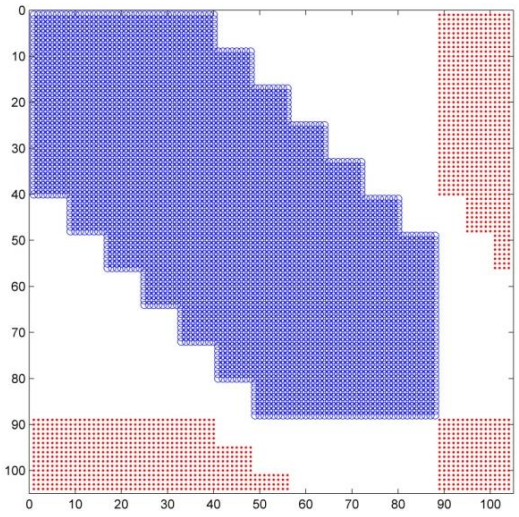


Figure 6.17
Additional shape function of the 50th node

The stiffness matrix of the initial 44 nodes $\mathbf{K}^{prev}$, is already calculated and it is kept constant. To perform the hierarchical refinement we only need to compute the extra matrices $\mathbf{K}^{hp}$ and $\mathbf{K}^{hh}$, using Eq. (6.30) and Eq. (6.32) respectively. The last node that is affected by the addition of the extra nodes is the 28th node. Therefore the dimension of the $\mathbf{K}^{hp}$ matrix is [16x56]. Figure 6.18 illustrates the sparsity pattern of the matrices, used to assemble the final stiffness matrix of the beam. As we can notice, in contrast to the total restoration method, the hierarchical restoration of the stiffness matrix does not modify the initial stiffness matrix.



(a) $\mathbf{K}^{prev}$ [88x88] **(b)** Additional matrixes



(c) $\mathbf{K}^{hier}$ [104x104]

Figure 6.18
Formulation of the hierarchical stiffness matrix

We have already mentioned that the hierarchical process does not offer the same solution for a problem domain, in which new nodes are added, compared to the usual EFG procedure. However it can be used as a refinement technique that can improve a certain solution, already calculated. In order to estimate the improvement of the final solution, we also compute the displacements and the stress field for the cantilever beam, after the attachment of the new nodes, using the usual EFG method.
To compare the two procedures, the energy norm and the displacement norm are determined, to estimate the error of the analysis, before and after the addition of new nodes. After that the same norms are calculated to define the error between the hierarchical and the usual EFG solution that is taken as reference solution. The shear stress fields for each case are plotted in Figure 6.19.

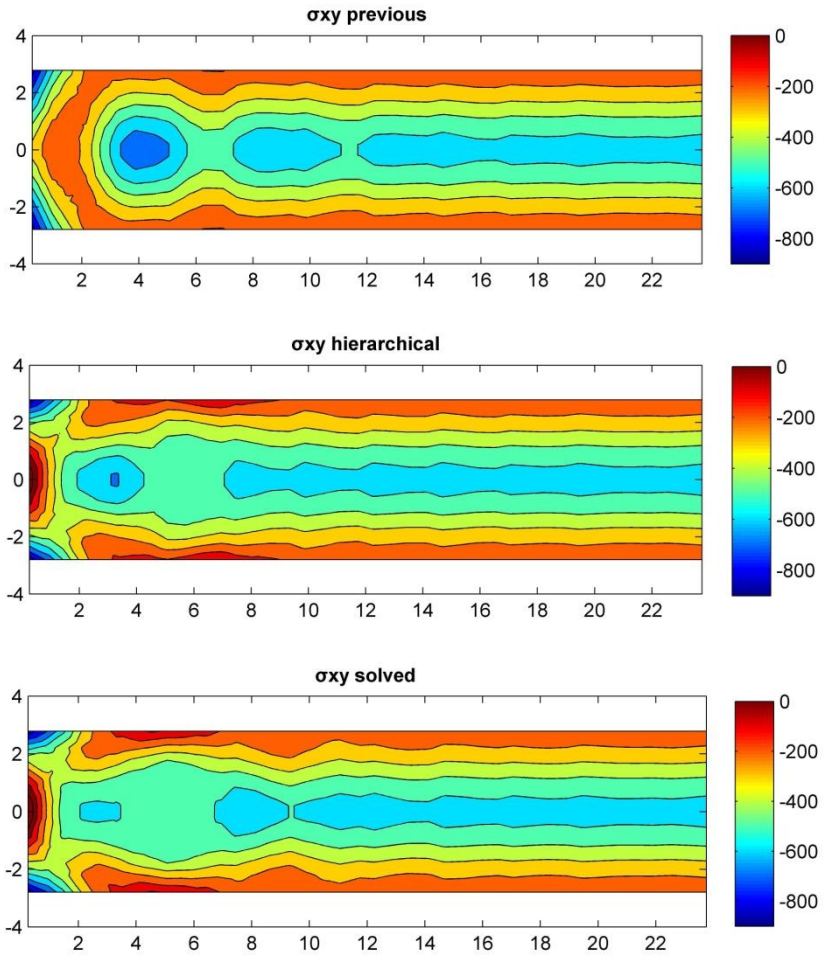


Figure 6.19
Shear stress field for the cantilever beam

The accuracy or the quality of the error estimators is measured by an *error index* θ_h, which is defined as:

$$\theta_h = \frac{e_{hier}}{e_{exact}} \tag{6.34}$$

where e_{hier} is the estimated error norm of the hierarchical refinement process , and e_{exact} is the already existing error, between the numerical solutions of the two node arrangements. For the above norms computed, the results can be seen in Tables 6-1 and 6-2.

Energy norm (%)			
	Initial sol(44nod)-Reference sol(52nod)	Hierarch. sol (52nod)-Reference sol(52nod)	error index θ_h
	26,97	9,59	0,36

Table 6-1
Energy norm for the cantilever beam

Displacement Norm			
	Initial sol(44nod)-Reference sol(52nod)	Hierarch. sol (52nod)-Reference sol(52nod)	error index θ
	4,68E-05	7,71E-06	0,16

Table 6-2
Displacement norm for the cantilever beam

As we can notice the hierarchical refinement leads to very good results. The displacement field as well as the stresses calculated are improved, and much higher accuracy is gained. In the next chapter a refinement technique in EFG method is proposed, using the hierarchical concepts presented in this section.

7 Hierarchical refinement

In this Chapter, a refinement procedure using the element free Galerkin method (EFG) for the solution of 2D elastostatic problem is suggested. This refinement procedure is based on the well-known Zienkiewicz and Zhu (Z–Z) error estimator for the a posteriori error estimation and a simple point refinement scheme for new point mesh generation. The new points are added in the problem domain in an 'hierarchical' way described in the previous Chapter. In order to make this process clear, the theoretical background of stress recovery is presented, as well as two dimensional linear solid mechanics problems are solved.

7.1 Error estimation and stress recovery procedure for EFG

7.1.1 Definition of error and the Zienkiewicz–Zhu error estimator

As already seen in previous chapters the energy norm for the EFG method is given by

$$e_{energy}^{exact} = \left[\frac{1}{2} \int_{\Omega} (\boldsymbol{\varepsilon}_{num} - \boldsymbol{\varepsilon}_{exact})^T \mathbf{D} (\boldsymbol{\varepsilon}_{num} - \boldsymbol{\varepsilon}_{exact}) \, d\Omega \right]^{1/2} \tag{7.1}$$

and the displacement norm is given by

$$e_{displacement}^{exact} = \left[\int_{\Omega} \left[(\mathbf{u}_{num} - \mathbf{u}_{exact})^T (\mathbf{u}_{num} - \mathbf{u}_{exact}) d\Omega \right] \right]^{1/2} \tag{7.2}$$

where $\boldsymbol{\varepsilon}_{\text{num}}$ and $\boldsymbol{\varepsilon}_{\text{exact}}$ are the numerical strain vector and the exact strain vector, respectively. The same symbolism is followed for the displacement vectors $\mathbf{u}_{\text{num}}$ and $\mathbf{u}_{\text{exact}}$. These error norms are often called exact norms.

The essence of the Zienkiewicz–Zhu error estimator is to obtain a smoothed strain field, by means of a smoothing or refinement process. We will denote this stress field as $\boldsymbol{\varepsilon}_{hier}$, because in our case an hierarchical refinement technique is employed. Respectively $\mathbf{u}_{\text{hier}}$ will be the displacement vector of the hierarchical solution. To study the performance of the error estimator, the recovery error norms will also be computed. The energy norm is given by

$$e_{energy}^{hier} = \left[\frac{1}{2} \int_{\Omega} (\boldsymbol{\varepsilon}_{hier} - \boldsymbol{\varepsilon}_{exact})^T \mathbf{D} (\boldsymbol{\varepsilon}_{hier} - \boldsymbol{\varepsilon}_{exact}) \, d\Omega \right]^{1/2} \tag{7.3}$$

and the displacement norm is given by

$$e^{hier}_{displacement} = \left[\int_{\Omega} \left[(\mathbf{u}_{hier} - \mathbf{u}_{exact})^T (\mathbf{u}_{hier} - \mathbf{u}_{exact}) d\Omega \right] \right]^{1/2} \tag{7.4}$$

In addition, in order to assess the efficiency of the error estimator, the already used *error index* θ_h will be calculated

$$\theta_h = \frac{e_{hier}}{e_{exact}} \tag{7.5}$$

We have mentioned that we will use the exact solution in order to compute the error norms and quantify the improvement of the recovery procedure. However in most cases the analytical solution of the problem is not known. We can therefore use as reference solution, the one which is calculated using the most nodes in the problem domain. If we look at the Section 3.6, the consistency of the EFG methods, ensures that if the number of nodes in a certain domain tends to become infinite, the approximation equation will represent the exact differential equation and the boundary conditions for a certain problem. As a result, while the number of nodes in a problem domain are increasing, the result becomes more accurate.

7.1.2 The T-Belytschko's stress recovery scheme (TB scheme)

This is a popular recovery stress scheme first suggested by Tabbara et al. and then subsequently used by Chung and Belytschko for error estimation in the EFG method. The recovery stress field can be written as follows

$$\boldsymbol{\varepsilon}^{\mathrm{TB}}(\mathbf{x}) = \sum_{i=1}^{n_{TB}} \Phi_i^{TB}(\mathbf{x}) \boldsymbol{\varepsilon}^{num}(\mathrm{x_i}) \tag{7.6}$$

where $\boldsymbol{\varepsilon}^{num}$ is the EFG stress evaluated at node i, Φ_i^{TB} is a set of recovery MLS shape functions and n_{TB} is the number of nodes whose support includes the point **x**. By using the Eq. (4.37) the recovery stress can be written as

$$\boldsymbol{\varepsilon}^{\mathrm{TB}}(\mathbf{x}) = \sum_{i=1}^{n_{TB}} \Phi_i^{TB}(\mathbf{x}) \left(\mathbf{D} \sum_{j=1}^{Ni} \mathbf{B}_j(x_i) \mathbf{u}_{\mathrm{j}} \right) \tag{7.7}$$

In Eq. (7.7) $\mathrm{N_i}$ is the number of nodal points associate with the support of the *i*th node. The recovery MLS shape function can be constructed through a similar process as in the case of MLS shape functions construction in the EFG analysis. In practice in order to get an accurate recovered stress field, Φ_i^{TB} are constructed using different

support domains, on the same nodes of the original. The procedure is illustrated in Figure 7.1, with the stress field calculated initially on the Gauss points.

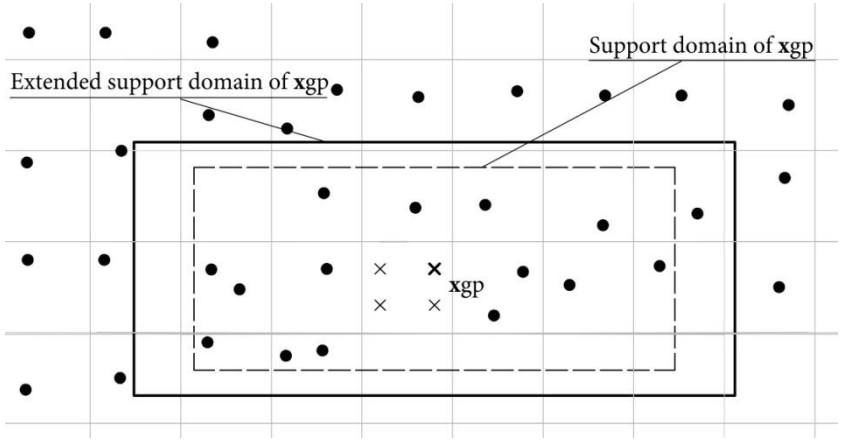


Figure 7.1
Stress recovery TB scheme

One can notice that the EFG stress is expressed in terms of the first derivatives of the MLS shape functions. However, the recovered stress is expressed in terms of another set of MLS shape functions. This means that with an appropriate choice of the extended support domain, $\boldsymbol{\varepsilon}^{TB}(\mathbf{x})$ will be more accurate than $\boldsymbol{\varepsilon}^{num}(\mathbf{x})$. From Eq. (7.7), one can also interpret that $\boldsymbol{\varepsilon}^{TB}(\mathbf{x})$ is computed by taking the product of MLS shape functions and the nodal EFG stress. Thus, it follows that the TB scheme is in fact nothing more than performing another MLS fitting over the nodal EFG stress.

7.2 Special procedures for EFG method refinement

7.2.1 Node generation strategy

In the EFG analysis the most convenient way to impose the above stress recovery scheme, is to define the T-Belytschko's stress field at the Gauss points used for integration. The procedure of hierarchical refinement can be written in steps, as follows:

- First, we compute the numerical stresses $\boldsymbol{\varepsilon}^{num}(\mathbf{x})$ at the Gauss points.
- Then the calculation of the T-Belytschko's stresses $\boldsymbol{\varepsilon}^{TB}(\mathbf{x})$ is carried out, using an extended support domain.
- For each Gauss point the local energy error is computed. In the numerical examples solved, the local error field is plotted in a contour pattern. The local energy error for a certain Gauss point is given as follows:

$$e^{gp}_{energy} = \frac{1}{2} w_{gp} \det \mathbf{J}^{gp} \left[\boldsymbol{\varepsilon}^{num}(\mathbf{x}_{gp}) - \boldsymbol{\varepsilon}^{TB}(\mathbf{x}_{gp}) \right]^T \mathbf{D} \left[\boldsymbol{\varepsilon}^{num}(\mathbf{x}_{gp}) - \boldsymbol{\varepsilon}^{TB}(\mathbf{x}_{gp}) \right] \tag{7.8}$$

- A finite number of Gauss points, where the local energy error is higher, is chosen.
- The Gauss cell for every chosen Gauss point is spotted uniquely.
- Finally a new node is added, at the centroid of each Gauss cell, as illustrated in Figure 7.2.

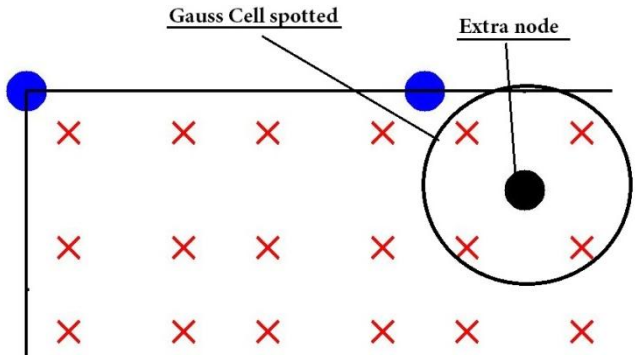


Figure 7.2
Node generation technique

7.2.2 Reduction of the size of the support domain of newly added nodes.

As we have already mentioned, when the hierarchical refinement technique is used, the partition of unity is no longer fulfilled. This occurs in fields when there is an overlap of the support domains of the extra nodes, and the already existing ones. To minimize this effect and get more accurate solutions, we reduce the size of the support domain of the newly added nodes as illustrated in Figure 7.3. This seems the best way to impose the hierarchical refinement proposed.

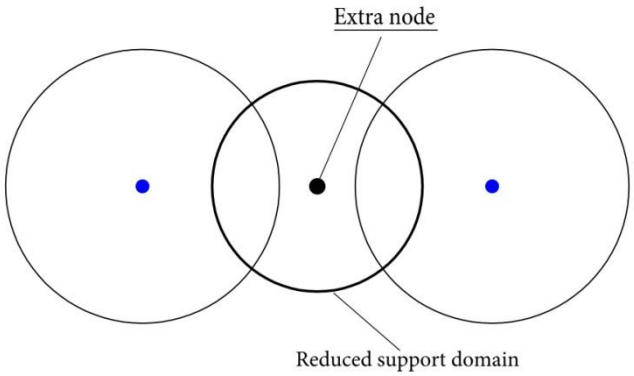


Figure 7.3
Reduction of the size of the support domain of the extra nodes

7.3 Numerical Examples

7.3.1 The Cantilever Beam Problem

We consider again the cantilever beam problem, subjected to a parabolic load at the free end. In this example, the parameters for this cantilever beam are taken as follows.

Loading	P=1000 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Height of the beam	D=12m
Length of the beam	L=48m

The regular node distribution was used. Firstly the beam is solved with a 13x4 node distribution. The displacements for every node and the stress field are calculated. The set of nodes is plotted in Figure 7.4.
Background mesh is employed for numerical integration of the weak form. In each Gauss cell 2x2 Gauss quadrature is used. A linear basis and a cubic spline weight function are used for the MLS approximation. The support domain is rectangular with a dimension of 2.2 times the nodal spacing. The penalty factor was chosen to be $a = 10^{10}$.

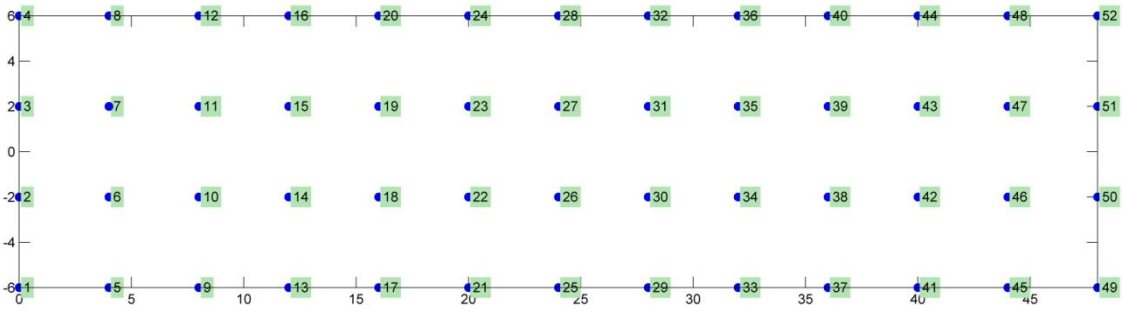


Figure 7.4
Initial arrangement of nodes

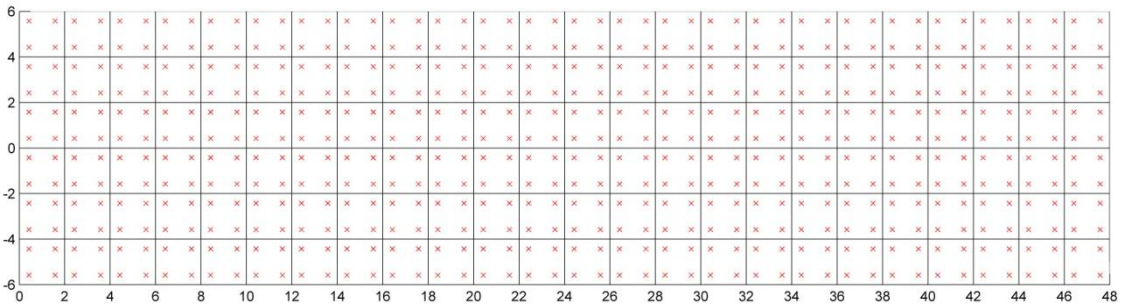


Figure 7.5
Mesh used for integration (24x6) with 2x2 Gauss points in each cell

Then the T-Belytschko's stresses are calculated, using an extended support domain 10% wider than the initial one. The local errors between the initial stresses and the T-Belytschko's stresses are computed for every Gauss point using the Eq. (7.8). The local error field is plotted in a contour pattern in Figures 7.6 and 7.7.

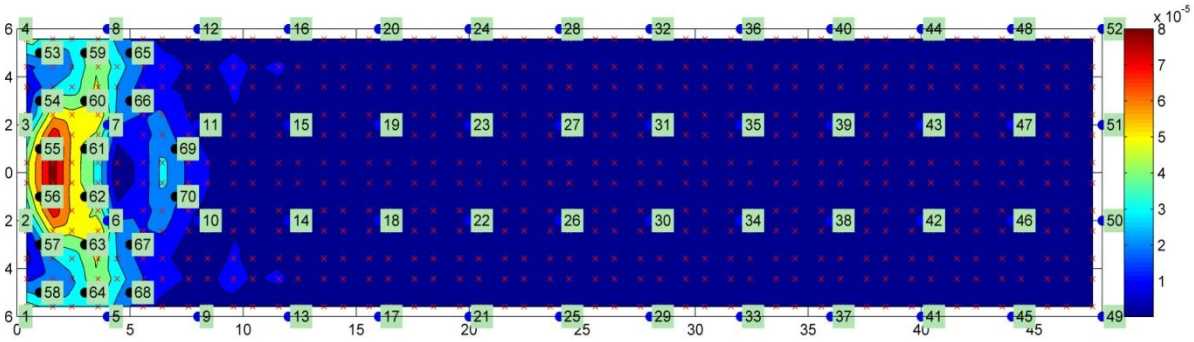


Figure 7.6
Color contour pattern showing the distribution of the local errors

As it is illustrated in both figures, after the above calculation, 18 new nodes are added in the problem domain, in correspondence to the contour lines, at the centroids of the Gauss cells of higher local error.

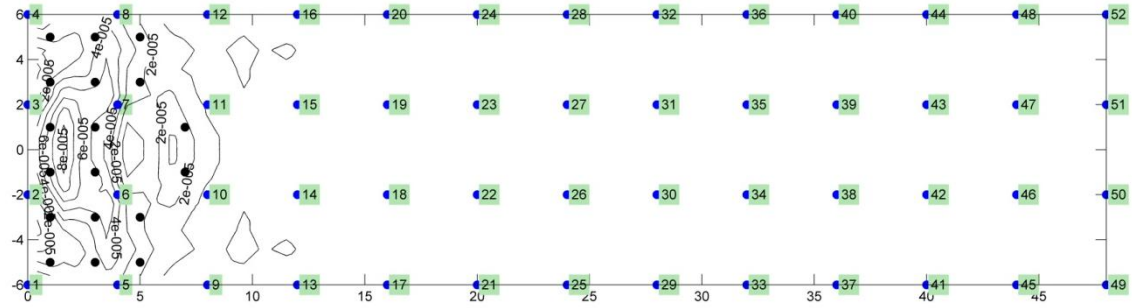


Figure 7.7
New arrangement of nodes

The addition of the extra nodes is performed with the 'hierarchical' way, described in section 6.2.3. The already calculated values of the shape functions of the first 52 nodes, are kept constant, and eighteen new forms are attached in the problem domain.
In order to estimate the improvement of the final solution, we also compute the displacements and the stress field for the cantilever beam, after the attachment of the new nodes, using the usual EFG method.
Figures 7.8 to 7.10 compare the stresses calculated using the analytical solution and the EFG with penalty method. The stresses are calculated at the cross section of x= L/4 of the beam. A good agreement is observed, and an important improvement in the shear stresses, when the hierarchical process is carried out.

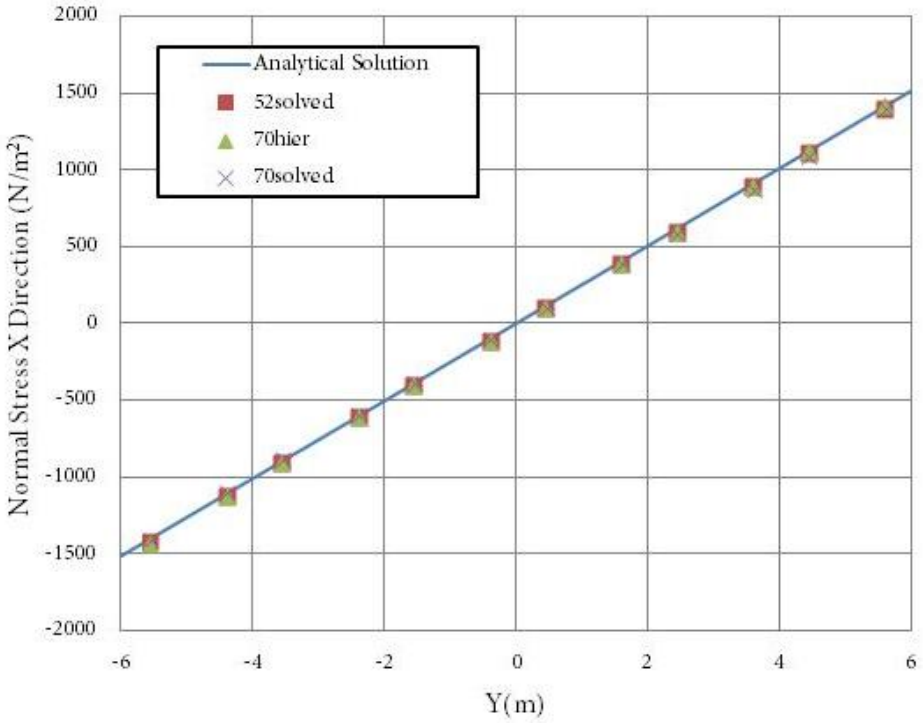


Figure 7.8

Analytical and numerical solutions for the normal stress σx at the section x= L/4 of the cantilever beam

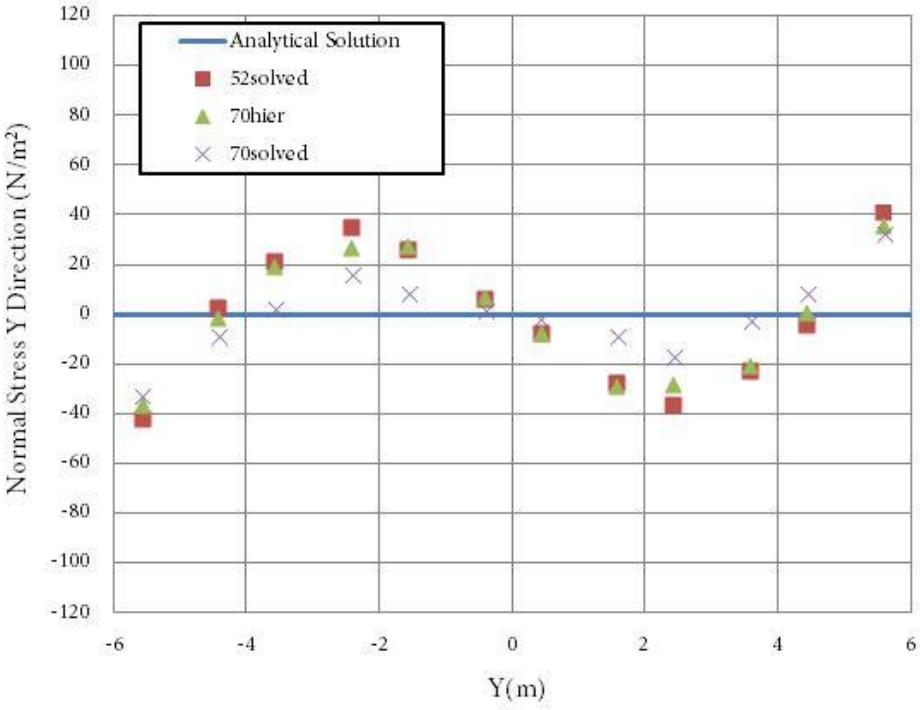


Figure 7.9

Analytical and numerical solutions for the normal stress σ_y at the section x= L/4 of the cantilever beam

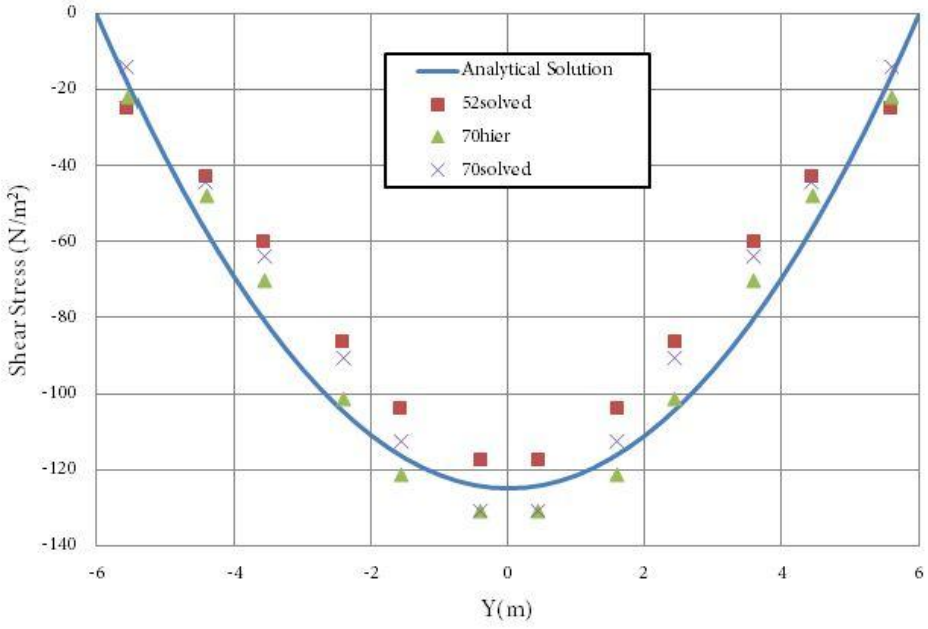


Figure 7.10

Analytical and numerical solutions for the shear stress σ_{xy} at the section x= L/4 of the cantilever beam

The energy norm and the displacement norm are again determined, to estimate the error of the analysis before and after the addition of new nodes, using Eq. (7.1) and Eq. (7.2). After that the same norms are calculated to define the error between the hierarchical and the final EGF solution according to Eq. (7.3) and Eq. (7.4). The quality of the hierarchical process, and the level of improvement in the displacement and stress field, is measured by the error index θ_h, which is defined in Eq. (7.5). The energy norms and the displacement norms for the particular example, as well as the respective error indexes can be seen in Tables 7-1 and 7-2.

Energy norm (%)			
	Initial sol(52nod)-Reference sol(70nod)	Hierarch. sol (70nod)-Reference sol(70nod)	error index θ_h
	14,24	4,41	0,31

Table 7-1

Energy norm for the cantilever beam

Displacement Norm			
	Initial sol(52nod)-Reference sol(70nod)	Hierarch. sol (70nod)-Reference sol(70nod)	error index θ_h
	4,45E-05	7,46E-07	0,017

Table 7-2

Displacement norm for the cantilever beam

Finally in Figures 7.11 and 7.12 the normal stress field in y direction, and the shear stress field for each case are illustrated. We can observe an important improvement in the profile of the stress field, especially near the support region.

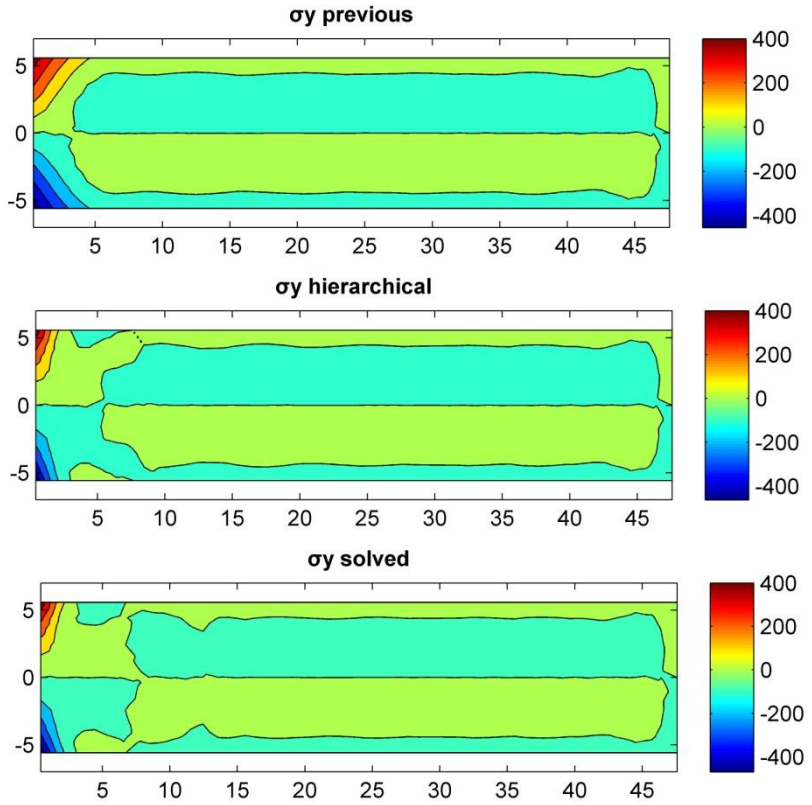


Figure 7.11
Normal stress field in y direction

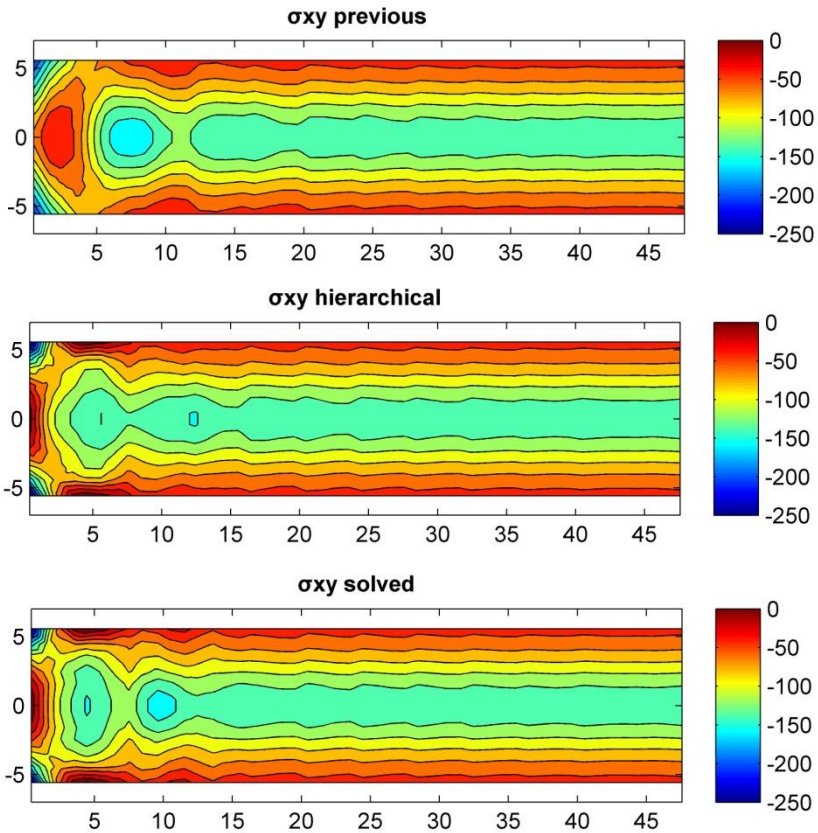


Figure 7.12
Shear stress field for the cantilever beam

7.3.2 L-shaped domain

7.3.2.1 Example 1

In this section we introduce the L-shaped domain, illustrated in Figure 7.12. This benchmark problem, is very important, as it offers stress concentration at a certain sub domain, which can lead to high energy norms. The parameters for this domain are taken as follows:

Loading	P=1500 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Width of the L-shaped domain	a=6 m
Thickness	t=0.2

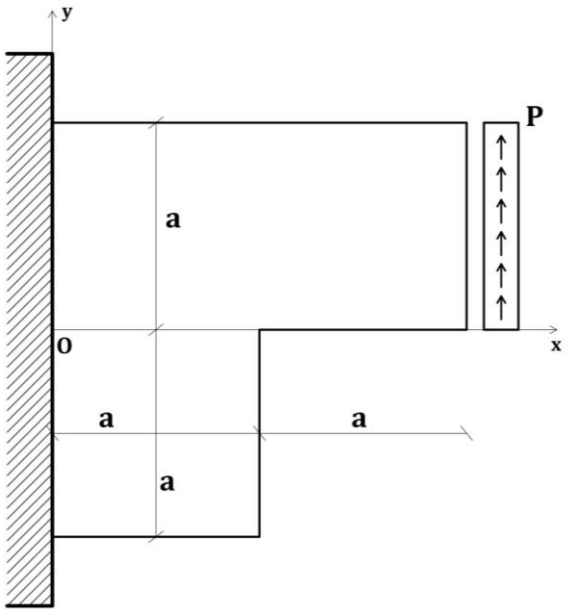


Figure 7.13
L-shaped domain

The regular node distribution is used. Firstly the beam is solved for 96 nodes, uniformly arranged. The displacements for every node and the stress field are calculated. The set of nodes and the deformed view of the domain are plotted in Figures 7.14 and 7.15 respectively. A scale factor is used for better illustration of the deformed view.
A linear basis and a cubic spline weight function are used for the MLS approximation. The support domain is rectangular with a dimension of 2.2 times the nodal spacing. The penalty factor was chosen to be $a = 10^{10}$.
After this first analysis, the T-Belytschko's stresses are computed, using an extended support domain 10% wider than the initial one. The local errors between the initial stresses and the T-Belytschko's stresses are calculated for every Gauss point using the Eq. (7.8).

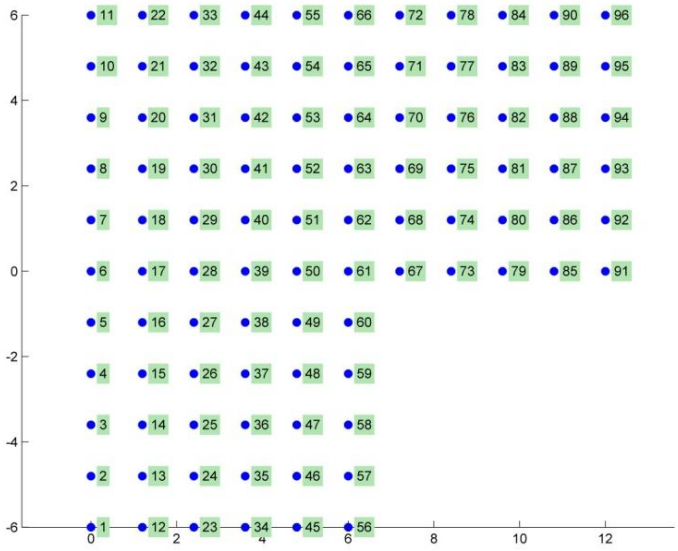


Figure 7.14
Initial arrangement of nodes

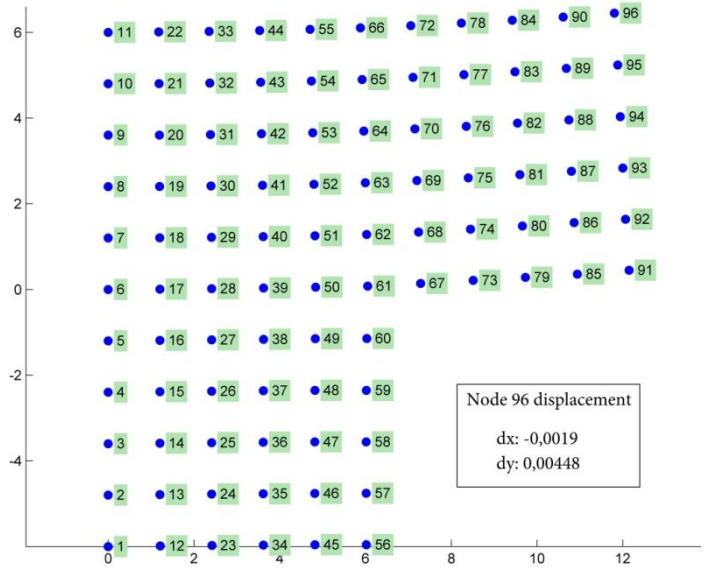


Figure 7.15
Deformed view of the L-shaped domain

Carrying out the procedure described in section 7.2.1 eight (8) new nodes are attached in the problem domain. The addition of the extra nodes is performed with the 'hierarchical' process, described in section 6.2.3. The already calculated values of the shape functions of the first 96 nodes, are kept constant, and eight new forms are attached in the problem domain. A reduced support domain 15% smaller than the initial is used for the extra nodes, in order to get a more accurate solution, as indicated in section 7.2.2.The extra nodes are plotted with red color in Figure 7.16.

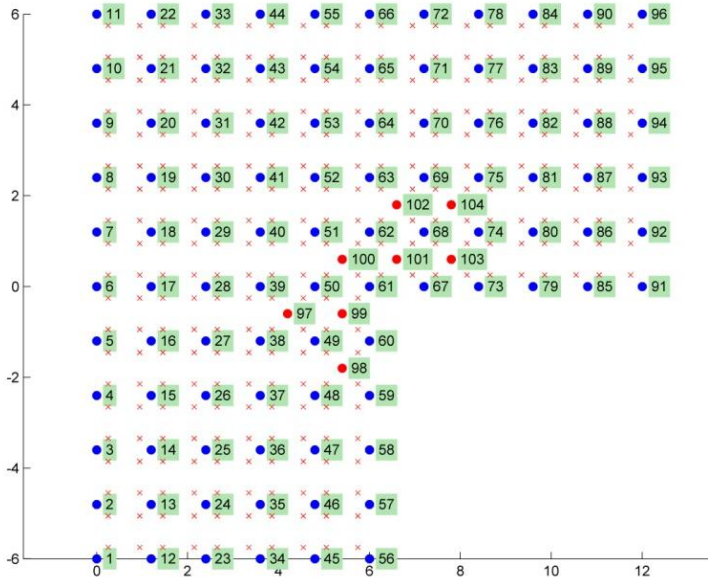


Figure 7.16
New arrangement of nodes

In order to estimate the improvement of the final solution, we also compute the displacements and the stress field for the L-shaped problem domain, after the attachment of the new nodes, using the usual EFG method.
The energy norm and the displacement norm are again determined, to estimate the error of the analysis before and after the attachment of new nodes, using Eq. (7.1) and Eq. (7.2). After that the same norms are calculated to define the error between the hierarchical and the final EGF solution according to Eq. (7.3) and Eq. (7.4). The quality of the hierarchical process, and the level of improvement in the displacement and stress field, can be measured by the error index θ_h, which is defined in Eq. (7.5). The energy norms and the displacement norms, as well as the respective error indexes can be seen in Tables 7-3 and 7-4.

Energy norm (%)			
	Initial sol(96nod)-Reference sol(104nod)	Hierarch. sol (104nod)-Reference sol(104nod)	error index θ_h
	42,39	13,45	0,32

Table 7-3
Energy norm for the L-shaped domain

Displacement Norm			
	Initial sol(96nod)-Reference sol(104nod)	Hierarch. sol (104nod)-Reference sol(104nod)	error index θ_h
	1,22E-05	9,08E-06	0,74

Table 7-4
Displacement norm for the L-shaped domain

Finally in Figures 7.17 and 7.18 the normal stress field in y direction, and the shear stress field for each case are illustrated. We can observe an important improvement in the profile of the stress field, especially in the area of high stress density.

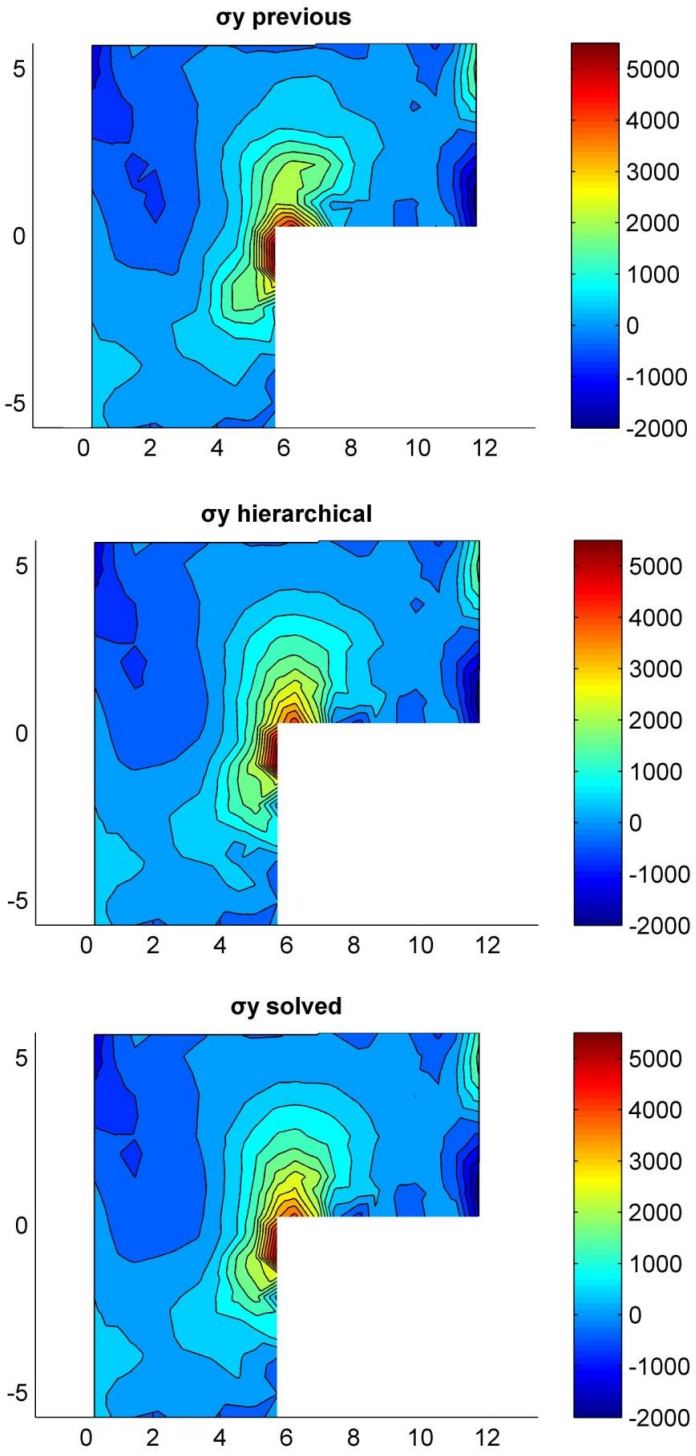


Figure 7.17
Normal stress field in y direction

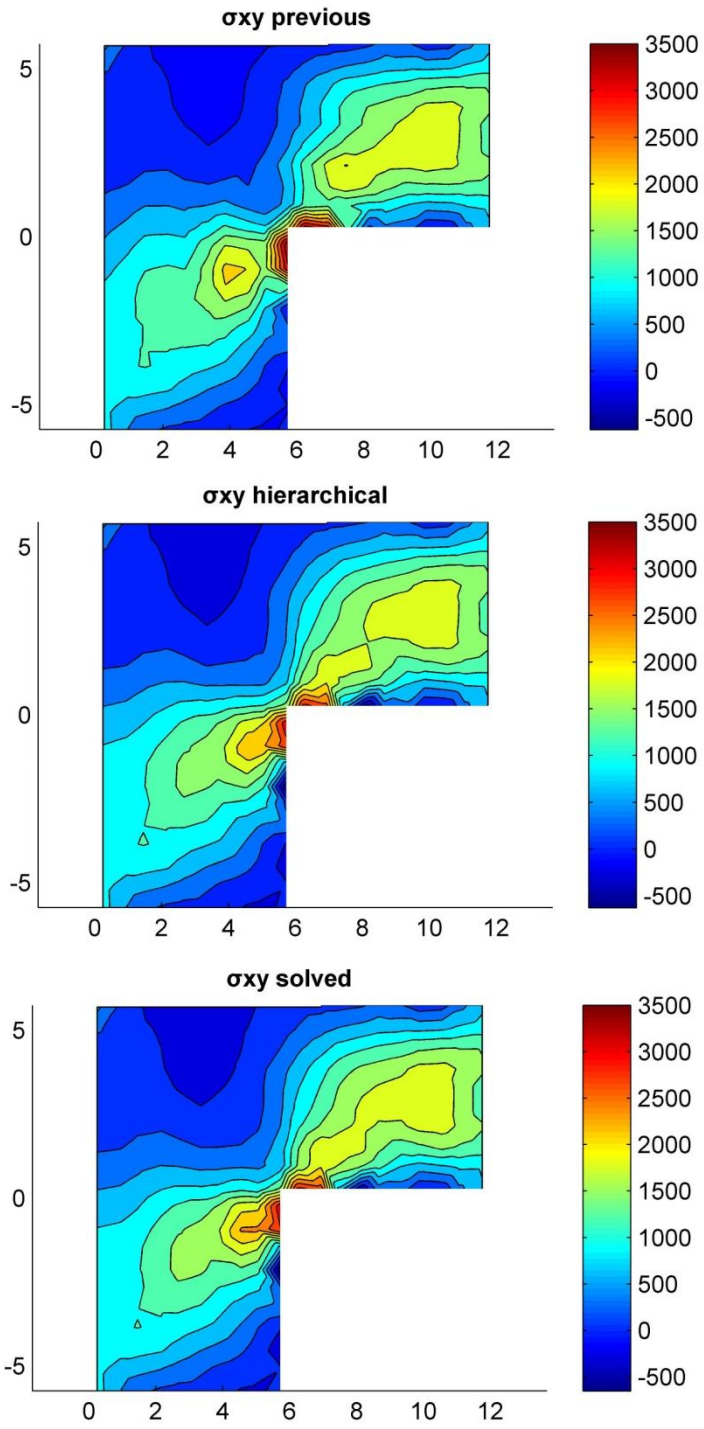


Figure 7.18
Shear stress field

7.3.2.2 Example 2

For this example we consider the L-shaped domain illustrated in Figure 7.19. The parameters for this domain are taken as follows:

Loading	P=2000 N
Young's modulus	E=3 x 10^7 N/m^2
Poisson's ratio	ν=0.30
Width of the L-shaped domain	a=6 m
Thickness	t=0.3

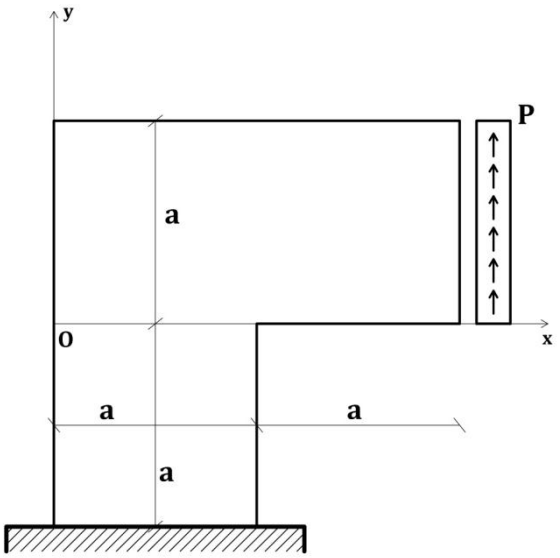


Figure 7.19
L-shaped domain

Firstly the beam is solved for 225 nodes, uniformly arranged. The displacements for every node and the stress field are calculated. The set of nodes and the deformed view of the domain are plotted in Figures 7.20 and 7.21 respectively. A scale factor is used for better illustration of the deformed view. The support domain is rectangular with a dimension of 2.4 times the nodal spacing. The penalty factor was chosen to be $a = 10^{11}$.
After the first analysis, the T-Belytschko's stresses are computed, using an extended support domain 10% wider than the initial one. The local errors between the initial stresses and the T-Belytschko's stresses are calculated for every Gauss point using the Eq. (7.8).

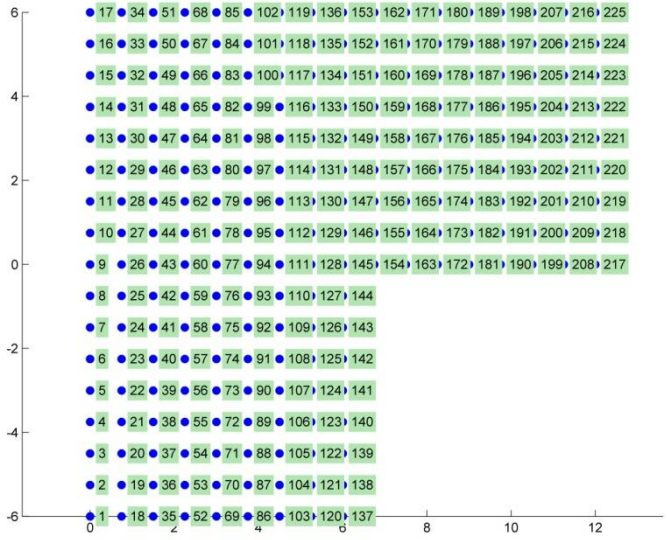


Figure 7.20
Initial arrangement of nodes

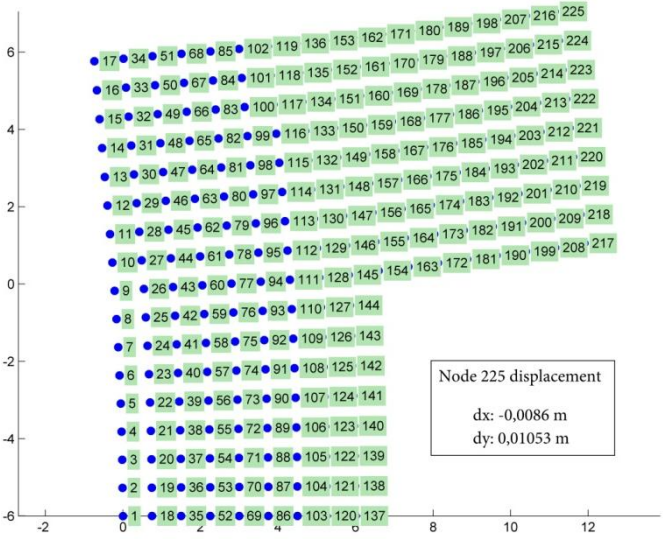


Figure 7.21
Deformed view of the L-shaped domain

Carrying out the procedure described in section 7.2.1 eighteen (18) new nodes are attached in the problem domain. The addition of the extra nodes is performed with the 'hierarchical' process, described in section 6.2.3. A reduced support domain 15% smaller than the initial is used for the extra nodes, in order to get a more accurate solution, as indicated in section 7.2.2.The extra nodes are plotted with red color in Figure 7.22.

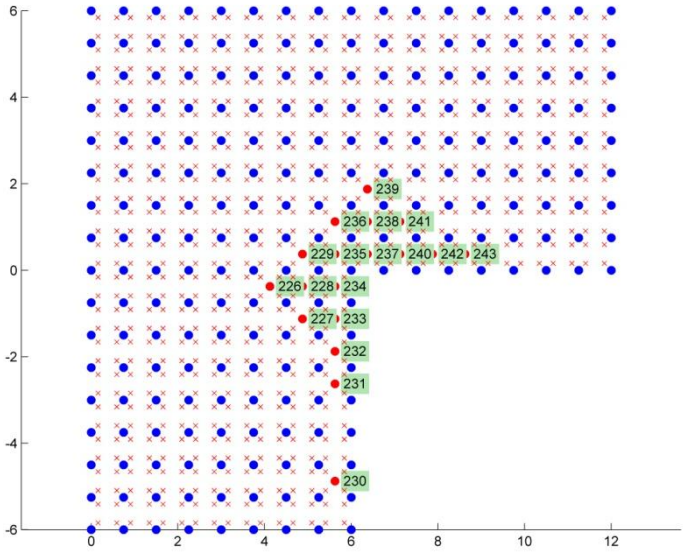


Figure 7.22
New arrangement of nodes

As seen in Example 1, we also compute the displacements and the stress field for the L-shaped problem domain, after the attachment of the new nodes, using the usual EFG method, in order to estimate the improvement of the final solution.
The energy norms and the displacement norms, as well as the respective error indexes can be seen in Tables 7-5 and 7-6.

Energy norm (%)			
	Initial sol(225nod)-Reference sol(243nod)	Hierarch. sol (243nod)-Reference sol(243nod)	error index θ_h
	45,07	15,70	0,35

Table 7-5
Energy norm for the L-shaped domain

Displacement Norm			
	Initial sol(225nod)-Reference sol(243nod)	Hierarch. sol (243nod)-Reference sol(243nod)	error index θ_h
	1,83E-05	9,74E-06	0,53

Table 7-6
Displacement norm for the L-shaped domain

Finally in Figure 7.23 the shear stress field for each case is illustrated. We can observe an important improvement in the profile of the stress field, especially in the area of high stress density.

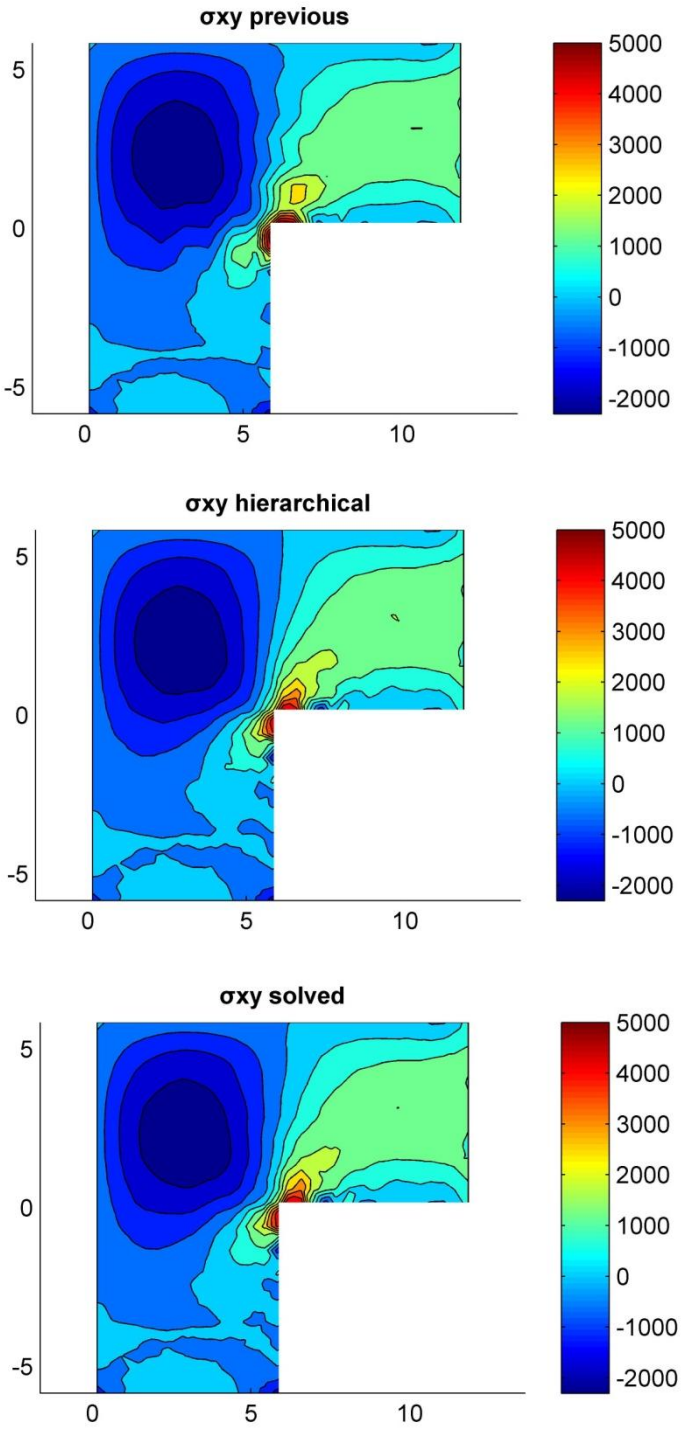


Figure 7.23
Shear stress field

8 Conclusions

The conclusions drawn from this diploma thesis, can be divided into two categories:

- Conclusions regarding the Element Free Galerkin (EFG) method, applied for solid mechanics applications. In this part we are referred to the effectiveness of the method, and other important characteristics such as the rate of convergence and implementation aspects.
- Conclusions concerning the Hierarchical concepts for the EFG method that were introduced in this diploma thesis.

8.1 Conclusions from the use of the EFG method

1. We have given an overview of the EFG method, which is based on a global weak form, with emphasis to implementation aspects. The EFG method was applied to linear elastic material problems, with great success, giving a good agreement with analytical solutions. In fact, one of the most startling remarks in this study, is the high rate of convergence which was observed, compared to the one of the Finite Element Method (Linear elements).
2. The Moving Least Square approximation was introduced for the creation of the shape functions. Special attention was given to the choice of the size of the support domain and of the appropriate weight function. Through this process we managed to understand the reason why a structured predefined mesh is not needed in the EFG approach. The cantilever beam benchmark problem was also solved for an irregular node distribution, leading to a very good approach of the final solution.
3. Since the EFG method does not fulfill the so-called Kronecker-Delta property, essential boundary conditions cannot be enforced as easily as in finite element methods. Two different opportunities to incorporate essential boundary conditions have been summarized, i.e. the penalty method and the Lagrange multiplier method. We have chosen to use the penalty method for the numerical examples that were presented. Imposing the essential boundary conditions in EFG method in that way seems to be easier. However some slight divergence from analytical solutions confirms that the boundary conditions are only satisfied approximately in the penalty method. The Lagrange multiplier method seems to ensure more accurate results, yet it remains more time consuming.
4. The Gauss quadrature scheme was used for integration. A background mesh is necessary for this technique. Background integration is best suited to the range of the problems we have studied i.e. small deformations. This process seems to be the most accurate way of obtaining the discrete equations.

5. Finally another advantage of the EFG method, deduced from this diploma thesis, is that no post processing for the output of strains and stresses is required. These quantities are already smooth, whereas in Finite Element Method various types of post processing are necessary, in order to obtain a smooth stress field suitable for contour plotting or color plotting.

8.2 Conclusions from the Hierarchical Concepts proposed

1. Through a mathematical process that was proposed, the expression of the shape functions of the EFG method has been separated in two parts. This identity was extended in terms of the stiffness matrix, leading to a numerical decomposition of the stiffness matrix in the EFG method. Therefore the addition of extra nodes in a problem domain, can easily be achieved by the computation of a new matrix $\boldsymbol{\delta K}$. This supplemental matrix is finally added to the already existing stiffness matrix-describing the initial arrangement of nodes in the problem domain- to give the final stiffness matrix.
2. Using the above analytical result, an hierarchical restoration of the stiffness matrix was introduced. In that way, we managed to take the addition of extra nodes into consideration, without modifying the existing stiffness matrix. This refinement technique can be used as an *a posteriori* treatment for the final results of an already calculated problem domain.
3. The hierarchical refinement process was performed by a simple and automatic procedure. In this procedure, the Zienkiewicz and Zhu error estimator based on the T-Belytschko stress recovery scheme is employed for the a posteriori error estimation. New nodes are added into the centroid of the background Gauss cells, located in areas of high local error. This technique seems to give good results, however it can be improved if the addition of extra nodes is also linked to the initial arrangement of nodes in the problem domain.
4. Last but not least, if we look at the error indexes of the problems solved, hierarchical refinement seems to give very good results, improving significally the displacement and the stress field of certain problem domains. If we now think of the low computational cost required for this refinement process, hierarchical refinement technique looks very promising.

8.3 Problems, prospects, and proposals for future research

The advantages of the EFG method however do not come without some cost. What is more, the proposed refinement process seems to give promising results, yet more effort needs to be made in order to solve some occurring problems. Such problems along with proposals for future research are given below.

- First of all, the EFG method is more cumbersome to implement and more computationally expensive compared to the Finite Element Method. In FEM, each Gauss point is typically involved in element-level computations for the formation of the element stiffness matrix which is then added to the appropriate positions of the global stiffness matrix. In EFG methods, although, the contribution of each Gauss point is directly added to the global stiffness matrix while the shape functions are not predefined and span across larger domains with a significantly higher amount of Gauss point-node interactions. Therefore special techniques need to be applied in order to reduce the computational cost of the EFG method.
- As we have already mentioned, the EFG shape functions do not fulfill the Kronecker delta property, making it difficult to enforce directly the boundary conditions of a problem domain. In this diploma thesis, we have noticed that while we reduce the dimension of the support domain, the values of the shape functions referred to the nodes, at the point of the node, tend to unity. Therefore, some kind of local Kronecker delta property may be feasible, allowing for a direct imposition of the essential boundary condition for the EFG method. On the other hand, it is already clear, that the use of small factors for the size of the support domain can lead to numerical instabilities and relatively large numerical errors. Thus, a local alteration of the support domain is suggested, near the boundaries, to help with the direct imposition of boundary conditions. Although, a question arises for a smooth transition between the areas with different support domains. Ideas can be found in *methods coupling mesh free and mesh-based* approximations that are already proposed.
- The construction of an underlying zone structure for purposes of numerical quadrature can be awkward if there are large differences in node spacing within one mesh. Background integration needs to be handled with special care, especially in cases where the boundaries of the problem domain are not well shaped. Work has already been done, to propose methods such as the S-PIM (Smoothed point interpolation method), making the integration in mesh free methods independent of any background mesh. Yet, extra attention needs to be paid, in the use of such mesh free methods, in order to obtain a smooth stress field, which is contrariwise ensured by the EFG method.
- The proposed hierarchical refinement technique seems to give very good results. However, when extra points are added, the partition of unity is no longer fulfilled in areas where the additional shape functions and the existing ones overlap. Consequently, it may be vital to consider ways of imposing the partition of unity in these certain areas. To minimize this effect and get more accurate solutions we have proposed a reduction of the size of the support domain of newly added nodes. Nevertheless, the potential benefits in many problems are so attractive that this technique deserves consideration.

- Finally, as it is already referred, the mathematical process that led to the total restoration of the stiffness matrix, when new nodes are fit in the problem domain, has led to a numerical decomposition of the EFG method stiffness matrix. This result can probably be of use for the parallelization of the formulation of the stiffness matrix, promising to accelerate this process. This diploma thesis proposes this idea for future research.

References

Adjerid, S., Aiffa, M., & Flaherty, J. E. (2001). Hierarchical finite element bases for triangular and tetrahedral elements. *Computer Methods in Applied Mechanics and Engineering* (190), pp. 2925–2941.

Afshar, M. H., Amani, J., & Naisipour, M. (2012). A node enrichment adaptive refinement in Discrete Least Squares Meshless method for solution of elasticity problems. *Engineering Analysis with Boundary Elements* (36), pp. 385–393.

Atluri, S. N., & Zhu, T. (1998). A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* (22), pp. 117-127.

Augarde, C. E., & Deeks, A. J. (2008). The use of Timoshenko's exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design* (44), pp. 595–601.

Barbieri, E., Petrinic, N., Meo, M., & Tagarielli, V. L. (2011). A new weight-function enrichment in meshless methods for multiple cracks in linear elasticity. *International Journal for Numerical Methods in Engineering* (90), pp. 177–195.

Belytschko, T., & Chung, H. -J. (1998). An error estimate in the EFG method. *Computational Mechanics* (21), pp. 91-100.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., & Krysl, P. (1996). Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* (139), pp. 3–47.

Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* (37), pp. 229–256.

Belytschko, T., Organ, D., & Krongauz, Y. (1995). A coupled finite element-element-free Galerkin method. *Computational Mechanics* (17), pp. 186-195.

Bochev, P. B., & Gunzburger, M. D. (2009). *Least-Squares Finite Element Methods-Applied Mathematical Sciences.* New York: Springer.

Bui, T. Q. (n.d.). Meshless dual analysis. *University of Siegen* .

Chen, Y., Lee, J. D., & Eskandarian, A. (2006). *Meshless Methods in Solid Mechanics.* Springer.

Fries, T. P., & Matthies, H. G. (2004). Classification and Overview of Meshfree Methods. *Institut für Wissenschaftliches Rechnen* .

Huang, X., Tian, Q., Mao, J., Jiang, L., & Liang, R. (2010). The Theory and Application of an Adaptive Moving Least Squares for Non-uniform Samples. *WSEAS Transactions on Computers* (9), pp. 686-695.

Hughes, T. J. (1987). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Englewood Cliffs, New Jersey: Prentice-Hall.

Karatarakis, A., Metsis, P., & Papadrakakis, M. (2013). GPU-acceleration of stiffness matrix calculation and efficient initialization of EFG meshless methods. *Computer Methods in Applied Mechanics and Engineering* (258), pp. 63–80.

Lee, C. K., & Zhou, C. E. (2004). On error estimation and adaptive refinement for element free Galerkin method: Part I: stress recovery and a posteriori error estimation. *Computers and Structures* (82), pp. 413–428.

Lee, C. K., & Zhou, C. E. (2004). On error estimation and adaptive refinement for element free Galerkin method: Part II: adaptive refinement. *Computers and Structures* (82), pp. 429–443.

Li, S., & Liu, W. K. (1999). Reproducing kernel hierarchical partition of unity, Part I—formulation and theory. *International Journal for Numerical Methods in Engineering* (45), pp. 251–288.

Liu, G. R. (2003). *Mesh free methods : moving beyond the finite element method.* CRC Press LLC.

Liu, G. R., & Gu, Y. T. (2001). A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering* (50), pp. 937–951.

Metsis, P., & Papadrakakis, M. (2012). Overlapping and non-overlapping domain decomposition methods for large-scale meshless EFG simulations. *Computer Methods in Applied Mechanics and Engineering* (229–232), pp. 128–141.

Nguyen, V. P., Rabczuk, T., Bordas, S., & Duflot, M. (2008). Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* (79), pp. 763–813.

Rabczuk, T., & Belytschko, T. (2004). Cracking particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering* (61), pp. 2316–2343.

Razmjoo, H., Movahhedi, M., & Hakimi, A. (2012). An improved truly meshless method based on a new shape function and nodal integration. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* (25), pp. 441–453.

Timoshenko, S. P., & Goodier, J. N. (1951). *Theory of Elasticity.* McGraw-Hill Book Company, Inc.

Zienkiewicz, O. C., & Zhu, J. Z. (1987). A simple error estimator and adaptive procedure for practical engineerng analysis. *International Journal for Numerical Methods in Engineering* (24), pp. 337-357.

Zienkiewicz, O. C., Gago, J. D., & Kelly, D. W. (1983). The hierarchical concept in finite element analysis. *Computers & Structures* (16), pp. 53–65.

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2005). *The Finite Element Method Its Basis and Fundamentals* (6th ed.). Elsevier Butterworth-Heinemann.

Παπαδρακάκης, Μ. (2001). *Ανάλυση Φορέων με τη Μέθοδο των Πεπερασμένων Στοιχείων*. Εκδόσεις Παπασωτηρίου.

Εκτεταμένη περίληψη

Στις μέρες μας, θέματα που σχετίζονται με την μοντελοποίηση και την προσομοίωση, διαδραματίζουν ένα σημαντικό ρόλο στην οικοδόμηση και την εξέλιξη των υπολογιστικών συστημάτων στον τομέα του μηχανικού. Η μέθοδος των πεπερασμένων στοιχείων υπήρξε η κυρίαρχη μέθοδος της υπολογιστικής μηχανικής κατά τις τελευταίες δεκαετίες, ενώ σημαντική παραμένει η συνεισφορά της στην επιστήμη, την τεχνολογία και την βιομηχανία. Παρόλα αυτά, και στην μέθοδο των πεπερασμένων στοιχείων υπάρχουν περιορισμοί, οι οποίοι μπορούν να γίνουν ορατοί κατά την επίλυση προβλημάτων μεγάλων μετατοπίσεων, ή κατά την προσομοίωση ασυνεχειών, όπως η διάδοση τυχούσας ρωγμής σε ένα συνεχές μέσο. Συνήθως τέτοια προβλήματα αντιμετωπίζονται από την μέθοδο των πεπερασμένων στοιχείων, με την πύκνωση του δικτύου στις συγκεκριμένες περιοχές του φορέα, μία χρονοβόρα διαδικασία, που θα πρέπει να εκτελείται με ιδιαίτερη προσοχή, ώστε να διατηρηθεί η καλή κατάσταση του αρχικού πλέγματος. Παρόλα αυτά, μερικές φορές φαντάζει πολύ δύσκολο έως αδύνατο, να ξεπεραστούν αυτές οι δυσκολίες που προκύπτουν από την παρουσία δικτύου.

Ετσι λοιπόν η ανάπτυξη μη πλεγματικών μεθόδων προσομοίωσης, έχει προσελκύσει το ενδιαφέρον μεγάλου μέρους της επιστημονικής κοινότητας τα τελευταία χρόνια. Οι μη πλεγματικές μέθοδοι διακριτοποιούν ένα συνεχές μέσο χρησιμοποιώντας έναν πεπερασμένο αριθμό σημείων, ενώ η προσέγγιση που προκύπτει βασίζεται εξ ολοκλήρου στα σημεία αυτά. Κατά την εφαρμογή των μεθόδων αυτών, δεν υπάρχει ανάγκη δημιουργίας δικτύου, ούτε η παρουσία στοιχείων κατά την προσομοίωση ενός φορέα. Μέχρι στιγμής μεγάλη πρόοδος έχει συντελεστεί στον τομέα της ανάπτυξης μη πλεγματικών μεθόδων, ενώ παράλληλα πολλές πτυχές τους επιδέχονται βελτίωσης. Είναι προφανές ότι οι παραπάνω μέθοδοι προσφέρουν ένα ευρύ φάσμα εφαρμογών, καθώς όπως φαίνεται μπορούν να παρέχουν λύσεις σε πολλά σύγχρονα υπολογιστικά προβλήματα.

1 Διάρθρωση της διπλωματικής εργασίας

1.1 Επιλογή μη πλεγματικής μεθόδου

Η παρούσα διπλωματική εργασία επικεντρώνεται και χρησιμοποιεί την μέθοδο element free Galerkin (EFG). Η μέθοδος EFG είναι μια μη πλεγματική μέθοδος που αναπτύχθηκε από τον T.Belytschko (Belytschko et al. -1994b) και βασίζεται στη μέθοδο διάχυσης στοιχείων (Diffuse Elements Method-DEM, Nayroles et al. -1992). Τα κύρια χαρακτηριστικά της μεθόδου EFG είναι τα εξής:

- Η μέθοδος των Κυλιόμενων Ελαχίστων Τετραγώνων (Moving least square approximation-MLS) χρησιμοποιείται για την δημιουργία την συναρτήσεων σχήματος.
- Η μέθοδος Galerkin εφαρμόζεται για την παραγωγή του συστήματος εξισώσεων.
- Ενα σύνολο βοηθητικών κελιών (κελιά Gauss), χρησιμοποιούνται για την πραγματοποίηση της αριθμητικής ολοκλήρωσης Gauss της εξίσωσης κίνησης.

1.2 Ιεραρχική μέθοδος προσαρμογής

Στην παρούσα διπλωματική εργασία προτείνεται μια τεχνική προσαρμογής για την μη πλεγματική μέθοδο EFG. Δεδομένου του ότι δεν είναι αναγκαία η ύπαρξη ενός προκαθορισμένου δικτύου, η προσαρμογή στις μη πλεγματικές μεθόδους μπορεί να γίνει εύκολα με την διασπορά τυχαίων κόμβων στις περιοχές όπου παρουσιάζεται συγκέντρωση σφάλματος. Η διαδικασία επιτυγχάνεται χωρίς την ανάγκη τροποποίησης των ήδη υπαρχόντων συναρτήσεων σχήματος του προβλήματος. Αυτό οδηγεί σε μείωση του υπολογιστικού κόστους, καθώς το αρχικό μητρώο δυσκαμψίας του προβλήματος, παραμένει σταθερό και δεν επαναϋπολογίζεται. Η μέθοδος προσαρμογής θα καλείται *ιεραρχική* καθώς βασίζεται σε ιεραρχικές μορφές των συναρτήσεων σχήματος που προτείνονται στη διπλωματική αυτή εργασία. Προκειμένου να διευκρινιστεί η παραπάνω τεχνική, παρουσιάζεται το θεωρητικό της υπόβαθρο ενώ επιλύονται προβλήματα επίπεδης ελαστικότητας.

2 Βασικές Αρχές

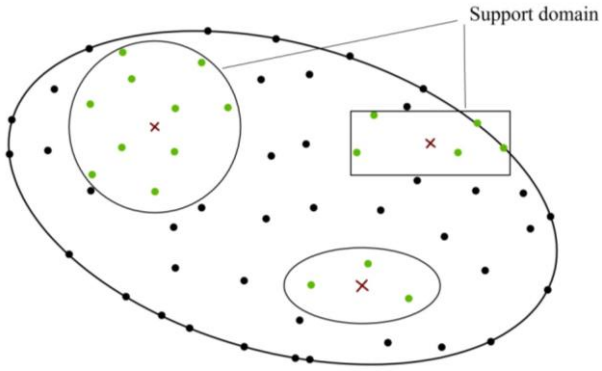
Το διάνυσμα των μετατοπίσεων ενός τυχαίου σημείου **x** = (x, y, z) του φορέα προσεγγίζεται χρησιμοποιώντας τις μετατοπίσεις σε κάθε κόμβο, εντός της περιοχής επιρροής του σημείου **x**, δηλαδή:

$$u(\mathbf{x},t)=\sum_{I\in S}\Phi_I(\mathrm{x})\,\mathrm{u}_I(\mathrm{t}) \tag{1}$$

όπου $\Phi_I : \Omega \rightarrow \mathbb{R}$ είναι οι συναρτήσεις σχήματος, u_I το διάνυσμα των μετατοπίσεων του κόμβου *I*, και S το σύνολο των κόμβων για τους οποίους $\Phi_I(x) \neq 0$.

2.1 Η ιδέα της περιοχής επιρροής

Η περιοχή επιρροής ενός σημείου **x** εντός του φορέα, καθορίζει τον αριθμό των κόμβων που είναι απαραίτητοι για να προσδιοριστεί η τιμή μιας συνάρτησης στη θέση **x**. Η περιοχή επιρροής μπορεί να είναι σταθμισμένη μέσω μιας *συνάρτησης βάρους*. Μπορεί να έχει διαφορετικές διαστάσεις και σχήματα ενώ μπορεί να είναι διαφορετική για διαφορετικά σημεία αναφοράς **x**, όπως φαίνεται στη Εικόνα 1.



Εικόνα 1

Οι περιοχές επιρροής απεικονίζονται με διαφορετικά σχήματα. Πιο συχνά χρησιμοποιούνται κυκλικά ή ορθογώνια σχήματα.

2.2 Συναρτήσεις βάρους

Οι συναρτήσεις βάρους παίζουν έναν πολύ σημαντικό ρόλο στην απόδοση της μεθόδου. Θα εξετάσουμε συναρτήσεις βάρους οι οποίες εξαρτώνται μόνο από την απόσταση μεταξύ δύο σημείων, ενός τυχαίου σημείου στη θέση **x** εντός του φορέα και ενός κόμβου αναφοράς στη θέση $\mathbf{x}_I$. Οι πιο συνηθισμένες συναρτήσεις βάρους, που χρησιμοποιούνται και σε αυτή την εργασία είναι :

- η συνάρτηση βάρους cubic spline

$$w(r)=\begin{cases}\frac{2}{3}-4r^2+4r^3, & r\le\frac{1}{2}\\ \frac{4}{3}-4r+4r^2-\frac{4}{3}r^3, & \frac{1}{2}<r<1\\ 0, & r\ge 1\end{cases} \tag{2}$$

- η συνάρτηση βάρους quartic spline

$$w(r)=\begin{cases}1-6r^2+8r^3-3r^4, & r<1\\ 0, & r\ge 1\end{cases} \tag{3}$$

όπου $r=\frac{\|\mathbf{x}-\mathbf{x}_I\|}{d_s}$ και d_s το μέγεθος της περιοχής επιρροής του κόμβου I.

2.3 Η Μέθοδος των Κυλιόμενων Ελαχίστων Τετραγώνων

Η προσέγγιση $u^h : \Omega \to \mathbb{R}$ της συνάρτησης $u : \Omega \to \mathbb{R}$ τίθεται ως πολυώνυμο m τάξης, με μη σταθερούς συντελεστές, ως εξής :

$$u^h(\mathbf{x}) = \sum_i^m p_i(x) a_i(x) = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{4}$$

όπου m είναι ο αριθμός των όρων του πολυωνύμου (πολυώνυμο βάσης) που δίνεται από τη σχέση :

$$\mathbf{p}^T(x) = \left\{1,\, x,\, x^2, \ldots, \mathrm{x}^m\right\} \tag{5}$$

και **a(x)** είναι ένα διάνυσμα συντελεστών :

$$\mathbf{a^T(x)} = \left\{a_0(x)\; a_1(x) \ldots a_m(x)\right\} \tag{6}$$

Οι άγνωστοι συντελεστές $\mathbf{a}_i\mathbf{(x)}$ προσδιορίζονται για ένα οποιοδήποτε σημείο **x**, ελαχιστοποιώντας ένα συναρτησιακό J που προσδιορίζεται από έναν σταθμισμένο μέσο όρο όλων των κόμβων $I \in \{1, \ldots, n\}$ ως εξής :

$$\begin{aligned} J &= \sum_{I=1}^{n} w(x - x_I)[\mathrm{u_h(x)} - \mathrm{u(x_I)}]^2 \\ &= \sum_{I=1}^{n} w(x - x_I)[\mathbf{p}^{\mathrm{T}}(x_I)\mathbf{a(x)} - \mathrm{u(x_I)}]^2 \end{aligned} \tag{7}$$

όπου $w(x - x_I)$ είναι μία συνάρτηση βάρους και n είναι ο αριθμός των κόμβων στην περιοχή επιρροής του x, για τους οποίους $w(x - x_I) \neq 0$.Λαμβάνοντας υπόψη την απαίτηση για ελαχιστοποίηση του συναρτησιακού J, προκύπτει :

$$\frac{\partial J}{\partial \mathbf{a}} = 0 \tag{8}$$

άρα

$$\sum_{I=1}^{n} w(x - x_I) 2\mathbf{p}(x_I)[\mathbf{p}^T(x_I)\mathbf{a}(x) - u_I] = 0 \tag{9}$$

το οποίο μετά από αναγωγές μπορεί να γραφεί ως :

$$\sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I)\mathbf{a}(x) = \sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)u_I \tag{10}$$

ή

$$\mathbf{A(x)a(x) = B(x)u} \tag{11}$$

όπου το **A** καλείται *μητρώο σταθμισμένων ροπών* και δίνεται από τη σχέση :

$$\mathbf{A}(x) = \sum_{I=1}^{n} w(x-x_I)\mathbf{p}(x_I)\mathbf{p}^T(x_I) \tag{12}$$

ενώ

$$\mathbf{B}(x) = \left[\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_n\right] \tag{13}$$

όπου

$$\mathbf{B}_I = w(x\text{-}x_I)\mathbf{p}(x_I) \tag{14}$$

επιλύοντας την Εξίσωση (11) ως προς $\mathbf{a(x)}$ και αντικαθιστώντας στην Εξίσωση (4) λαμβάνουμε :

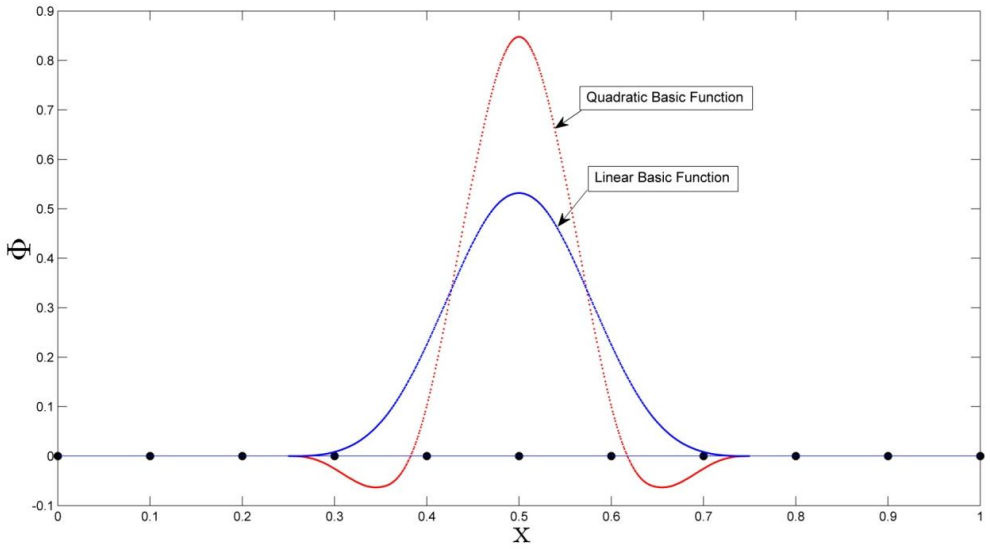
$$u^h(\mathbf{x}) = \mathbf{p^T}(x)\mathbf{a}(x) = \mathbf{p^T}(x)\mathbf{A^{-1}(x)B(x)u} \tag{15}$$

Η Εξίσωση (15) μπορεί να γραφεί συνοπτικά :

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} \varphi_I(\mathbf{x})u_I = \Phi^{\mathrm{T}}(\mathbf{x})\,\mathbf{u} \tag{16}$$

όπου $\mathbf{\Phi(x)}$ είναι το διάνυσμα των συναρτήσεων σχήματος :

$$\Phi(\mathbf{x}) = \left[\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_n(\mathbf{x})\right] \tag{17}$$



Εικόνα 2
Συναρτήσεις σχήματος για γραμμική και δευτεροβάθμια συνάρτηση βάσης.

2.4 Μέθοδος Galerkin: Ασθενής μορφή

Με την εφαρμογή της μεθόδου Galerkin η ασθενής μορφή της διαφορικής εξίσωσης κίνησης για στατικά προβλήματα, μπορεί να γραφεί σε όρους μετατοπίσεων ως :

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}}\, d\Gamma = 0 \qquad (18)$$

$\mathbf{L}$ = διαφορικός τελεστής
$\mathbf{u}$ = το διάνυσμα των επικόμβιων μετατοπίσεων
$\mathbf{b}$ = το διάνυσμα των μαζικών δράσεων
$\bar{\mathbf{t}}$ = το διάνυσμα των επιφανειακών δράσεων που επιβάλλονται στο φυσικό σύνορο Γ_t

2.5 Επιβολή συνοριακών συνθηκών-Μέθοδος ποινής

Ενα βασικό χαρακτηριστικό των συναρτήσεων σχήματος που προκύπτουν από την μέθοδο των κυλιόμενων ελαχίστων τετραγώνων (MLS), είναι ότι δεν πληρούν την ιδιότητα που εκφράζει η συνάρτηση δέλτα του Κρόνεκερ. Στα μαθηματικά, η συνάρτηση δέλτα του Κρόνεκερ, είναι μια συνάρτηση δύο μεταβλητών, συνήθως ακεραίων. Η τιμή της συνάρτησης είναι 1 αν οι μεταβλητές είναι ίσες, διαφορετικά είναι ίση με 0.
Λόγω της μη ικανοποίησης της συνάρτησης δέλτα του Κρόνεκερ των συναρτήσεων σχήματος, οι συνοριακές συνθήκες δεν μπορούν να επιβληθούν με τον ίδιο τρόπο όπως στα Πεπερασμένα Στοιχεία. Μεταξύ των διαφόρων διαθέσιμων τεχνικών για

την επιβολή των συνοριακών συνθηκών (Μέθοδος των πολλαπλασιαστών Lagrange, μέθοδος ποινής), για τις ανάγκες της παρούσας διπλωματικής εργασίας επιλέγεται η μέθοδος της ποινής. Εφαρμόζοντας την μέθοδο της ποινής στην Εξίσωση (18) η ασθενής μορφή Galerkin της διαφορικής εξίσωσης κίνησης μπορεί να γραφεί ως :

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{c}(\mathbf{Lu}) d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}}\, d\Gamma - \int_{\Omega} \delta\mathbf{C}^T(\mathbf{u})\boldsymbol{\alpha}\mathbf{C}(\mathbf{u})\, d\Omega = 0 \quad (19)$$

όπου το **α** είναι ένα διαγώνιο μητρώο που δίνεται από τη σχέση :

$$\boldsymbol{\alpha} = \begin{bmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & a_k \end{bmatrix} \quad (20)$$

με $\alpha_1, \alpha_2, \ldots, \alpha_k$ να είναι σταθεροί θετικοί συντελεστές γνωστοί ως *συντελεστές ποινής* και **C** ένα δεδομένο μητρώο συντελεστών, που προσδιορίζει την επιρροή των συνοριακών συνθηκών στον φορέα.

2.6 Ολοκλήρωση Gauss

Ενα σύνολο βοηθητικών κελιών (κελιά Gauss), χρησιμοποιούνται για την πραγματοποίηση της αριθμητικής ολοκλήρωσης Gauss της εξίσωσης κίνησης σύμφωνα με τη σχέση :

$$\int_{\Omega} f(\mathbf{X})\, \mathrm{d}\Omega = \sum_{J} f(\boldsymbol{\xi}_J)\, \mathrm{w}_J \det J^{\xi}(\boldsymbol{\xi}) \quad (21)$$

όπου **ξ** είναι οι συντεταγμένες στο φυσικό σύστημα και $\det J^{\xi}(\boldsymbol{\xi})$ είναι η ορίζουσα του Ιακωβιανού μητρώου, που υπολογίζεται για τις ανάγκες της απεικόνισης του Καρτεσιανού συστήματος στο φυσικό σύστημα.

3 Ιεραρχικές Μορφές για την Μέθοδο EFG

Η ιδέα για την παραγωγή ιεραρχικών συναρτήσεων σχήματος στην μέθοδο EFG μπορεί να γίνει κατανοητή αν εξετάσουμε το μονοδιάστατο πρόβλημα της Εικόνας 3. Η έκφραση των συναρτήσεων σχήματος για κάθε κόμβο μπορεί να γραφεί ως :

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})[\mathbf{A}(\mathbf{x})]^{-1}\mathbf{B}(x) \tag{22}$$

σχέση η οποία εφαρμόζεται για κάθε σημείο Gauss. Με την χρήση της γραμμικής συνάρτησης βάσης η Εξίσωση (22) μπορεί να γραφεί υπό όρους σημείων Gauss ως :

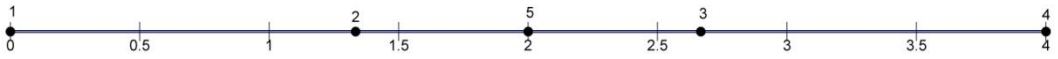
$$\underbrace{\mathbf{\Phi}(\mathbf{x}_{gp})}_{1x5} = \underbrace{\mathbf{p}^T(\mathbf{x}_{gp})}_{[1x2]}\underbrace{[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}}_{[2x2]}\underbrace{\mathbf{B}(x_{gp})}_{[2x5]} \tag{23}$$

όπου

$$\mathbf{p}^T(\mathbf{x}_{gp}) = [1 \quad x_{gp}] \tag{24}$$

$$\mathbf{A}_{1:5}(\mathbf{x}_{gp}) = \sum_{i=1}^{5} w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_i)\mathbf{p}(x_i)\mathbf{p}^T(x_i) = \sum_{i=1}^{5} w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_i)\begin{bmatrix}1\\ x_i\end{bmatrix}[1 \quad x_i] \tag{25}$$

$$\mathbf{B}(x_{gp}) = \mathbf{p}_{gp}\begin{bmatrix} w(x_{gp}-x_1) & w(x_{gp}-x_2) & w(x_{gp}-x_3) & w(x_{gp}-x_4) & w(x_{gp}-x_5)\end{bmatrix} \tag{26}$$



Εικόνα 3
Ελαστικό μονοδιάστατο πρόβλημα

Η Εξίσωση (23) μπορεί να χωριστεί σε δύο μέρη. Ο πρώτος όρος εκφράζει τις τιμές των συναρτήσεων σχήματος για τους τέσσερις πρώτους κόμβους, σε ένα συγκεκριμένο σημείο Gauss, και ο δεύτερος όρος την τιμή της συνάρτησης σχήματος για το πέμπτο κόμβο.

$$\mathbf{\Phi}_{1:5}(\mathbf{x}_{gp}) = \mathbf{\Phi}_{1:4}(\mathbf{x}_{gp}) + \mathbf{\Phi}_{5}(\mathbf{x}_{gp}) = \mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{1:4}(x) + \mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:5}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{5}(x) \tag{27}$$

Το μητρώο σταθμισμένων ροπών **A** μπορεί επίσης να χωρισθεί σε δύο μέρη σε αντιστοιχία με τους κόμβους του φορέα :

$$\mathbf{A}_{1:5}(\mathbf{x}_{gp}) = \mathbf{A}_{1:4}(\mathbf{x}_{gp}) + \mathbf{A}_{5}(\mathbf{x}_{gp}) = \sum_{i=1}^{4} w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_i)\mathbf{p}(x_i)\mathbf{p}^T(x_i) + w(\mathrm{x}_{\mathrm{gp}} - \mathrm{x}_5)\mathbf{p}(x_5)\mathbf{p}^T(x_5) \tag{28}$$

Αν τώρα χρησιμοποιήσουμε το Διωνυμικό Θεώρημα, μπορούμε να αντιστρέψουμε το μητρώο σταθμισμένων ροπών σε τμήματα :

$$\left[\mathbf{A}_{1:5}(\mathbf{x}_{gp})\right]^{-1} = \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp}) + \mathbf{A}_{5}(\mathbf{x}_{gp})\right]^{-1} = \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1} + \mathbf{X} \tag{29}$$

όπου

$$\mathbf{X} = -\left\{\mathbf{I} + \left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1}\mathbf{A}_5(\mathbf{x}_{gp})\right\}^{-1}\left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1}\mathbf{A}_5(\mathbf{x}_{gp})\left[\mathbf{A}_{1:4}(\mathbf{x}_{gp})\right]^{-1} \tag{30}$$

Συνδυάζοντας τις Εξισώσεις (30) και (27) τελικά λαμβάνουμε :

$$\boldsymbol{\Phi}_{1:4}(\mathbf{x}_{gp}) = \underbrace{\mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:4}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_{1:4}(x)}_{\boldsymbol{\Phi}_{1:4}:4\ \sigma\eta\mu\epsilon\acute{\iota}\alpha\ \delta\iota\alpha\kappa\rho\iota\tau o\pi o\acute{\iota}\eta\sigma\eta\varsigma} + \underbrace{\mathbf{p}^T(\mathbf{x}_{gp})\mathbf{X}\mathbf{B}_{1:4}(x)}_{\delta\boldsymbol{\Phi}_{1:4}:\pi\rho o\sigma\theta\acute{\eta}\kappa\eta\ 5ou\ \kappa\acute{o}\mu\beta ou} \tag{31}$$

$$\boldsymbol{\Phi}_5(\mathbf{x}_{gp}) = \mathbf{p}^T(\mathbf{x}_{gp})[\mathbf{A}_{1:4}(\mathbf{x}_{gp})]^{-1}\mathbf{B}_5(x) + \mathbf{p}^T(\mathbf{x}_{gp})\mathbf{X}\mathbf{B}_5(x) \tag{32}$$

Μέσω της παραπάνω διαδικασίας καταφέραμε να χωρίσουμε τις συναρτήσεις σχήματος σε δύο διακριτούς όρους, με τον δεύτερο να εκφράζει την τροποποίηση των ήδη υπαρχόντων συναρτήσεων σχήματος, όταν ένας επιπλέον κόμβος προτίθεται στο πρόβλημα. Αυτή η διαδικασία πρέπει να γίνεται, ώστε να διατηρηθεί η πληρότητα της περιγραφής του φορέα μέσω συναρτήσεων σχήματος. Όπως διαπιστώθηκε η παραπάνω ιδιότητα μπορεί να επεκταθεί και για τις πρώτες παραγώγους των συναρτήσεων σχήματος, άρα και για τα αντίστοιχα μητρώα δυσκαμψίας, οδηγώντας στην δυνατότητα υπολογισμού του τελικού μητρώου δυσκαμψίας ενός φορέα σε τμήματα.

Οι παραπάνω ιδέες υπήρξαν πολύ βοηθητικές και συνετέλεσαν στην εισαγωγή μιας νέας μεθόδου προσαρμογής για την μέθοδο EFG. Σύμφωνα με αυτή, οι επιπλέον βαθμοί ελευθερίας προτίθενται στον φορέα χωρίς την τροποποίηση των ήδη υπαρχόντων συναρτήσεων σχήματος. Η προσαρμογή αυτή ονομάστηκε ιεραρχική, καθώς η σύλληψή της ξεκίνησε από την ιεραρχική διατύπωση των συναρτήσεων σχήματος που έχει ήδη προταθεί για τα πεπερασμένα στοιχεία. Αποτελέσματα της εφαρμογής της μεθόδου σε προβλήματα επίπεδης ελαστικότητας, μπορούν να βρεθούν στον έβδομο κεφάλαιο την παρούσας διπλωματικής.