Higher-Order Discontinuous Modelling of Fracturing in Quasi-Brittle Materials

by

Dimitrios Kourepinis

A thesis submitted to the University of Glasgow in fulfilment of the requirements for the degree of

Doctor of Philosophy

in the field of Computational Structural Mechanics

Department of Civil Engineering
Faculty of Engineering
University of Glasgow
February 2008

Abstract

Quasi-brittle failure is characterised by material degradation, fracturing and potential interaction of fragmented parts. The computational description of this behaviour has presented significant challenges to the mechanics community over the past few decades, driven by the development of technology, the increasing social and economical constraints for safer and more complicated engineering designs and consequently by the increasing requirements for more accurate understanding of macro- and micro-structural processes.

Finite element methods have been pushed to their limits in an attempt to resolve strain localisation and ultimately fracturing in a unified and objective manner, while discrete methods have been utilised by artificial connection of discrete bodies which are identified *a priori* to act as continua. Neither of these attempts comprises a *diritta via* for modelling the transition from continuum to discontinuum efficiently and this has led to the investigation of alternative techniques.

Herein, the numerical modelling of quasi-brittle localisation and fracturing is investigated using the Numerical Manifold Method (NMM) as an alternative unifying framework to industry-established techniques such as the Finite Element Method (FEM) and Discrete Element Method (DEM).

One of the particularly interesting aspects of NMM is with respect to its potential for modelling both continuum and discontinuum states and providing an efficient framework for modelling the entire transition between continuum to discontinuum, from a continuum point of view, without remeshing. The attractive nature of this capability advocates potential for modelling mechanics of materials such as concrete, rock and masonry, but also a more general class of quasi-brittle materials.

This work investigates and extends NMM primarily with respect to the following characteristics:

1. Discontinuities, such as cracks, are introduced naturally in a discrete manner, but in a continuum setting, without the need for remeshing

2. The approximation is improved globally or locally, for any arbitrary level, without remeshing

3. Integration is undertaken explicitly, for any arbitrary level of local improvement of the approximation

Furthermore, NMM is reformulated using a constrained variational approach for generalised three-dimensional problems. Essential boundary conditions are enforced using Lagrange multipliers and projection matrices and potential higher-order boundary issues are investigated. The developments are implemented algorithmically in MATLAB and higher-order enrichment is demonstrated with the use of adaptivity.

Acknowledgments

"*Fools rush in where angels fear to tread*"

Alexander Pope

During the last stage of this research I truly started realising how significant it is to not only undertake such an endeavour, but also to strive to communicate any acquired knowledge and ideas efficiently. As a result, this manuscript has not only been the most frustrating, but also the most satisfying product of the experience. In the words of an old samurai "a warrior's way is twofold, of pen and sword, and he should have a taste for both".

My most sincere appreciation is directed to Chris Pearce and Nenad Bićanić for their ideas, guidance, belief, motivation, enthusiasm, support, understanding and wisdom. Not only did they mentor throughout this work, but they also inspired and shared tools to develop my academic and career path during my undergraduate studies at the University of Glasgow.

I am indebted to Natalie as well as friends and family for their general support. Thanks also to colleagues, both at the University of Glasgow and Halcrow, for their ideas and invaluable input. In particular, I extend my gratitude to Angus McConnell.

Furthermore, I gratefully acknowledge the support of the Engineering and Physical Sciences Research Council (Grant Ref: GR/N03365) and Halcrow Group Ltd. for a scholarship to conclude this research.

Dimitrios Kourepinis
Glasgow, February 2008

Table of contents

1 Introduction

1.1 Scope

Physical processes range significantly in complexity. Certain relatively simple problems, such as the dynamics of a one-degree-of-freedom mass-spring system, may be described adequately by simple mathematical equations. Other processes, such as the impact and fragmentation of a body on a composite structure, the three-dimensional response of a concrete arch dam subjected to seismic actions and generally phenomena that involve complex loads and geometries, moving boundaries and random material microstructures, are complicated enough that analytical solutions are not sufficient, efficient or even possible. In such cases, numerical approximation techniques provide the only efficient and economically viable way to approach a safe solution.

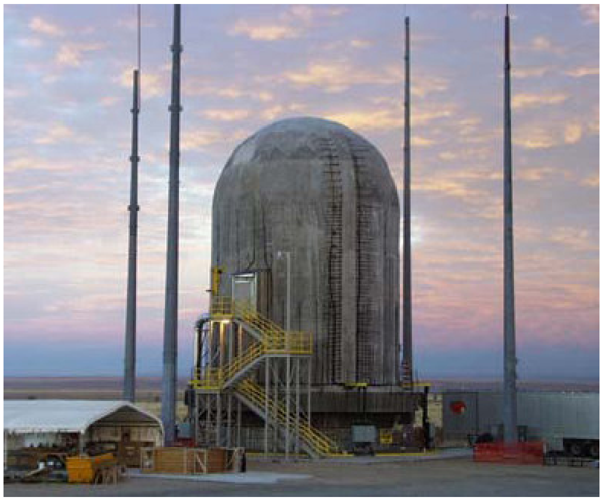


Figure 1-1. Scaled pre-stressed concrete containment vessel (PCCV), Sandia National Laboratories [88]

Numerical approximation techniques offer powerful and efficient non-destructive means to investigate complicated engineering problems in order to aid the design process, assess the performance of existing structures in present or future loading events, or substantiate forensic investigations. However, numerical tools are not without their difficulties and limitations. The numerical description of quasi-brittle failure is one such case, which has presented significant challenges to the computational mechanics community over the past few decades.

Structures composed of quasi-brittle materials and components are abundant in most parts of the world, due to availability and mechanical properties of natural or human-engineered materials such as concrete, masonry, ice, metals, rock and geomaterials. Furthermore, several special safety-critical structures such as large containment vessels (Figure 1-1), dams and large floating structures are commonly quasi-brittle.

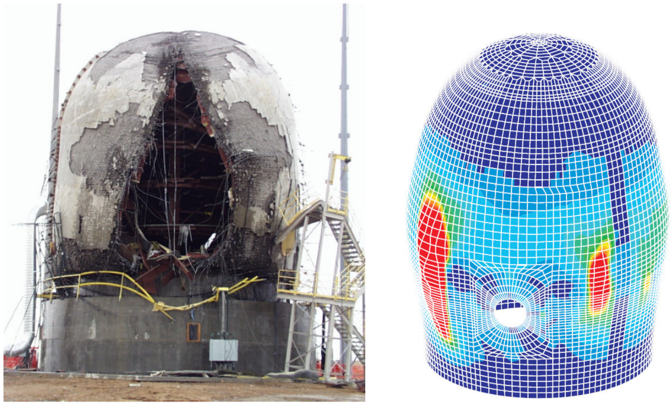


Figure 1-2. PCCV physical model after structural failure mode test (left) [89]. PCCV numerical model indicating contours of maximum principal strain (right) [90]

The failure of quasi-brittle structures is characterised by the gradual transition from continuum to discontinuum caused by strain localisation and material degradation caused by development of microcracks, their coalescence into macrocracks (fracturing) and, potentially, interaction of fragmented parts. Therefore, in order to adequately describe the behaviour of quasi-brittle structures, three key elements are required:

First, a method is required for the spatial discretisation of fracturing. In order to be adequate and versatile, this has to be capable to describe the transition from continuum to discontinuum without *a priori* assumptions regarding the location and potential path of localisation phenomena.

Second, material behaviour needs to be represented numerically on a constitutive level. Once localisation occurs, material behaviour is governed by nonlinear constitutive phenomena. It is important that this reflects the state of gradual localised degradation before failure (fracturing) occurs; for example, the case of a partially-developed crack before it becomes fully open.

Third, the approximation has to be accurate and potentially adaptive in order to predict initiation and propagation of discontinuities with minimum error. The solution has to be robust and free from numerical instabilities. In addition, computational efficiency is a key practical issue as it is directly associated with the performance of available computer resources and it is affected by the scale, importance and time constraints.

The presence or appearance of discontinuities and local failure phenomena are not necessarily associated with complete structural failure or catastrophic collapse. Examples are pre-existing joints in structural rock and allowed thermal cracks in concrete structures that do not necessarily hinder the serviceability state. However, negligence of such allowable discontinuities from numerical models can, in certain cases lead ultimately to unconservative and potentially unsafe solutions due to over-estimation of structural capacity.

The primary aim of this research is to investigate and develop a unified method for modelling discontinuities in quasi-brittle structures as an alternative framework to techniques such as the Finite Element Method (FEM) and Discrete Element Method (DEM). Both FEM and DEM are established in several different engineering industries. However, FEM is continuum-based and consequently has been traditionally more attractive for modelling localisation phenomena, whereas DEM employs discrete interconnected bodies which are specified *a priori*, and hence has been more attractive for modelling problems involving discontinua.

Particular emphasis is given to the investigation and development of the Numerical Manifold Method (NMM), as the framework of choice for undertaking this research. NMM is not established or developed as much as FEM and DEM, and it has so far been applied almost exclusively by geotechnical communities in Japan, U. S. and China. Nevertheless, there are attractive aspects that reflect the method's potential for application to concrete, masonry and a more general class of materials.

One of the most attractive aspects of NMM is with respect to its potential for modelling the complete transition from continuum to discontinuum in a discrete manner, but within a continuum setting, with or without remeshing and without the requirement to specify potential failure boundaries *a priori*. This aspect, coupled to a potential to enhance the approximation field globally or locally (with minimal additional computational expense) in order to minimise error, also without remeshing up to any theoretical level suggest that NMM is much more powerful than has been realised previously. The ability to enhance the approximation field without remeshing render the technique particularly attractive for adaptive simulations. Furthermore, there are similarities with other numerical techniques such as FEM and Partition of Unity methods which are worth investigating.

1.2 Outline

The general layout of the thesis can be identified in three main parts:

1. Chapter 2 provides a literature review of strategies and numerical techniques that aim to resolve localisation and failure phenomena.

2. Chapters 3, 4 and 5 investigate and develop the basis of NMM (Chapter 3), the ability to improve the approximation (Chapter 4) without remeshing and using adaptivity, and the ability to describe localised failure (Chapter 5), also without remeshing.

3. Chapters 6 and 7 discuss and develop further aspects such as integration and the behaviour and treatment of higher-order phenomena.

More specifically, the thesis is organised as follows:

Chapter 2 provides an introduction to numerical techniques for modelling localisation and failure. The main advantages and disadvantages of these methods are discussed briefly with emphasis on their application in problems that involve quasi-brittle structures. This sets the scene which ultimately leads to a general introduction of NMM and the reasons for which it is the framework of choice for undertaking this research.

Chapter 3 establishes and develops further the basis of NMM. In contrast to traditional formulations of NMM, here the technique is established using a variational approach. Shape and basis functions are introduced for any arbitrary level of the approximation, for any number of spatial dimensions. The enforcement of essential boundary conditions is discussed for a variety of traditional and non-traditional methods and extended with the application of Lagrange multipliers. Furthermore, a technique which employs projection matrices is implemented. This restores the problem to its original number of unknowns and does not depend on arbitrary stiffness constraints. Similarities with FEM are discussed at the end of the chapter.

Chapter 4 treats the higher-order enhancement of the approximation (in essence a form *p*-enhancement), for any theoretical level, without remeshing. The practical implementation of this approach is restricted only by computing capabilities. The chapter also unveils an algorithm to undertake local enhancement and illustrates the use of error indicators for adaptive enhancement.

Chapter 5 focuses on the discontinuous modelling aspects of the method. The topological and constitutive resolution of discontinuities, such as cracks, is developed and illustrated with numerical examples. A particularly interesting aspect of the approach is that it can also be used to potentially introduce discontinuities in other partition of unity methods. This leads to an examination of similarities and differences between NMM and the recent extension of FEM, the Extended Finite Element Method (XFEM).

Integration in higher-order and discontinuous domains is discussed in Chapter 6, whereas Chapter 7 treats issues of modelling higher-order boundaries. Finally, Chapter 8 provides closure with a summary of the main conclusions and future perspectives of this work.

1.3 Key advances

This research explores and develops NMM as an alternative framework for modelling the transition of continuum to discontinuum in quasi-brittle materials. The following points summarise the key advances achieved during the project:

1. Discontinuities are introduced naturally, in a discrete manner but within a continuum setting, without *a priori* assumptions and without the need for remeshing

2. A local enhancement strategy for the approximation field is achieved, for any arbitrary level of enhancement, without the need for remeshing

3. Essential boundary conditions are introduced in a robust and efficient manner using projection matrices, for any order of the approximation field without the use of artificial penalty constraints

4. Integration of the discretised system of equations is undertaken explicitly (exactly) using simplex integration, for any arbitrary level of local enhancement of the approximation field

Furthermore, NMM is recast in a more rigorous form than before and many aspects of the work originally introduced by Shi in 1996 [96] and other researchers are extended. The formulation is generalised for the first time for domains of three spatial dimensions and the enforcement of essential boundary conditions is improved with the implementation of a technique which employs projection matrices and does not depend on artificial stiffness or increases the number of unknowns. In addition, certain issues with higher-order boundaries are investigated and potential treatments are proposed.

1.3.1 Higher-order approximation and local enhancement

Similar to FEM, the NMM approximation field can be improved by increasing the number of nodes of the mesh (*h*-enrichment) or relocating existing nodes (*r*-refinement). Furthermore, the approximation can also be improved by increasing hierarchically basis polynomials without increasing the number of nodes.

The original NMM has been extended by several researchers [21, 65] such that the level of approximation can be theoretically of any order. Whilst the foundations for higher-order NMM have been laid, there have been few attempts to implement this in practice for any arbitrary level of approximation. Uniquely, this project has demonstrated how enhancement of the approximation field may be carried out using algorithms that perform the process for any arbitrary level.

Moreover, this research developed a novel technique whereby the order of the displacement function can be increased for a selected number of nodes, thereby only enhancing the level of approximation locally. This allows the approximation field to improve for only critical areas of the domain with no remeshing and at minimal additional computational expense. The procedure can be desirable for *p*-adaptivity simulations and this is demonstrated with the development and use of an adaptive algorithm driven by simple error indicators. It is worthwhile to note that the implementation of this adaptivity approach is easy and it is similar for problems of any spatial dimension.

1.3.2 Enforcement of essential boundary conditions

Traditionally, essential boundary conditions in NMM are enforced using penalty methods. Although the implementation of penalty constraints is normally efficient, it may not be entirely satisfactory due to the use of artificial constraints and the potential development of conditioning issues.

The enforcement of essential boundary conditions was extended with the use of a robust and efficient technique which employs projection matrices for any order of the approximation field. Unlike penalty methods the technique eliminates the need for artificial constraints and unlike traditional Lagrange multiplier methods it restores the problem to its original number of unknowns.

Furthermore, the use of an alternative enforcement approach was investigated by modifying directly the form of displacement polynomials employed at the boundary.

1.3.3 Modelling of discontinuities

Traditionally, quasi-brittle failure has been resolved numerically using either continuum-based approaches (for example smeared crack approaches) or via the introduction of discrete displacement discontinuities. The former approach can provide a realistic description of strain localisation and microscopic processes but does not fully resolve the complete mechanism of discrete failure zones observed in quasi-brittle structures and can result in severe numerical difficulties.

The latter approach provides a potentially better description of discontinuous processes but has been traditionally incorporated in numerical techniques via the use of remeshing and *a priori* assumptions with regard to the location and path of potential failure zones.

A unified continuous-discontinuous method is viewed as a more appealing and efficient approach for simulating the entire range of quasi-brittle phenomena. This project has shown how NMM can be utilised to introduce arbitrary displacement discontinuities without the need for remeshing and without the use of failure zones which have been specified *a priori.*

Shape and displacement functions remain in essence unaltered and only the influence domain of weight functions is augmented. As a result, the approach is relatively simple to implement and can be potentially utilised in other techniques with shape or weight functions that form a partition of unity.

In addition it is demonstrated that there are strong parallels between NMM and XFEM with respect to the introduction of displacement discontinuities. This means that the substantial amount of research that has been carried out in the field of XFEM, such as tracking discontinuities with level set methods and resolving crack branching, can be potentially utilised in further developments of NMM.

1.3.4 Exact integration for arbitrary levels of enhancement

In NMM the number of nodes per element and weighting functions remain normally constant but the order of displacement polynomials that constitute the basis functions can be increased. Since the integration domain and the structure of displacement polynomials are simple, the integration associated with the discretisation process may be undertaken explicitly (exactly) for any arbitrary level of enhancement without a loss of generality using simplex integration.

This work developed a strategy for simplex integration that does not require explicit derivation of the element matrices for any arbitrary level of enhancement of the approximation.

1.4 *Symbolic notation*

In principle, bold capitals denote vectors and matrices, whereas normal characters denote scalar quantities. References in literature are displayed in square brackets, while equation references are denoted in parentheses.

Latin symbols

$\mathbf{A}$	constraint matrix
$\mathbf{B}$	strain interpolation matrix
$\mathbf{C}$	constraint matrix
E	Young's modulus
$\mathbf{E}$	elasticity matrix
$\mathbf{L}$	differential operator
$\mathbf{M}$	mass matrix
$\mathbf{N}$	normal vector
N_i	order of displacement function of cover i
$\mathbf{Q}$	projection matrix
$\mathbf{R}$	auxiliary matrix
$\mathbf{T}$	shape matrix
$\mathbf{T}_i$	cover shape matrix
$\mathbf{a}$	deformation vector
$\mathbf{a}_i$	cover deformation vector
$\mathbf{b}$	body force per unit volume
k	penalty number
m_i	number of terms of displacement function of cover i
$\mathbf{n}$	outward normal to $\boldsymbol{\Omega}$
p	spatial dimension
$\hat{\mathbf{t}}$	surface traction with outward normal $\mathbf{n}$
$\boldsymbol{u}$	continuous displacements
$\hat{\boldsymbol{u}}$	prescribed displacements
$\mathbf{u}$	discretised displacements
w_i	weighting (or cover) function
$\mathbf{x}$	spatial coordinate vector

Greek symbols

Γ	boundary of a domain
Γ_u	subset of Γ on which displacements are prescribed
Γ_t	subset of Γ on which tractions are prescribed
Δt	finite solution increment
$\Delta\mathbf{u}$	displacement increment
Π	functional
Ω	volume, surface or line domain, excluding boundaries
Ω^e	element domain
Ω_i	domain of cover i
$\alpha_{i,k}$	cover displacement function for cover i, direction k
$a_i, b_i\ldots$	weighting function coefficients
${}^{i,k}\beta_{m_i}$	mth polynomial coefficient of function $\alpha_{i,k}$
δ	variation
$\boldsymbol{\varepsilon}$	continuous strain
$\boldsymbol{\lambda}$	discretised Lagrange multipliers
ν	Poisson's ratio
$\boldsymbol{\sigma}$	stress vector

Miscellaneous symbols

$\mathcal{L}$	continuous Lagrange multipliers
$\mathbb{R}$	Euclidian space of real numbers

Abbreviations

BEM	boundary element method
EFG	element-free Galerkin
DEM	discrete element method
DDA	discontinuous deformation analysis
FEM	finite element method
NMM	numerical manifold method

XFEM	extended finite element method
PU	partition of unity

2 Numerical resolution of quasi-brittle localisation and failure

2.1 Introduction

The family of quasi-brittle materials consists of a wide range of modern and traditional structural and architectural materials such as concrete, fibre-reinforced concrete, cementitious mortars, masonry, rock, grouted soils, stiff clays and several geomaterials, polymers, laminates, fiber composites, ceramics, ice, consolidated snow, dental cements, biological shells, bone, paper and wood [12, 51, 125].

Furthermore, certain metals and intermetallic compounds are classified as quasi-brittle [53, 84] and certain predominantly brittle or ductile materials can also exhibit quasi-brittle response under suitable environmental conditions and confinement [125].

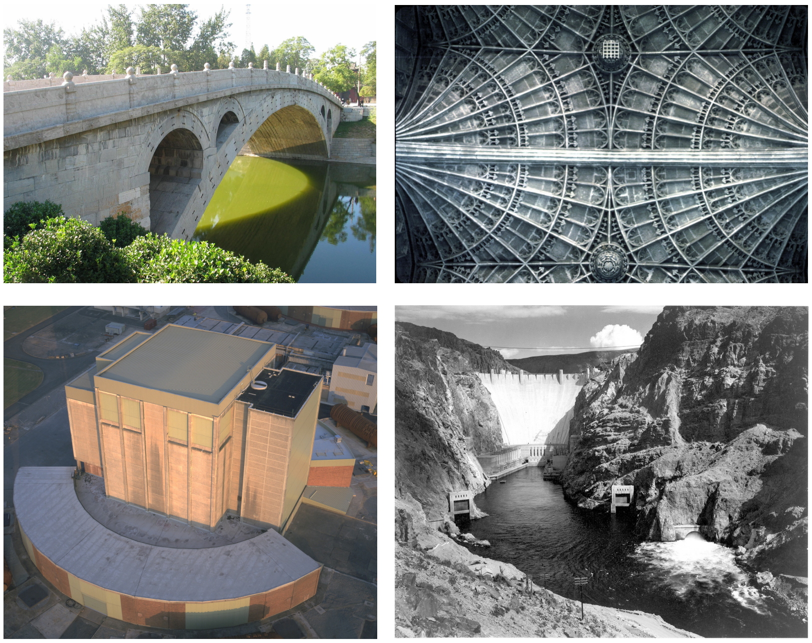


Figure 2-1. Examples of structures constructed of quasi-brittle materials. From top left to right: 1. The 1,400 year old Zhaozhou arch bridge in China [128] 2. Detail of vault of King's College Chapel in Cambridge, U. K [119]. 3. Reactor building of Berkeley nuclear power station, U. K. [20] 4. Hoover Dam, Colorado River, U. S. [78]

Quasi-brittle behaviour is generally desirable because it is related to the capability of material to dissipate energy. In principle, this can be engineered by enhancing material inhomogeneities [51]. While brittle materials fail with very little or no energy release and deformation, quasi-brittle materials undergo some deformation and energy release before they give rise to the appearance of distinct discontinuities (cracks) and eventually fail.

Quasi-brittle materials are of paramount importance in several engineering fields such as structures, geotechnics, tectonics, arctic-ice mechanics and even aeronautics. They are also important in several engineering industries, such as nuclear, mining, petrochemical and civil. Apart from structural function, they also provide important roles in applications such as containment, shielding and aesthetics.

Quasi-brittle structures exhibit a range of complicated nonlinear stress-strain phenomena when their yield capacities are exceeded. Structural members may exhibit large plastic strains or fail more abruptly depending on their ductility, while properties such as stiffness, tensile and compressive strength can increase due to rate effects and confinement or reduce due to time-dependent phenomena such as shrinkage and creep [18, 29, 68, 93, 122].

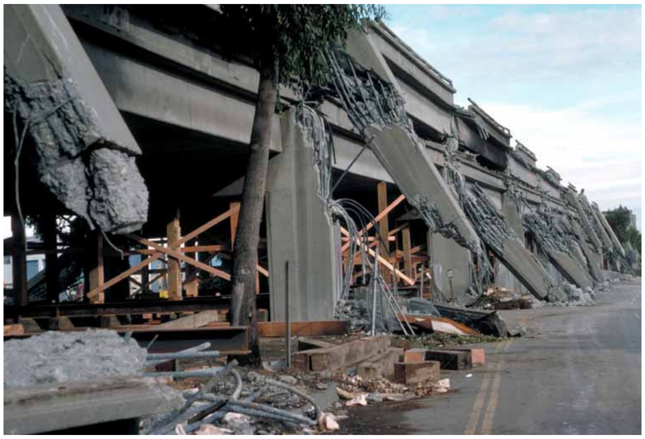


Figure 2-2. Part of the reinforced concrete Cypress Street Viaduct, in California, which collapsed as a result of the Loma Prieta earthquake in 1989 and caused 42 fatalities [118]

Furthermore, the interaction of distinct failure mechanisms (tension, compression, shear, torsion) in multi-axial scenarios and the interaction with reinforcement components and bond materials in composite structures (for example in reinforced concrete and masonry structures), can complicate the understanding of structural response even further.

The inhomogeneous, anisotropic, multi-constituent and often multi-phase character of quasi-brittle materials renders understanding of their behaviour notoriously challenging to resolve mathematically. The following section provides an introduction in fundamental concepts behind the implementation of numerical models that attempt to describe quasi-brittle response.

2.2 *Numerical resolution of localisation and failure*

When quasi-brittle materials are subjected to mechanical, thermal or chemical actions, their mechanical capacities can be exceeded. Near the peak strength limit and prior to failure, inelastic strains tend to localise in relatively thin bands, or process zones [34, 125]. At some stage between localisation and failure, distinct discontinuities appear (Figure 2-3), which give rise to flexural, shear and torsional cracks, or damage and crushing of material due to high inelastic compressive strains.

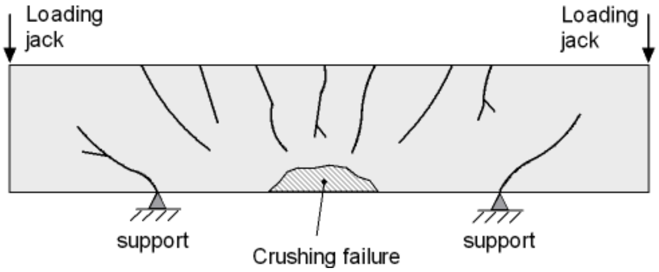


Figure 2-3. Typical crack pattern in a concrete beam

Once strength capacity is reached and inelastic strains begin to localise, quasi-brittle materials display some ductility by undergoing deformation and energy release while stress is reduced gradually to zero. This gradual reduction of strength is known as ‘softening’ (Figure 2-4, Figure 2-5).

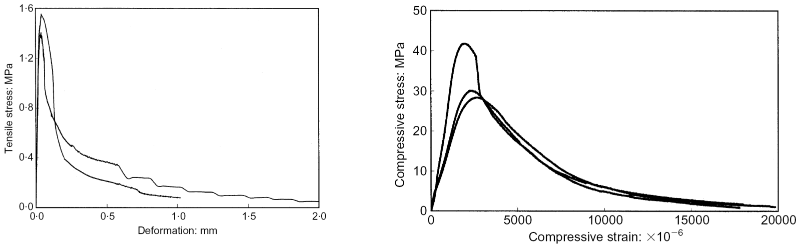


Figure 2-4. Experimental tensile stress-deformation (left) and compressive (right) stress-strain curves of concrete specimens in uniaxial tension and compression [59]

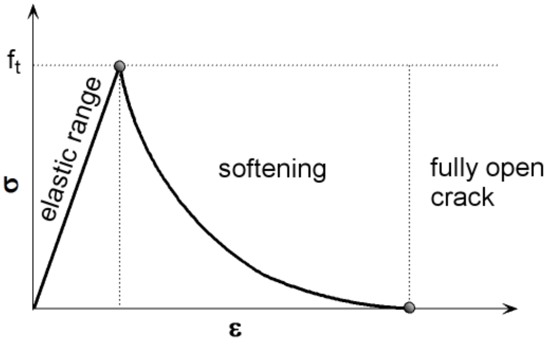


Figure 2-5. Typical stress-strain relationship of quasi-brittle materials in uniaxial tension

Numerical models that aim to resolve localisation and failure phenomena such as those discussed above fall principally into three distinct classes [49, 99, 125]:

1. Continuous
2. Regularised-continuous
3. Discontinuous

The first class resolves displacement jumps that occur from the appearance of discontinuities using smooth field approximations. In physical terms, this implies that displacement jumps are distributed (or 'smeared' as it is commonly termed) over a discretised region of the continuum. Since the displacement field is differentiable, strain can be defined uniquely and therefore the smeared representation can be replaced by an equivalent inelastic strain. This type of discontinuity is commonly referred to as a 'weak' discontinuity, since jumps in the displacement field are not considered explicitly.

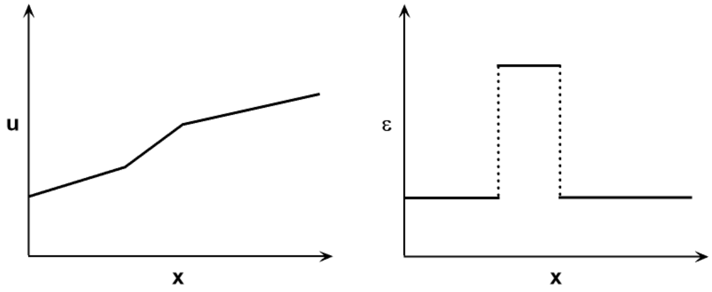


Figure 2-6. Representation of a weak discontinuity

The second class is an enhancement of the first and consists of regularised models that represent displacement jumps using smooth approximations in both stress and strain terms. This is achieved using enhancements of the classical continuum theory by means of gradients of internal variables [33] or non-local terms [83].

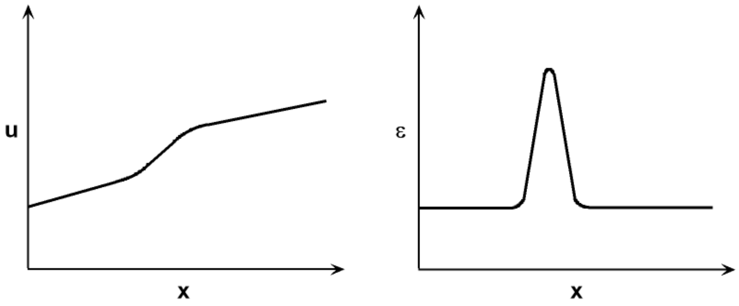


Figure 2-7. Representation of a weak discontinuity in regularised media

The third class allows for jumps in the displacement field in an explicit manner. In finite elements this is undertaken traditionally using *a priori* assumptions regarding the location of potential discontinuous boundaries. In essence, discontinuities are represented by additional degrees of freedom and special interface models that allow for cohesive or traction-free boundaries. This type of discontinuities is commonly referred to as 'strong' discontinuities, since jumps in the displacement field are modelled explicitly.

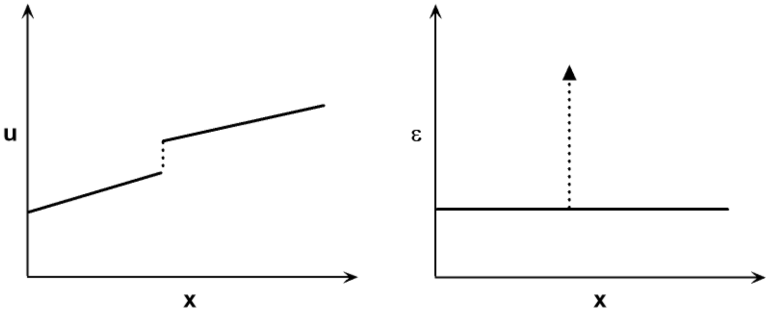


Figure 2-8. Representation of a strong discontinuity

Numerical approximations based on the first class are attractive from the point of view that the problem can be solved within a continuum setting. However, as strain localisation occurs, the governing equations become ill-posed, which causes numerical difficulties and requires regularisation of the continuum model in order to overcome this.

Regularised-continuous techniques can restore well-posedness of the governing equations and potentially enable the inclusion of size effects using concepts that take into account the material microstructure. This is achieved using additional variables which are typically related to some characteristic length associated with the material or localisation pattern. However, these additional variables can be difficult to determine.

Discontinuous techniques treat fracture and failure in a more straightforward manner, in terms of displacement jumps and tractions rather than in terms of stresses and strains. However, this traditionally necessitates intensive remeshing to represent evolving boundaries, or *a priori* assumptions regarding the location of discontinuities, and can consequently lead to increased computational cost and mesh bias pathologies.

An overview of numerical techniques commonly employed for the discontinuous representations discussed above is given in sections 2.4 to 2.9. First, an introduction of the main strategies with which these are employed is provided in the following section.

2.3 Numerical strategies for modelling quasi-brittle failure

Current strategies for the utilisation of numerical models and techniques that aim to represent structural behaviour of quasi-brittle structures fall in the following two main classes:

1. Macro-models (or continuous/homogeneous models), in which structural behaviour is obtained from the homogenised description of material as isotropic or anisotropic composite continuum.
2. Micro-models (or discrete models), in which material constituents or components and their interaction are considered individually.

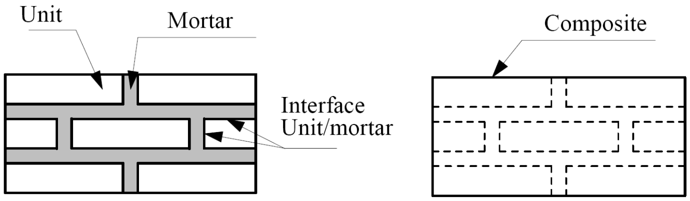


Figure 2-9. Examples of micro- (left) and macro-models (right) in masonry [63]

In certain situations an intermediate meso-scopic scale is identified (Figure 2-10) to differentiate models in which the smallest scale consists of material constituents or components (concrete aggregates, masonry mortar) from models of even smaller scales (e.g. molecular).

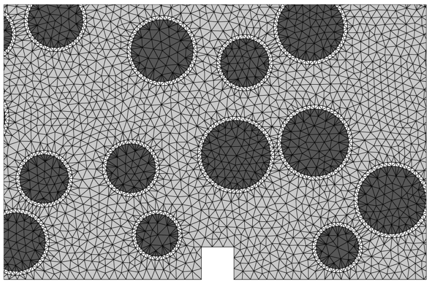


Figure 2-10. Meso-scopic finite element model of concrete matrix and aggregates [82]

Macro-models involve simplified hypotheses in order to describe the normally complex material microstructure as that of an isotropic or anisotropic composite. Consequently, they tend to be inaccurate for resolving micro-mechanical failure processes but they are associated with significant reduction of computational resources when compared to more sophisticated micro-models, and reduced pre- and post-processing effort.

Advanced homogenization techniques [2, 52, 63] aim to enhance the representation of micro-structural behaviour in such models by resolving micro-constitutive laws into an anisotropic macroscopic level, in a way that the macro-constitutive law is not actually implemented, or even known. However, such techniques are not yet widely available in practice.

Micro-models advocate the capability for more realistic resolution of structural behaviour via the explicit representation of individual material components and their interaction. In practice, due to their complexity they can be more difficult to implement and validate than more simplified macro-models and they tend to be computationally involved. As a result, their application is often limited to the analysis of small structural specimens and details.

Traditionally, in the context of resolution of discontinuous processes, macro-models are frequently associated with continuous approximation techniques and weak discontinuity models, whereas micro-models are associated with discrete techniques and strong discontinuities, although this is not a definite rule. A new generation of techniques that consists mainly of meshless and partition of unity approximations (sections 2.7 and 2.8) advocates the potential to provide unified descriptions of the transition from continuum to discontinuum in a realistic and computationally efficient manner.

2.4 *Overview of numerical techniques to resolve discontinuous processes*

The following sections aim to provide an overview of numerical techniques that are used to resolve discontinuous processes. It is not the intention for this to be a comprehensive review, but a brief discussion of the main advantages and disadvantages of most popular techniques.

This overview will ultimately set the scene for the introduction of the Numerical Manifold Method in section 2.9 and provide the basis for drawing parallels between this and other techniques such as the Finite Element method, the Extended Finite Element method and Discontinuous Deformation Analysis.

2.5 *Continuous methods*

On a practical level, the resolution of localization and failure has been traditionally approached using continuum-based techniques such as the Finite Element Method (FEM) [27], the Finite Difference Method (FDM) [42] and Boundary Element Method (BEM) [8] normally coupled with weak discontinuity models based on plasticity, damage or smeared crack concepts.

Continuous (or continuum) techniques are based on domain or boundary discretisations that resolve continua in a finite number of interconnected sub-domains. Sub-domains (or elements) are associated with sets of functions which define the approximation field within their domain. Typically, the resulting equations are assembled into a system on which constraints and loads are applied in order to obtain a solution.

FEM is without doubt the most known and broadly applied technique used to solve problems in solids. Its origins can be traced back to the pioneering work of Argyris [4], Zienkiewicz [132] and the work of Varga on variational finite differences [120].

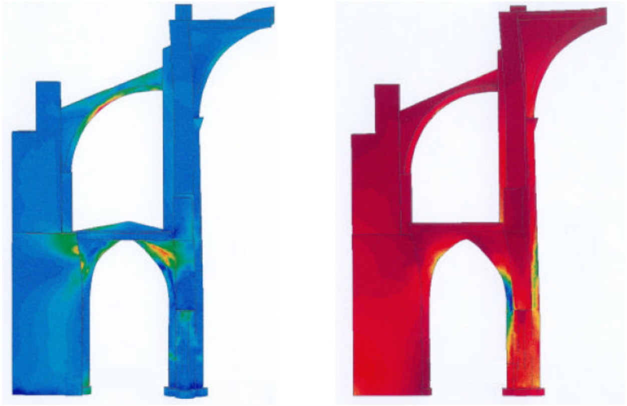


Figure 2-11. FEM-damage model of a gothic cathedral [85]

Continuum methods are often applied in macro-models of practical industrial applications as they are capable to derive adequate information about the global structural behaviour with relatively minimal initial model preparation and computational cost. However, modelling of discontinuous phenomena traditionally requires either the use of weak discontinuity models or some form of remeshing so that discontinuities can be resolved in a strong sense.

The latter situation can be difficult to implement, it can be computationally involved and it can potentially lead to mesh bias issues. On the other hand, weak discontinuity models resolve jumps in the displacement field in a smooth sense (section 2.2) and can be practical when no remeshing is undertaken.

An alternative approach to remeshing or the use of weak discontinuity models is the use of interface zones with interface elements. Interface elements can be used in strategic *a priori* specified locations or they can be dispersed throughout the finite element mesh [35, 112] to model existing or potential jumps in the displacement field in a strong manner (Figure 2-13). However, the former depends on knowledge or conjecture about where discontinuities may appear, while the latter can lead to considerably increased computational expense. Interface behaviour is defined by nonlinear interface laws which can be traction-free or cohesive while the surrounding continuum material may remain elastic or it may be nonlinear.

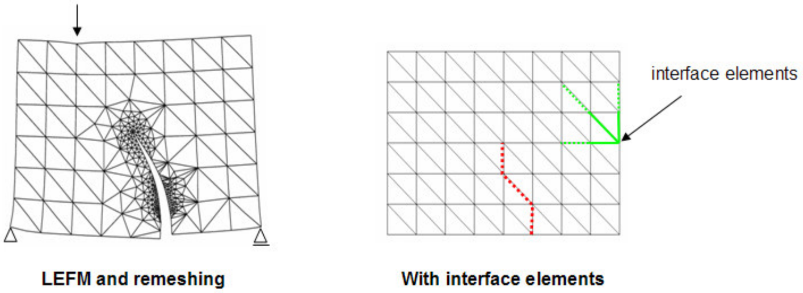


Figure 2-12. Modelling of a strong discontinuity in a concrete beam using Linear Elastic Fracture Mechanics and remeshing (left) and FEM with interface elements

One of the main disadvantages of this approach is that potential failure mechanisms are normally defined *a priori*. This can render the model preparation process particularly time consuming. Furthermore, models may suffer from path dependence and computational cost can be high as relatively fine discretisation or remeshing is often required in order to capture the evolution of discontinuities in an accurate and robust manner.

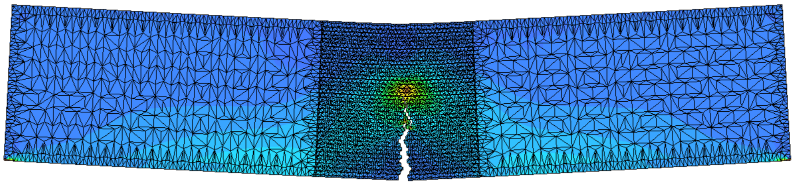


Figure 2-13. Three-point bending test of an unreinforced concrete beam using a combined FEM-cohesive interface model

Here, it is worthwhile to also note the lattice framework, which dates back to the work of Hrennikoff [44]. Lattice techniques adopt a strong discontinuity approach and as a result they are sometimes perceived as discrete techniques although do not normally entail automatic detection of new contacts (see section 2.6).

Lattice techniques replace the continuum with an equivalent beam or truss structure (which is known as the lattice). Elements can brake into lattices based on criteria of strain, force, or energy, as determined from the displacement solution [116]. Typically, an element is removed from the solution if it meets the adopted criterion.

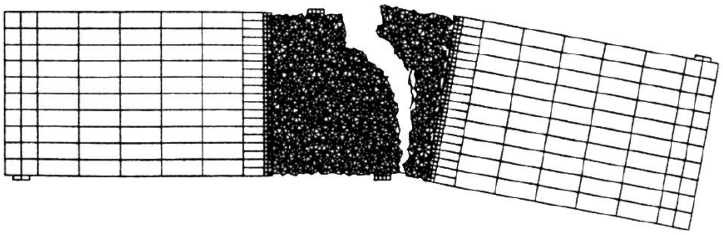


Figure 2-14. Lattice crack pattern of SEN concrete specimen test [92]

2.6 *Discrete methods*

Similar to certain continuum-based techniques, discrete methods allow strong discontinuities to be represented *a priori* in the numerical model. However, they are inherently more thorough with regard to the way they enforce and detect contact constraints and they generally appear to be better suited for simulating complete failure of solids and interaction of fragmented parts.

In discrete methods, a domain with pre-existing or no discontinuities is constructed as the assemblage of a finite number of discrete deformable or rigid bodies (also called blocks), interconnected with contact constraints and interfacial constitutive relationships. In contrast to continuum-based techniques, discrete methods do not normally augment the original mesh although the configuration of blocks is allowed to change, since discontinuities are aligned with the topology of discrete bodies.

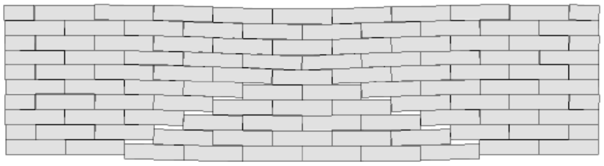


Figure 2-15. Deformed DDA model of dry brick assembly problem [17]

According to Zienkiewicz [134], the discrete framework is one which:

1. Allows finite displacements and rotations of discrete bodies, including complete detachment
2. Recognizes new contacts automatically, as calculation progresses.

The Discrete or Distinct Element Method (DEM) [30] and Discontinuous Deformation Analysis (DDA) [94] are merely examples of a large array of discrete techniques. Key differences between different techniques are usually identified with regard to the way contact is enforced or detected, the time-integration scheme (implicit or explicit), the type of deformability and the type of interfacial constitutive laws.

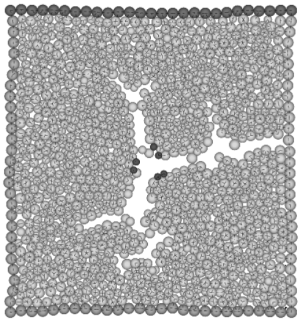


Figure 2-16. Fracturing of wellbore stability model using DEM [32]

Discrete frameworks are particularly attractive in low and high-speed impact applications where fragmentation and debris scatter can be important, modelling of structures that consist of interconnected deformable blocks, such as masonry [102], although they have also been applied in simulations of fracturing of concrete and rock materials [32] in a variety of scenarios.

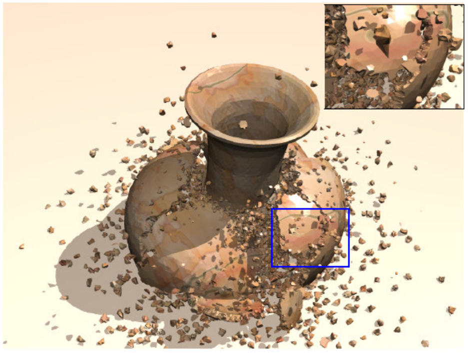


Figure 2-17. DEM simulation of a vase shattered onto the ground [131]

Potential difficulties or limitations of discrete techniques can be associated with computational cost issues [86] due to large numbers of bodies or blocks often required to discretise continua, the sensitivity of explicit schemes to the solution of static or quasi-static scenarios using dynamic relaxation, the simulation of deformable continua and the detection and enforcement of contact constraints.

2.7 *Meshless methods*

The desire to alleviate difficulties associated with mesh generation and re-meshing associated with traditional continuum-based techniques has recently given rise to an increased interest in the development and application of meshless methods.

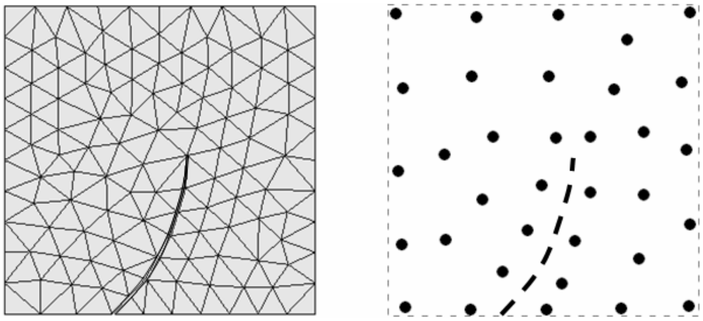


Figure 2-18. Modelling of a discontinuity using a mesh-based (left) and meshless approach (right)

Meshless techniques generally conceive the domain as an assemblage of overlapping domains of influence $\Omega_i \subset \Omega$, also known as covers, or patches. The domains of influence are effective within the physical domain and ineffective outside the domain or on emerging discontinuities (Figure 2-19).

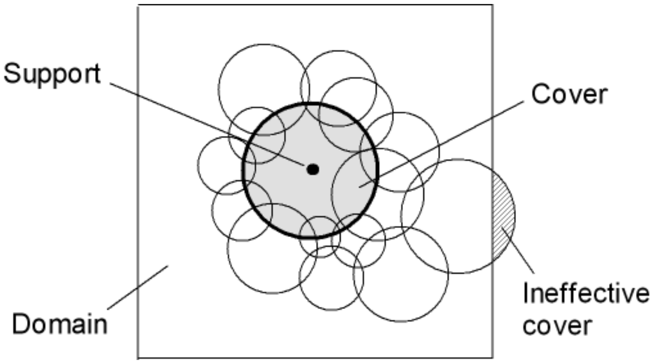


Figure 2-19. Meshless cover domain of influence

Each cover Ω_i is associated with a weighting function $\phi_i(\mathbf{x})$. If $\phi_i(\mathbf{x})$ is nonzero only in Ω_i, and is such that:

$$\sum_i \phi_i(\mathbf{x}) = 1 \qquad (2.1)$$

then the weighting function is a partition of unity [6]. This property has been the impetus of an important new class of mesh-based and meshless numerical techniques discussed in section 2.8.

In contrast to traditional continuum-orientated techniques (such as FEM), the basis of meshless (or meshfree) methods lies in that for any given set of points, or nodes, there is no requirement to define elements in order to determine where the weighting function is zero or non-zero. Consequently, the approach is particularly attractive in simulations of moving boundaries, large deformation and strongly discontinuous phenomena, since no remeshing is required.

In all, there are several meshless methods and families of meshless methods, based on moving least squares [57], smooth particle hydrodynamics [66], natural neighbour Galerkin approximations [107] or reproducing kernel methods [61] to name a few.

A comprehensive overview and developments is given by Belytschko, Duarte, as well as others [14, 36, 58]. The Element-Free Galerkin (EFG) [13], a refinement of the Diffuse Element Method originally proposed by Nayroles [79] based on the moving least squares approximation, can be given as an example of a relatively recent and popular approach which is also based on the partition of unity.

It is worthwhile to note that, as in the case of EFG, shape functions and their derivatives can be continuous. This is in contrast to FEM in which computed strains and stresses are non-smooth and post-processing of such quantities normally requires additional considerations.

Although meshless methods have distinct advantages over traditional continuum and discrete-based techniques with regard to pre-processing and resolution of moving boundaries, from the point of view of accuracy and efficiency there are limitations reported in literature, which in some situations they can negate certain inherent advantages over other techniques [37].

Reported difficulties are primarily associated with the enforcement of essential boundary conditions [14] due to the non-interpolating character of shape functions over nodal parameters, errors due to numerical integration [107] and issues related to computational expense [5, 14] due to the complicated character of shape functions.

2.8 Partition of unity methods

The realisation that it is possible to exploit the partition of unity (PU) property of shape functions of certain meshless methods [13, 80] and FEM [133], leads to the identification and development of the so called PU framework [6].

PU methods advocate distinct advantages over traditional continuum-based methods with regard to modelling of discontinuum states, since they can explicitly resolve jumps in the approximation field without the requirement to undertake remeshing even when no *a priori* assumptions are made with respect to the discontinuity path. Consequently, PU methods appear particularly attractive in applications that involve simulation of moving boundaries, such as fracturing and crack propagation.

The Numerical Manifold Method (NMM) [96], the Extended Finite Element Method (XFEM) [6, 15], Partition of Unity Finite Elements [98, 125] and Polygonal Finite Elements [109] are only a few examples of variants based on the PU concept. FEM may also be seen as a PU method although it does not traditionally exploit these properties with regard to discontinuity modelling. Instead it employs remeshing techniques, predefined interfaces or smeared continuum methodologies (see section 2.5).

NMM appears as a particularly interesting technique for modelling a range of phenomena, as it combines PU characteristics with the possibility to locally improve the approximation also without remeshing, theoretically for any arbitrary level. Although this is also possible with other PU variants, in NMM integration can be undertaken explicitly, for any level of the approximation.

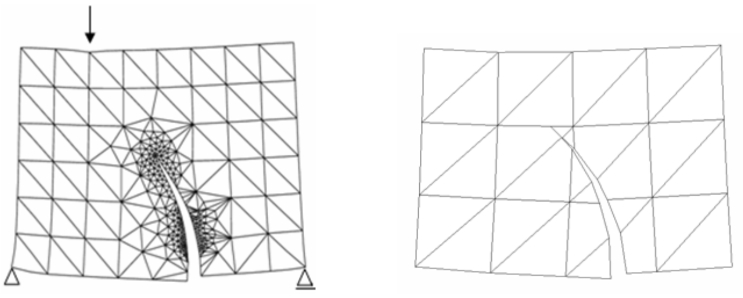


Figure 2-20. Fracturing of a simply supported concrete beam by FEM and linear elastic fracture mechanics with remeshing (left), NMM with no remeshing (right)

Similar to meshless methods, PU methods identify the trial function as the product of weighting functions (or shape functions), which are centred on supports, and functions which describe the approximation field within a discretised region of the domain.

The weighting function associated with a given support is equal to unity on that support and decreases to zero on neighbouring supports and boundaries. At any point within the physical domain, the sum of weighting functions equals unity.

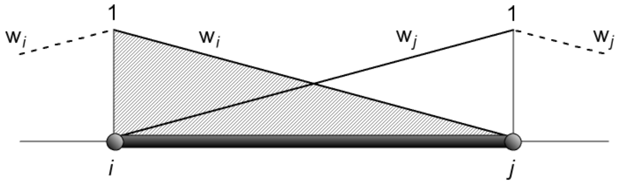


Figure 2-21. FEM/NMM/XFEM partition of unity shape/weight functions on a one-dimensional line element

Jumps in the approximation field are introduced naturally by rendering covers ineffective over the discontinuity domain. This involves the introduction of additional supports, which however overlap existing supports so that the mesh topology remains similar.

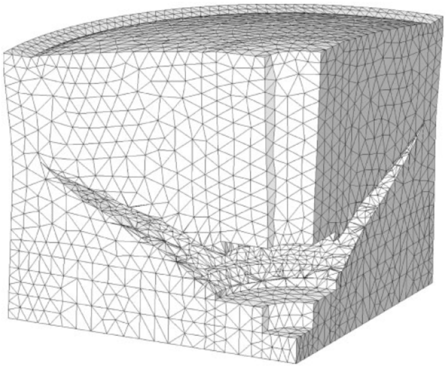


Figure 2-22. XFEM pull-out test of a steel anchor embedded in a concrete cylindrical block [3]

2.9 The Numerical Manifold Method

The Numerical Manifold Method (NMM) [96, 97] is a relatively recent yet potentially powerful numerical analysis technique based on the partition of unity concept [6] and ideas similar to those used in meshless methods [5]. Furthermore, NMM integrates aspects of traditional and hierarchical finite element methods [133, Chapters 3 and 4]. Due to the partition of unity characteristics, it provides the possibility to model both continuum and discontinuum states, as well as the transition from continuum to discontinuum in a single unified framework.

NMM can be viewed as a more generalised formulation of DDA, whereby blocks are substituted by assemblages of elements formed by overlapping covers, or supports, or domains of influence. Similar to DDA, the approximation can be enhanced using higher-order polynomial basis functions to achieve a variable-strain field within elements without altering the mesh (Figure 2-24).

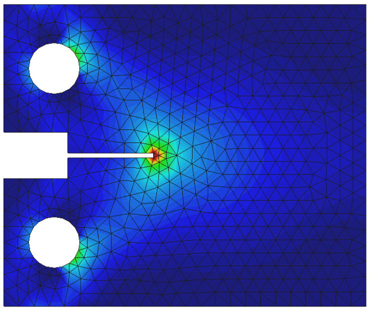


Figure 2-23. Uncracked configuration, mesh and principal stress contours of a notched Concrete Compact Tension Specimen (CTS)

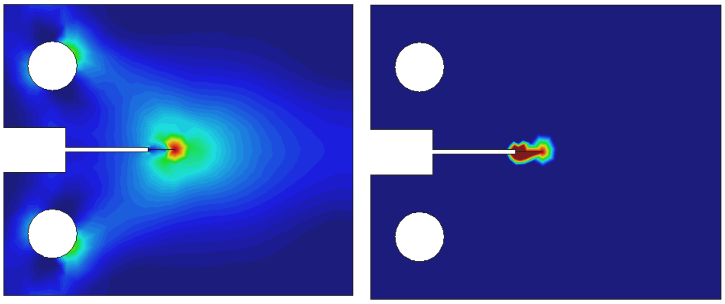


Figure 2-24. Major principal stress contours (left) of CTS crack propagation without remeshing. The approximation is enriched around discontinuities (right) also without remeshing

The principal difference with regard to the way the approximation is constructed in each case is that DDA blocks utilise single covers, while NMM elements utilise multiple covers. Furthermore, NMM does not normally include rotational terms.

Similar to meshless methods, NMM treats discontinuities by introducing the concept of effective cover regions. Where a cover intersects a disconnected domain, that part of the cover becomes ineffective. However, covers and the associated shape functions are defined differently in each case, in both topological and mathematical terms. In NMM the cover domain of influence is typically defined by the element topology, i.e. a mesh is required, whereas in meshless methods definition of elements is not required to determine where shape functions are zero or non-zero.

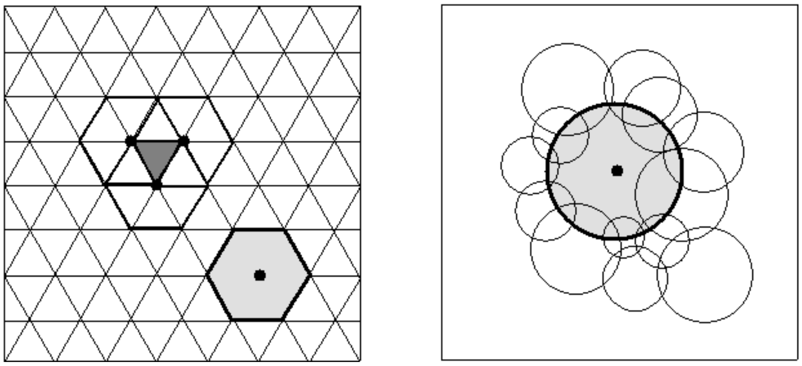


Figure 2-25. Typical cover domain of influence of NMM (left) and a meshless approach (right)

On the other hand, because of this more flexible, with regard to pre-processing, definition of the meshless space, the associated weighting functions [1, 13] can be significantly more complicated that the linear weighting functions typically employed in NMM. Consequently, shape functions and their spatial derivatives as well as integration can be more involved in meshless methods than in NMM.

In addition to similarities with meshless methods, DDA and FEM with standard and hierarchical shape functions, NMM exhibits strong parallels with the more recently developed Extended Finite Element Method (XFEM) [15, 73, 104, Chapter 5].

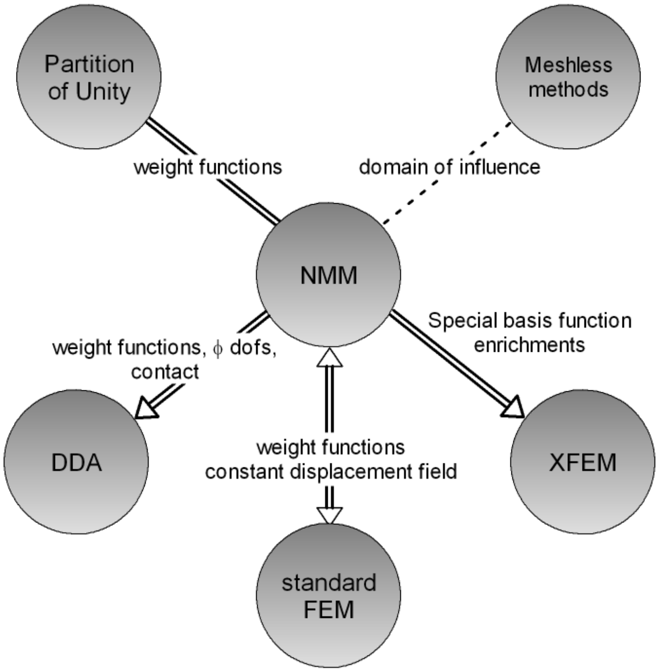

Partition of Unity
Meshless methods
weight functions
domain of influence
NMM
weight functions, ϕ dofs, contact
Special basis function enrichments
DDA
weight functions constant displacement field
XFEM
standard FEM

Figure 2-26. Schematic relationship between NMM and other techniques

The application of NMM has so far been limited predominantly within geotechnical communities in Japan, China and the U. S. [64, 81, 91, 108, 123, 129, 130]. However, the similarities of NMM with meshless methods and hierarchical FEM indicate a possibility of use for modelling failure processes in a wider range of materials and applications.

To date, the original NMM [96] has seen relatively little development although it has been extended in attempts to exploit its potential to improve the level of approximation with higher-order basis functions [23, 65], while preserving the ability to undertake integration analytically. Most developments have not been demonstrated, and there is no evidence of potential issues associated with convergence or enforcement of constraints in higher-order deformability states.

Three important complementary characteristics can best describe the attractiveness of NMM:

1. First, the level of approximation can be improved globally, or locally, without the necessity for remeshing. This characteristic renders the method ideal for *p*-adaptivity.

2. Second, discontinuities in the displacement field can be introduced explicitly without the need for remeshing due to the PU property.

3. Third, integration associated with the discretisation procedure may be undertaken analytically rather than numerically, for any arbitrary level of the approximation.

These characteristics will be introduced in the following chapter and developed further in subsequent chapters.

3 The Numerical Manifold Method

3.1 *Introduction*

In mathematics and physics, a manifold is defined as an abstract topological space, in which locally and around every point there is a neighbourhood topologically similar to the Euclidian structure (although the global structure appears more complicated). The Euclidian space itself is a manifold and in essence, all engineering structures can be perceived as manifolds.

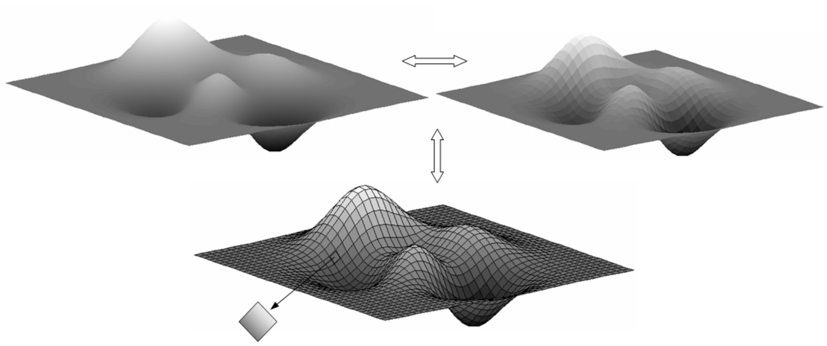


Figure 3-1. The concept of manifolds

The concept of manifolds is important as it facilitates the understanding of complicated structures in terms of simpler ones, which are elements of the former. At this point it may be evident that fundamental concepts associated with most modern numerical techniques – the concepts of discretisation and approximation - are closely associated with the concepts of manifolds.

FEM and other analysis techniques essentially employ the concept of manifolds by the realisation that a domain can be discretised into an assemblage of simpler finite regions, for which the behaviour is well understood. However, a key difference advocated by NMM, which was first introduced by Shi in 1996 [96], comparing to traditional numerical techniques is that its structure allows it to treat problems that involve discontinuities and moving boundaries in a more natural way, without the requirement of computationally involved remeshing techniques.

NMM possesses several similarities with FEM, DDA, meshless and partition of unity methods, integrated in a unique setting. A brief introduction of NMM has been provided in Chapters 1 and 2. This chapter lays out and extends further the basis of the method, which in turn reveals why it is advocated as a potentially powerful candidate for modelling continua, discontinua, and the transition from continuum to discontinuum.

Traditionally, the main equilibrium equations in NMM have stemmed from the principle of minimization of potential energy [96, 97] for two-dimensional problems and for limited forms of displacement functions, whereas essential boundary conditions have been imposed using penalty constraints. Here, NMM is cast in a more general constrained variational form for problems of any spatial dimension, and for any level of approximation, whereby essential boundary conditions are satisfied using a Lagrange multiplier technique although the can be satisfied in several different other ways. The proposed technique employs projection matrices, it does not involve fictitious penalty constraints and restores the problem to its original number of unknowns.

Furthermore, while the key abilities to model the transition from continuum to discontinuum and to improve the approximation without remeshing are unravelled in detail in subsequent chapters, the potentially powerful ideas of NMM can become progressively evident from examination of its fundamental structure.

3.2 *Strong form of the governing equations*

Consider a domain $\Omega \in \mathbb{R}^p$ bounded by Γ. The boundary is composed of sets Γ_u and Γ_t, on which prescribed displacements and tractions are imposed respectively, such that $\Gamma = \Gamma_u \cup \Gamma_t$ and $\Gamma_u \cap \Gamma_t = 0$, in order to obtain a unique solution.

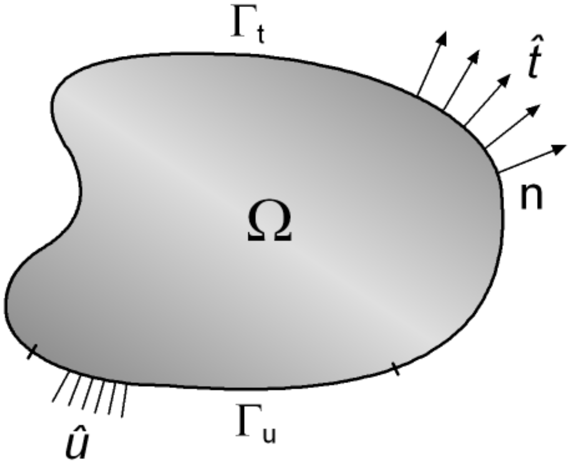


Figure 3-2. Example domain $\Omega \in \mathbb{R}^2$ with boundary sets Γ_u and Γ_t

In the absence of inertial forces, the governing equation for static equilibrium is given as:

$$\text{div}\,\boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega \tag{3.1}$$

or

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \text{ in } \Omega \tag{3.2}$$

or

$$\sigma_{ij,j} + b_i = 0 \text{ in } \mathbf{\Omega} \tag{3.3}$$

which is known as the equation of linear momentum balance and $\sigma_{ij,j}$ and b_i are the stress and body force components respectively. This can be written in full as:

$$\begin{aligned}
&\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + b_x = 0 \\
&\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + b_y = 0 \\
&\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + b_z = 0
\end{aligned} \tag{3.4}$$

Alternatively, using Voigt's notation (3.4) can be written as:

$$\mathbf{L}^T \boldsymbol{\sigma} + \mathbf{b} = 0 \tag{3.5}$$

where **L** is a differential operator:

$$\mathbf{L}^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial z} \end{bmatrix} \tag{3.6}$$

and $\boldsymbol{\sigma}$ is a vector containing the stress components:

$$\boldsymbol{\sigma}^T = \left[\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx} \right] \tag{3.7}$$

Restricting the problem to small displacements, strains are related to displacements via the following kinematic relationship:

$$\boldsymbol{\varepsilon} = \mathbf{L}\mathbf{u} \tag{3.8}$$

where the strain vector is defined as:

$$\boldsymbol{\varepsilon}^T = \left[\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, 2\varepsilon_{xy}, 2\varepsilon_{yz}, 2\varepsilon_{zx}\right] \tag{3.9}$$

On the boundary of the domain, the displacements and tractions are prescribed as follows:

$$\mathbf{g}(\mathbf{u}) = \mathbf{u} = \hat{\mathbf{u}} \text{ on } \Gamma_u \tag{3.10}$$

$$\mathbf{t} = \hat{\mathbf{t}} = \mathbf{N}^T \boldsymbol{\sigma} \text{ on } \Gamma_t \tag{3.11}$$

Equations (3.10) and (3.11) are the essential and natural boundary conditions respectively [133]. Whereas natural boundary conditions are satisfied automatically by the variational form (given later in this chapter), hence the designation 'natural', it is required to modify the variational form in order to satisfy essential boundary conditions.

The matrix **N** contains the components of the outward normal vector:

$$\mathbf{n} = [n_x, n_y, n_z] \text{ to } \Gamma \tag{3.12}$$

so that:

$$\mathbf{N}^T = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_z \\ 0 & n_y & 0 & n_x & n_z & 0 \\ 0 & 0 & n_z & 0 & n_y & n_x \end{bmatrix} \tag{3.13}$$

3.3 *Shape functions*

In NMM the physical domain is covered by a grid of nodes and each node has a domain of influence or support. In NMM literature, this domain of influence is known as the cover. Similar to meshless methods, a cover is associated with a weighting function which specifies the interpolation within the domain. This domain of influence is defined by element topology. The common area of an arbitrary number of overlapping covers constitutes an element (Figure 3-3) and the entire physical domain is completely covered by elements.

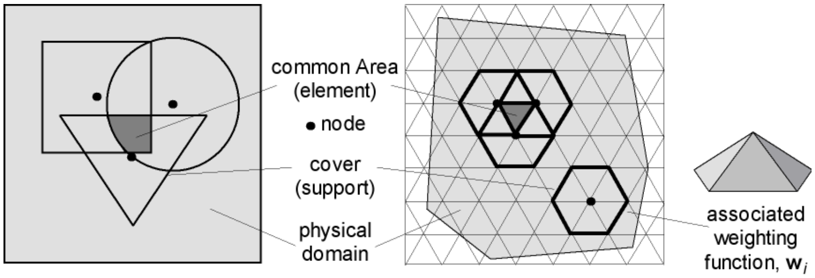


Figure 3-3. Covers and cover functions in NMM. General cover functions (left), simple hexagonal cover functions defined by a regular background mesh (right).

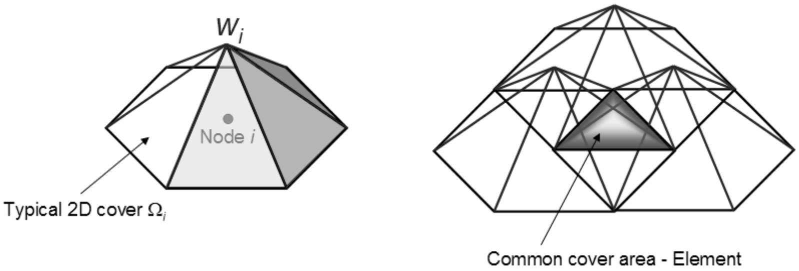


Figure 3-4. Schematic representation of typical hexagonal covers in two dimensions

In principle, the extent of a cover can be arbitrary. However, the nature of the integration of the weak form is closely related to the weighting function and the extent of the common area. Since the shape and extent of the cover is crucial to integration, a simple choice for the shape of cover is usually adopted.

In order for the integration associated with the discretisation process to be undertaken explicitly the extent of each cover associated with a particular node has to be defined by the node's neighbours. Figure 3-3 shows a regular grid of nodes covering a two-dimensional physical domain, where each nodal cover is hexagonal and the common area of overlapping covers is a triangular element. In the case that an unstructured grid is adopted, each cover is still defined by its neighbours and therefore will, in general, be an irregular polygon. However, the elements associated with overlapping covers are still triangular, albeit not necessarily regular in shape.

For simplex elements, or elements that adopt the form of the simplest possible shape in any given space, the cover weighting function for node i is typically defined as a linear function of position in the global coordinate system:

$$w_i = a_i + \sum_{s=1}^{p} \boldsymbol{\kappa}_{s,i} \cdot x_s \tag{3.14}$$

where a_i and $\kappa_{s,i}$ are the cover weighting function coefficients and x_s are the components of the position vector in the p-dimensional Euclidian space.

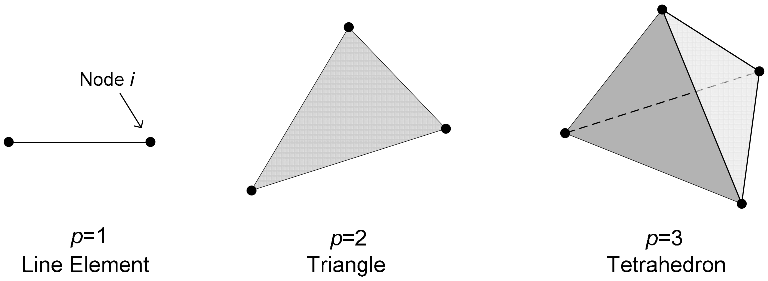


Figure 3-5. Simplex elements in $\mathbb{R}^p$, for $p = 1,2,3$

For example, in the case of four-node tetrahedral elements ($p = 3$, $i = 1,2,3,4$), the weighting function w_i is:

$$w_i = a_i + \sum_{s=1}^{3} \kappa_{s,i} \cdot x_s = a_i + \kappa_{1,i} x_1 + \kappa_{2,i} x_2 + \kappa_{3,i} x_3 \tag{3.15}$$

or, alternatively:

$$w_i = \left(a_i + b_i x_1 + c_i x_2 + d_i x_3\right) \tag{3.16}$$

where a_i, $b_i = \kappa_{1,i}$, $c_i = \kappa_{2,i}$, $d_i = \kappa_{3,i}$ are the weighting function coefficients and $x_{1,2,3}$ are components of the position vector.

A particularly significant property of the weighting function is that it is a partition of unity. Hence, the weighting function has to satisfy the following conditions:

$$\sum_{i=1}^{n} w_i(\mathbf{x}) = 1 \quad \forall\, \mathbf{x} \in \Omega^e \tag{3.17}$$

$$0 \le w_i(\mathbf{x}) \le 1 \quad \forall\, \mathbf{x} \in \Omega_i \tag{3.18}$$

$$w_i(\mathbf{x}) = 0 \qquad \forall\, \mathbf{x} \notin \Omega_i \tag{3.19}$$

where Ω^e is the element domain and $\Omega_i \subset \mathbb{R}^3$ represents a cover, which may not be necessarily a subset of Ω; for example a cover may extend outside the physical domain, as in the case of evolving discontinuities discussed in Chapter 5. Equations (3.17), (3.18) and (3.19) ensure that the weighting function that corresponds to any part of a cover Ω_i that lies outside the physical domain Ω is set to zero. Equation (3.18) is in fact the Kronecker delta function with which FEM shape functions are also associated:

$$\delta_{ij} = \begin{cases} 0 & \mathit{for}\, i \neq j \\ 1 & \mathit{for}\, i = j \end{cases} \tag{3.20}$$

In other words, cover functions are ineffective (zero) outside physical boundaries and the cover grid can be independent of the physical domain. This concept implies that it is possible to model geometric discontinuities without the aid of complicated and computationally involved remeshing techniques, by simply forcing cover functions to be zero.

In practice, this concept implies that rather than creating a mesh that coincides with the boundaries of the problem, and therefore having to resort to unstructured meshing, the analyst can employ the benefits of a regular structured mesh and simply identify boundaries and enforce appropriate constraints (Figure 3-6). In order to do this, it is necessary to employ some form of discontinuity tracking, which is arguably less involved than the implementation of remeshing algorithms.

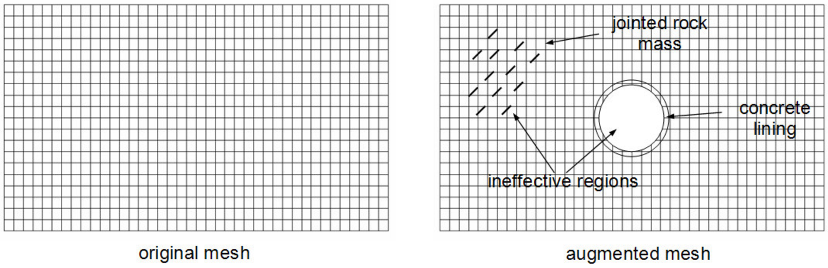


Figure 3-6. NMM mesh of a circular concrete tunnel in jointed rock. Left: original mesh; right: illustration of mesh with effective and ineffective boundaries. No remeshing has taken place.

Whereas cover topology is not in principle required to coincide with the topology of the physical problem, it can often be advantageous to ensure that the two overlap. In this case, it is possible to easily employ the widely available pre- and post-processing software originally intended for finite element codes.

3.4 *Displacement functions*

In three dimensions, the unknown displacement function within an element (defined by n number of covers and nodes) is defined as:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \sum_{i=1}^{n} w_i \, \boldsymbol{\alpha}_i \tag{3.21}$$

where w_i is the weighting function (or shape function) associated with the cover of node i and $\boldsymbol{\alpha}_i$ is the vector of displacement functions associated with the cover of node i. $\boldsymbol{\alpha}_i$ adopts the following form:

$$\boldsymbol{\alpha}_i = \begin{pmatrix} \alpha_{i,1} & \alpha_{i,2} & \alpha_{i,3} \end{pmatrix}^T \tag{3.22}$$

where $\alpha_{i,k}$ is the cover displacement function of node i in the x_k direction. Typically, the displacement functions $\alpha_{i,k}$ are trivariate polynomials of order N_i:

$$\begin{aligned} \alpha_{i,k} &= {}^{i,k}\beta_1 + {}^{i,k}\beta_2 x_1 + {}^{i,k}\beta_3 x_2 \\ &+ {}^{i,k}\beta_4 x_3 + {}^{i,k}\beta_5 x_1 x_2 + \ldots + {}^{i,k}\beta_{m_i} x_3{}^{N_i} \end{aligned} \tag{3.23}$$

where ${}^{i,k}\beta_1 \ldots {}^{i,k}\beta_{m_i}$ are the polynomial coefficients and m_i is the total number of coefficients of $\alpha_{i,k}$. The terms of $\alpha_{i,k}$ can be determined for any given order N_i from Figure 3-7 or Figure 3-8.

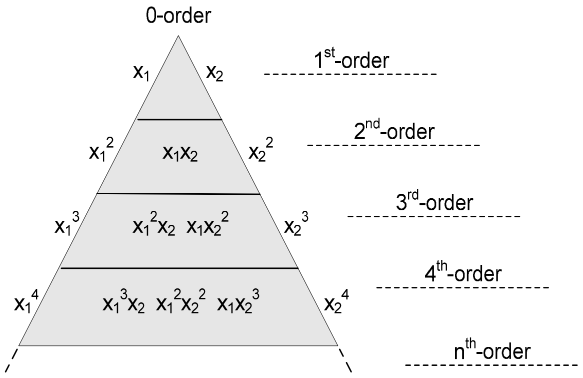


Figure 3-7. Schematic representation of displacement function monomials in two dimensions

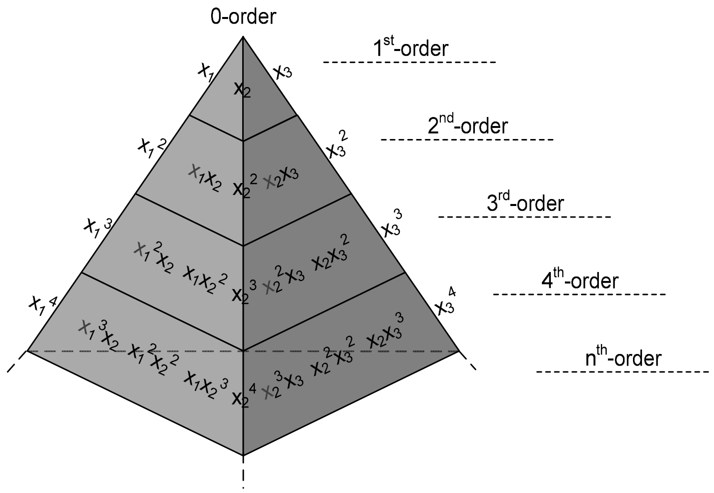


Figure 3-8. Schematic representation of displacement function monomials in three dimensions

The number of terms m_i of $\alpha_{i,k}$ of cover i is given by:

$$m_i = \frac{\prod_{l=1}^{p}\left(N_i + l\right)}{p!} \tag{3.24}$$

For example, if the displacement functions are zero-order ($N_i = 0$) in $\mathbb{R}^2$, then the number of terms according to Equation (3.24) is equal to:

$$m_i = \frac{\prod_{l=1}^{2}(0+l)}{2!} = 1 \tag{3.25}$$

and Equation (3.23) reduces to:

$$\alpha_{i,k} = {}^{i,k}\beta_1 \tag{3.26}$$

In this case, the displacement function equals a constant term; therefore equation (3.22) is a vector of nodal displacement contributions.

If the displacement function of the same cover is increased by one order of magnitude ($N_i = 1$), then:

$$m_i = \frac{\prod_{l=1}^{2}(1+l)}{2!} = 3 \tag{3.27}$$

and Equation (3.23) becomes:

$$\alpha_{i,k} = {}^{i,k}\beta_1 + {}^{i,k}\beta_2 x_1 + {}^{i,k}\beta_3 x_2 \tag{3.28}$$

Since the displacement function is a polynomial, the unknowns are the coefficients ${}^{i,k}\beta_1, {}^{i,k}\beta_2, \ldots$ and not nodal displacements as in the case with zero-order displacement functions.

Similarly, if in a three-dimensional domain ($p = 3$) the displacement functions are of first-order ($N_i = 1$), then:

$$m_i = \frac{\prod_{l=1}^{3}(1+l)}{3!} = 4 \tag{3.29}$$

$$\alpha_{i,k} = {}^{i,k}\beta_1 + {}^{i,k}\beta_2 x_1 + {}^{i,k}\beta_3 x_2 + {}^{i,k}\beta_4 x_3 \tag{3.30}$$

A closer examination of equations (3.26) and (3.21) reveals that in the special case that zero-order displacement functions are used, the unknowns are nodal displacements similar to FEM. In addition, for the simplex elements typically used in NMM the weighting function is a linear function of position and exactly the same as the FEM shape functions. Thus, for simplex elements, zero-order NMM and FEM yield exactly the same trial functions (see section 3.11).

It can be observed that the approximation identified by equation (3.21) may be improved not only by increasing the number of nodes n, and therefore undertaking remeshing, but also by increasing the order of $\alpha_{i,k}$. In this latter case, remeshing does not take place as the approximation is enhanced by augmenting the form of the displacement function rather than adding additional nodes.

Furthermore, it can be observed that the order N_i of each of the cover displacement functions in equation (3.23) associated with an element can be different for different nodes. Therefore, it is possible to enhance the displacement field only locally. This aspect constitutes an integral part of the hierarchical enhancement strategy treated later in the thesis.

3.5 Remarks regarding element technology

In principle, NMM covers can have arbitrary shapes and therefore it is not necessary to employ elements which are simplices. For example this has been shown by Sasaki & Ohnishi [91] in the context of NMM and analysis of rock mass using four-node iso-parametric elements, Cheng et al [25] with the development of Wilson non-conforming elements, and Chen et al [23] with the development of circular elements used to discretise geotechnical problems that consider the interaction of soil grains.

The main advantage of simplex elements (triangles in two dimensions, or tetrahedra in three dimensions) as opposed to elements with more complicated shapes is that they can be easily adapted to fit almost any geometry and as a result they are particularly convenient in *h*-adaptivity strategies. For example, it is possible to discretise domains into element shapes similar to those employed by recent polygonal finite element methods [106, 109], since polygons are in fact assemblages of simplices. There are several algorithms available in the public domain that can be used for the triangulation of complex two- or three-dimensional domains in a fully automated manner (Figure 3-9).

Furthermore, from a numerical point of view, polynomial descriptions of the approximation field can be integrated easily over simplex elements, whereas integration over more complex domains normally necessitates numerical integration.

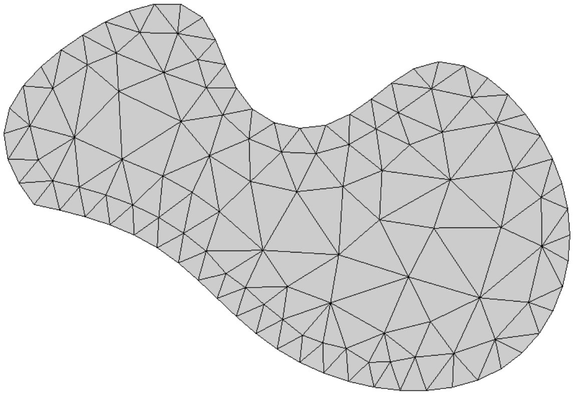


Figure 3-9. Discretisation of an arbitrary domain with an unstructured mesh of simplex elements using the GiD [46] pre- and post-processor

However, simplex elements are commonly associated with poor performance in the context of FEM with linear shape functions. This is due to the poor quality of deformability represented by FEM linear shape functions in three-node (in two dimensions) and four-node (in three-dimensions) interpolation schemes, which consequently results in constant strain fields within such domains.

In NMM with higher-order formulations (more than one unknown per node at each axis), similar performance issues do not apply, since element deformability improves significantly from the zero-order case and higher-order strain fields are recovered as discussed and demonstrated later. Integration can be undertaken analytically for any order of the approximation field. Furthermore, it can be argued that the use of effective or ineffective covers to model arbitrary boundaries can also negate the necessity to implement more complex element shapes for the purpose of discretising arbitrary geometries.

3.6 Discretised system of equations

In previous sections it was shown that the approximation to the unknown function within an element in three-dimensions is described by:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \sum_{i=1}^{n} w_i \, \boldsymbol{\alpha}_i \tag{3.31}$$

and that the weighting function w_i associated with cover i is typically defined as:

$$w_i = \left(a_i + b_i x_1 + c_i x_2 + d_i x_3\right) \tag{3.32}$$

If the coefficients and position components of $\boldsymbol{\alpha}_i$ are separated, then equation (3.32) may be alternatively cast in the following form:

$$\mathbf{u} = \sum_{i=1}^{n} \mathbf{T}_i \, \mathbf{a_i} \tag{3.33}$$

where $\mathbf{T}_i$ is a cover shape matrix which contains the position components of the displacement function vector $\boldsymbol{\alpha}_i$, multiplied by the weight function w_i:

$$\mathbf{T}_i = w_i \begin{pmatrix} 1 & 0 & 0 & x_1 & 0 & 0 & \ldots & x_3^{N_i} & 0 & 0 \\ 0 & 1 & 0 & 0 & x_1 & 0 & \ldots & 0 & x_3^{N_i} & 0 \\ 0 & 0 & 1 & 0 & 0 & x_1 & \ldots & 0 & 0 & x_3^{N_i} \end{pmatrix} \tag{3.34}$$

and $\mathbf{a}_i$ is a vector which contains the deformation coefficients of $\boldsymbol{\alpha}_i$:

$$\mathbf{a}_i = \left({}^{i,1}\beta_1 \quad {}^{i,2}\beta_1 \quad {}^{i,3}\beta_1 \quad \ldots \quad {}^{i,1}\beta_{m_i} \quad {}^{i,2}\beta_{m_i} \quad {}^{i,3}\beta_{m_i} \right)^T \tag{3.35}$$

Note that the subscript of β is the number of the coefficient term, while the left superscript is the node number, followed by the direction of the position component associated with the displacement function which contains β.

The cover shape matrix $\mathbf{T}_i$ has dimensions $p \times (m_i)\, p$ and its form and composition depends on the form of the weighting function, the spatial dimensions and the order of the cover displacement function.

For example, in two dimensions ($p = 2$), if zero-order displacement functions are used ($N_i = 0$), $m_i = 1$ so that $\mathbf{T}_i$ is a 2×2 matrix.

The cover deformation vector $\mathbf{a}_i$ has dimensions $(p \times m_i)$ and its shape and composition depends only on the spatial dimensions considered and the total number of terms of the cover displacement function (thus, the order of the displacement function).

The unknown function of (3.33) may be rewritten in a simplified form as:

$$\mathbf{u} = \mathbf{T}\,\mathbf{a} \tag{3.36}$$

where $\mathbf{T}$ and $\mathbf{a}$ are the shape matrix and deformation vector respectively. The form of $\mathbf{T}$ and $\mathbf{a}$ depends on the number of sub-matrices $\mathbf{T}_i$ and $\mathbf{a}_i$ respectively, and therefore on the number of covers, or nodes n associated with an element. $\mathbf{T}$ is defined as:

$$\mathbf{T} = \left[\mathbf{T}_1 \quad \mathbf{T}_2 \quad ... \quad \mathbf{T}_n \right] \tag{3.37}$$

For example, in the case of a two-dimensional simplex (three-node triangle), $n = 3$ so $\mathbf{T}$ is always a 1×3 matrix, containing sub-matrices of variable dimensions $2 \times \left(N_i + 1 \right) 2$ that depend on the order of cover displacement functions. For a three-dimensional simplex (four-node tetrahedron) $n = 4$, therefore $\mathbf{T}$ is 1×4 and contains sub-matrices of variable dimensions $3 \times \left(N_i + 1 \right) 3$.

Similarly, the deformation vector is defined as:

$$\mathbf{a} = \left[\mathbf{a}_1 \quad \mathbf{a}_2 \quad ... \quad \mathbf{a}_n \right]^T \tag{3.38}$$

For two-dimensional simplices ($n = 3$), $\mathbf{a}$ is a 3×1 vector of sub-vectors of variable dimensions $\left(2 \times m_i \right) \times 1$ which depend on the order of cover displacement functions employed. For three-dimensional simplices ($n = 4$), $\mathbf{a}$ is a 4×1 vector containing sub-vectors of dimensions $\left(3 \times m_i \right) \times 1$ which depends on the order of displacement functions employed at each individual cover.

Strains can be related to the unknowns $\mathbf{a}$ as:

$$\boldsymbol{\varepsilon} = \mathbf{L}\,\mathbf{T}\,\mathbf{a} = \mathbf{B}\,\mathbf{a} \tag{3.39}$$

The differential matrix $\mathbf{B}$ can be expressed in a general form as:

$$\mathbf{B} = \left[\mathbf{B}_1 \quad \mathbf{B}_2 \quad ... \quad \mathbf{B}_n \right] \tag{3.40}$$

where $\mathbf{B}_i$ are sub-matrices of dimensions that depend on the number of strain components and the number of terms of the shape sub-matrices $\mathbf{T}_i$. Similar to $\mathbf{a}$, the number of sub-matrices of the differential matrix depend on the number of nodes (two for one-dimensional elements, three for two-dimensional elements and so on).

From equations (3.39) and (3.40) it follows that in a three-dimensional case, the differential sub-matrices adopt the following general form:

$$
\mathbf{B}_i = \left[\begin{matrix}
\frac{\partial w_i}{\partial x_1} & 0 & 0 & \frac{\partial \left(w_i x_1\right)}{\partial x_1} & \cdots \\
0 & \frac{\partial w_i}{\partial x_2} & 0 & 0 & \cdots \\
0 & 0 & \frac{\partial w_i}{\partial x_3} & 0 & \cdots \\
\frac{\partial w_i}{\partial x_2} & \frac{\partial w_i}{\partial x_1} & 0 & \frac{\partial \left(w_i x_1\right)}{\partial x_2} & \cdots \\
0 & \frac{\partial w_i}{\partial x_3} & \frac{\partial w_i}{\partial x_2} & 0 & \cdots \\
\frac{\partial w_i}{\partial x_3} & 0 & \frac{\partial w_i}{\partial x_1} & \frac{\partial \left(w_i x_1\right)}{\partial x_3} & \cdots
\end{matrix}\right.
$$

$$
\left.\begin{matrix}
\cdots & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_1} & 0 & 0 \\
\cdots & 0 & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_2} & 0 \\
\cdots & 0 & 0 & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_3} \\
\cdots & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_2} & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_1} & 0 \\
\cdots & 0 & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_3} & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_2} \\
\cdots & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_3} & 0 & \frac{\partial \left(w_i x_3^{N_i}\right)}{\partial x_1}
\end{matrix}\right] \tag{3.41}
$$

The differential matrix will be discussed in more detail later.

3.7 *Weak form*

The weak form, or approximation of the equilibrium equations is obtained by multiplying equation (3.5) by a variation of the displacement $\delta\mathbf{u}$, integrating over the domain Ω and seeking a solution which makes stationary the resulting functional. In the case that the essential boundary conditions are imposed explicitly (as is commonly the case in FEM), then $\delta\mathbf{u}$ must be kinematically admissible; i.e. it must *a priori* satisfy the essential boundary conditions. However, in NMM, since shape functions do not interpolate nodal displacements, the essential boundary conditions must be imposed using either some form of Lagrange multipliers, or the penalty method. In the case of Lagrange multipliers:

$$\int_{\Omega} \delta\mathbf{u}^T \left(\mathbf{L}^T\boldsymbol{\sigma} + \mathbf{b}\right) d\Omega + \int_{\Gamma_u} \delta\left(\boldsymbol{\mathcal{L}}^T\mathbf{g}\right) d\Gamma = 0 \tag{3.42}$$

where $\mathbf{g}(\mathbf{u}) = \mathbf{u} - \hat{\mathbf{u}} = 0$ are the kinematic constraints and $\boldsymbol{\mathcal{L}}^T = [\mathcal{L}_x, \mathcal{L}_y, \mathcal{L}_z]$ are the Lagrange multipliers which can be physically interpreted as the reaction forces required to impose the prescribed displacements. Applying the divergence theorem, the first term in (3.42) can be expressed as:

$$\int_{\Omega} \delta\mathbf{u}^T \left(\mathbf{L}^T\boldsymbol{\sigma}\right) d\Omega = -\int_{\Omega} \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma}\, d\Omega + \int_{\Gamma_t} \delta\mathbf{u}^T \left(\mathbf{N}^T\boldsymbol{\sigma}\right) d\Gamma \tag{3.43}$$

Substituting the natural boundary conditions, incorporating (3.43) into (3.42) and expanding the Lagrange multiplier term results in:

$$\begin{aligned} &\int_{\Omega} \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma}\, d\Omega - \int_{\Gamma_u} \delta\boldsymbol{\mathcal{L}}^T\mathbf{g}\, d\Gamma - \int_{\Gamma_u} \delta\mathbf{u}^T\boldsymbol{\mathcal{L}}\, d\Gamma \\ &\qquad = \int_{\Omega} \delta\mathbf{u}^T\mathbf{b}\, d\Omega + \int_{\Gamma_t} \delta\mathbf{u}^T\hat{\mathbf{t}}\, d\Gamma \end{aligned} \tag{3.44}$$

The unknown $\mathbf{u}$ is discretised spatially according to (3.31) and $\boldsymbol{\mathcal{L}}$ is discretised in a similar manner as:

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} \mathcal{L}_x \\ \mathcal{L}_y \\ \mathcal{L}_z \end{bmatrix} = \begin{bmatrix} h_1 & 0 & 0 & h_2 & 0 & 0 & \cdots \\ 0 & h_1 & 0 & 0 & h_2 & 0 & \cdots \\ 0 & 0 & h_1 & 0 & 0 & h_2 & \cdots \end{bmatrix} = \mathbf{H}\boldsymbol{\lambda} \tag{3.45}$$

where $\mathbf{H}$ is typically taken equal to $\mathbf{T}$ although this does not necessarily have to be the case.

The respective variations of $\mathbf{u}$ and $\boldsymbol{\mathcal{L}}$ are discretised in a similar manner as:

$$\delta\mathbf{u} = \mathbf{T}\delta\mathbf{a} \tag{3.46}$$

$$\text{and } \delta\boldsymbol{\mathcal{L}} = \mathbf{H}\delta\boldsymbol{\lambda} \tag{3.47}$$

Substitution of these discretisations into (3.44) results in:

$$\begin{aligned} &\delta\mathbf{a}^T \int_\Omega \mathbf{B}^T \boldsymbol{\sigma}\, d\Omega - \delta\boldsymbol{\lambda}^T \int_{\Gamma_u} \mathbf{H}^T \left(\mathbf{T}\mathbf{a} - \hat{\mathbf{u}}\right) d\Gamma \\ &-\delta\mathbf{a}^T \int_{\Gamma_u} \mathbf{T}^T \mathbf{H}\, d\Gamma\, \boldsymbol{\lambda} = \\ &\qquad \delta\mathbf{a}^T \int_\Omega \mathbf{T}^T \mathbf{b}\, d\Omega + \delta\mathbf{a}^T \int_{\Gamma_t} \mathbf{T}^T \hat{\mathbf{t}}\, d\Gamma \end{aligned} \tag{3.48}$$

Assuming linear elasticity, stress is related to strain as:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{3.49}$$

It is worthwhile to note that although the constitutive relation does not have to be elastic, in subsequent chapters nonlinearities will be introduced in terms of discontinuous displacement fields rather than nonlinear material laws.

Therefore, equation (3.48) can be rewritten as:

$$\begin{aligned} &\delta\mathbf{a}^T \int_\Omega \mathbf{B}^T \mathbf{D}\mathbf{B}\, d\Omega\, \mathbf{a} - \delta\boldsymbol{\lambda}^T \int_{\Gamma_u} \mathbf{H}^T \left(\mathbf{T}\mathbf{a} - \hat{\mathbf{u}}\right) d\Gamma \\ &-\delta\mathbf{a}^T \int_{\Gamma_u} \mathbf{T}^T \mathbf{H}\, d\Gamma\, \boldsymbol{\lambda} = \\ &\qquad \delta\mathbf{a}^T \int_\Omega \mathbf{T}^T \mathbf{b}\, d\Omega + \delta\mathbf{a}^T \int_{\Gamma_t} \mathbf{T}^T \hat{\mathbf{t}}\, d\Gamma \end{aligned} \tag{3.50}$$

Since (3.50) must hold for arbitrary $\delta\mathbf{a}$ and $\delta\boldsymbol{\lambda}$, the following system of equations must hold:

$$\begin{bmatrix} \mathbf{K} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{q} \end{pmatrix} \tag{3.51}$$

where the first set of equations emanates from the $\delta\mathbf{a}$ terms in (3.50) and the second set of equations from the $\delta\boldsymbol{\lambda}$ terms.

The structural stiffness matrix is expressed as:

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbf{D}\mathbf{B}\, d\Omega \tag{3.52}$$

Matrix $\mathbf{A}$ couples displacement degrees of freedom to Lagrange multipliers:

$$\mathbf{A} = -\int_{\Gamma_u} \mathbf{T}^T \mathbf{H} \, d\Gamma \tag{3.53}$$

whereas the external forces are defined as:

$$\mathbf{f} = \int_{\Omega} \mathbf{T}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{T}^T \hat{\mathbf{t}} \, d\Gamma \tag{3.54}$$

The vector containing the applied displacements is defined as:

$$\mathbf{q} = -\int_{\Gamma_u} \mathbf{H}^T \hat{\mathbf{u}} \, d\Gamma \tag{3.55}$$

The integrals of equation (3.51) can be evaluated either numerically, or using simplex integration. Integration is a key aspect of the formulation presented here and will be discussed in depth later.

3.8 *Enforcement of essential boundary conditions*

Due to the non-interpolating nature of the weighting function with respect to nodal displacements, the enforcement of essential boundary conditions in NMM can be rather more complicated than it is in FEM.

Traditionally, essential boundary conditions in NMM (as well as DDA) [21, 65, 94, 96] have been imposed using the penalty method [10, 133]. Penalty constraints eliminate the problem mentioned above; however, the solution depends on artificially high stiffness constraints and the matrix system can be susceptible to ill-conditioning. Particular issues associated with the enforcement of constraints are examined later.

Alternatively, the enforcement of essential boundary conditions can be resolved in a more robust way by means of Lagrange multipliers [133]. In paradox, whereas this approach eliminates the requirement for artificial constraining forces, it introduces additional unknowns. However, it is possible to restore the problem to its original number of unknowns by modifying the variational principle, as discussed by Zienkiewicz in the context of FEM [133]. Furthermore, an alternative Lagrange multiplier technique presented by Ainsworth [1] in the context of Finite Elements and which restores the problem to its original number of unknowns is presented here.

Another approach is the direct enforcement of displacement constraints, as it is common in FEM [134]. As noted in the beginning of this section, this method is not straightforward when high-order displacement functions are employed. This issue can be alleviated by coupling with finite elements. This coupling resolution may not always be desirable in other numerical techniques (such as meshless methods) as it is not natural and requires some modification of the solution process. However, in NMM coupling with FEM can be effectively achieved in a natural way by treating the boundary with zero-order displacement functions. Section 3.11 shows that elements associated with zero-order functions result in identical trial functions as those employed by FEM, and Chapter 7 examines further the use of zero-order enforcement.

3.8.1 Lagrange multipliers

The essence of the Lagrange multiplier technique for enforcing essential boundary conditions is that each multiplier can be identified as a force used to constrain an equivalent nodal displacement. This was presented briefly in section 3.7. In order to complete this formulation it is necessary to give details of the choice of the Lagrange multiplier shape functions $\mathbf{H}$ for which there are a number of possibilities. However, the most straightforward approach is to use the point collocation method [133].

In this case, for a given point $\hat{x}_i$ on $\mathbf{\Gamma}_u$, the shape function is given as:

$$h_i = \delta\left(x - \hat{x}_i\right) \tag{3.56}$$

where δ is the Dirac delta function. Substitution of (3.56) into (3.53) and (3.55) results in:

$$A_{ij} = -\mathbf{T}_j^T\left(\hat{x}_i\right) \tag{3.57}$$

and:

$$q_i = \hat{u}\left(x_i\right) \tag{3.58}$$

The Lagrange multiplier method enforces the boundary condition exactly at the specified points on the boundary and the corresponding reactions forces are determined directly from solution of the discrete algebraic equations. The main disadvantage of the method is that the dimension of the resulting system of equations is increased and the global matrix is symmetric but no longer positive definite, thereby restricting the solvers that can be used.

It is worthwhile to note that it is possible to restore the problem to its original number of unknowns [133] with the use of Lagrange multipliers. Generally, this can be achieved by expressing the augmented parts of the functional with quantities that identify the Lagrange multipliers in a physical way, although a more mathematical approach is implemented in section 3.8.3.

3.8.2 Penalty constraints

The penalty method may be considered an approximation of the Lagrange multiplier method, whereby the Lagrange multiplier is approximated as follows:

$$\mathcal{L} \cong k\mathbf{g}(\mathbf{u}) \tag{3.59}$$

where k is a large positive number having the physical interpretation of a stiff spring constant in mechanics. With this approximation, (3.42) becomes:

$$\int_{\Omega} \delta\mathbf{u}^T \left(\mathbf{L}^T\boldsymbol{\sigma} + \mathbf{b}\right) d\Omega + \int_{\Gamma_u} \frac{1}{2} k\delta\left(\mathbf{g}^T\mathbf{g}\right) d\Gamma = 0 \tag{3.60}$$

Expressing the first term as an integration by parts and introducing the same spatial discretisation for $\mathbf{u}$ and $\delta\mathbf{u}$ as before, (3.60) results in:

$$\begin{aligned} &\delta\mathbf{a}^T \int_{\Omega} \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega\, \mathbf{a} - k\delta\mathbf{a}^T \int_{\Gamma_u} \mathbf{T}^T\mathbf{T}\mathbf{a}\, d\Gamma = \\ &\qquad - k\delta\mathbf{a}^T \int_{\Gamma_u} \mathbf{T}^T\hat{\mathbf{u}}\, d\Gamma + \delta\mathbf{a}^T \int_{\Omega} \mathbf{T}^T\mathbf{b}\, d\Omega + \delta\mathbf{a}^T \int_{\Gamma_t} \mathbf{T}^T\hat{\mathbf{t}}\, d\Gamma \end{aligned} \tag{3.61}$$

Since this must hold for arbitrary $\delta\mathbf{a}$, the following system of equations entails:

$$\mathbf{K}\mathbf{a} + k\mathbf{W}\mathbf{a} = \mathbf{f} + k\hat{\mathbf{f}} \tag{3.62}$$

where $\mathbf{K}$ and $\mathbf{f}$ are the same as before,

$$\mathbf{W} = -\int_{\Gamma_u} \mathbf{T}^T\mathbf{T}\, d\Gamma \tag{3.63}$$

and:

$$\hat{\mathbf{f}} = -\int_{\Gamma_u} \mathbf{T}^T\hat{\mathbf{u}}\, d\Gamma \tag{3.64}$$

The penalty method presents some advantages, in that the dimension of the system is not increased and the matrix in the resulting system is symmetric and positive definite. However, a shortcoming is that the essential boundary conditions are only imposed weakly, whereby the parameter k controls the degree to which they are enforced and the matrix $\mathbf{K}+k\mathbf{W}$ is usually ill-conditioned, since the condition number increases with k.

3.8.3 Enforcement of essential boundary conditions using projection matrices

The discrete system of equations can also be derived from the minimization of the discrete version of the modified (constrained) energy functional as:

$$\min_{\mathbf{a},\,\boldsymbol{\lambda}} \Pi = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{a} - \mathbf{f}^T\mathbf{a} + \boldsymbol{\lambda}^T\left(\mathbf{A}\mathbf{a}-\mathbf{q}\right) \tag{3.65}$$

Assuming this problem is well-posed, the following matrices are well-defined:

$$\mathbf{Q} = \mathbf{I} - \mathbf{R}\mathbf{A} \tag{3.66}$$

$$\mathbf{R} = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1} \tag{3.67}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{R}$ is an auxiliary matrix and $\mathbf{Q}$ is a projection matrix. Thus, there is a unique solution to the following modified problem:

$$\overline{\mathbf{K}}\mathbf{a} = \overline{\mathbf{f}} \tag{3.68}$$

where the modified stiffness matrix and force vector are defined as:

$$\overline{\mathbf{K}} = \mathbf{Q}^{\mathbf{T}}\mathbf{K}\mathbf{Q} + \mathbf{A}^{\mathbf{T}}\mathbf{A} \tag{3.69}$$

$$\mathbf{f} = \mathbf{A}^T\mathbf{q} + \mathbf{Q}^T\left(\mathbf{f} - \mathbf{K}\mathbf{R}\mathbf{q}\right) \tag{3.70}$$

and the corresponding Lagrange multipliers can be recovered from:

$$\boldsymbol{\lambda} = \mathbf{R}^T\left(\mathbf{f} - \mathbf{K}\mathbf{a}\right) \tag{3.71}$$

The constrained system has the same number of unknowns as the original problem and it can be solved using standard solvers. For computational efficiency, it is also possible to implement a sequential approach [1] rather than a single step using a global constraint matrix $\mathbf{A}$. Thus the approach offers clear advantages over both the Lagrange multiplier method and penalty method.

A simple procedure for obtaining the constrained system of equations in NMM is illustrated below.

```
% Compute constraint matrix A
for point=1:npoint      %loop constraints
    for elem=1:nel      %loop elements
            if current element contains P
                    Obtain nodal coordinates of 'elem'
                    Compute element shape matrix (T)
                    Augment A with contributions from T
            end
    end
end
R=A'*inv(A*A');
Q=I-R*A;
Kc=Q'*K*Q+A'*A;
fc=A'*g+Q'*(f-K*R*g);
b=Kc\fc;
```

Figure 3-10. MATLAB pseudo-code for computation of constraint system and derivation of unknowns

3.8.4 Direct enforcement of essential boundary conditions

In FEM, displacement boundary constraints are typically enforced by specifying nodal values and eliminating equations for which displacements are known. This approach is relatively simple to implement and ensures that boundary displacements vanish completely without the issues associated with the penalty constraints and Lagrange multipliers.

In NMM with higher-order displacement functions, direct enforcement of displacement constraints is not as straightforward as in FEM, hence it is usually attractive to employ penalty constraints and Lagrange multipliers as discussed earlier. This is due to the fact that the vector of unknowns is populated by terms that are not all necessarily nodal displacements.

Perhaps an obvious resolution is the coupling of NMM with finite elements at the boundary. Whereas this treatment is not natural in other methods (e.g. meshless methods) and generally requires the implementation of a mixed formulation, in NMM it can be a natural consequence of the choice of displacement functions employed at the boundary, without altering the formulation.

If elements attached to Γ_u are associated with constant cover displacement functions and linear weighting functions, they are equivalent to finite elements and thus displacement constraints can be imposed directly in the system matrices using the standard FEM approach. This equivalence has been briefly advocated in section 3.4 and it is verified later.

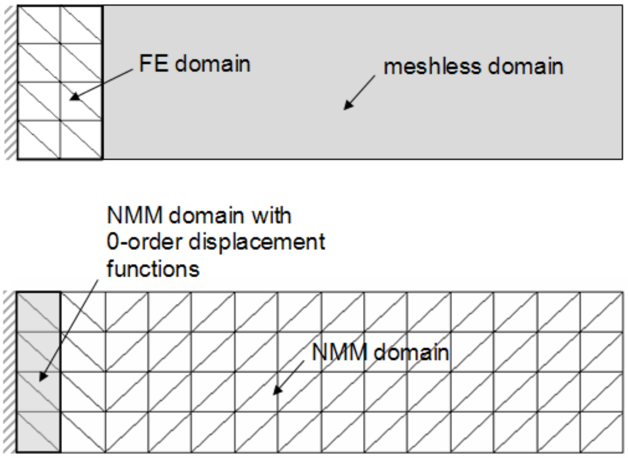


Figure 3-11. Direct enforcement of displacement constraints using a coupled FE-meshless approach (top) and NMM (bottom)

A drawback of this approach is that it depends on the choice of displacement polynomials and weighting functions and it can hence be perceived as limiting in terms of the order of the approximation employed. However, higher-order elements in NMM do not necessarily have to employ the same order of displacement polynomials at each node. Thus, it is possible to discretize the boundary with nodes that are associated with zero-order displacement functions while the interior nodes of the same boundary elements can be associated with higher-order functions (Figure 3-11). In this case the boundary unknowns are readily nodal displacements. The system equations for which values are known can then be eliminated in similar ways as in FEM.

The concept and implementation of this approach is developed in Chapter 4 while issues regarding its applications are examined in Chapter 7.

3.9 *Quasi-static and dynamic time discretisation*

Static and dynamic problems in NMM can be resolved using the same concepts applied in FEM and other methods.

In linear statics, the set of unknowns of the discretised system can be obtained directly using a standard elimination technique, such as Gauss elimination, or an iterative technique. However, in situations where there is a nonlinear relationship between stresses and strains, or between displacement jumps and tractions, and generally in cases where the load-displacement relationship bifurcates, then a form of an incremental approach is required. This is also true for transient problems.

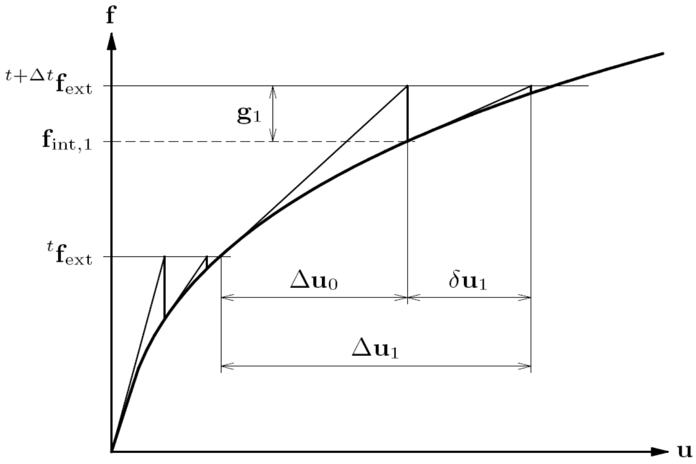
Whereas for given mesh discretisations linear statics are associated with unique solutions, in quasi-statics a single static, or a series of static solutions may be sought but the solution becomes path-dependent. In order to achieve equilibrium of internal and external forces in this latter case, the external forces or prescribed displacements have to be applied in a series of finite steps Δt .

In principle, the total applied increment (for example $\Delta \mathbf{u}$) is adapted by iterative increments $\delta \mathbf{u}$ until equilibrium is attained.

$$\Delta \mathbf{u}_{t+1} = \Delta \mathbf{u}_t + \delta \mathbf{u}_{t+1} \tag{3.72}$$

The iterative procedure typically linearizes (in each iteration) the nonlinear equations associated with this increment (Figure 3-12). A successful iteration terminates when a convergence criterion is satisfied. The convergence criterion can be typically a force norm, displacement or energy norm. As a new equilibrium is sought at each increment, this process implies that the tangent stiffness has to be assembled at each increment.

In this work, a Newton-Raphson procedure constrained by a force norm has been implemented. However, a wide range of other well-known iterative procedures [134] are also suitable.



f
$^{t+\Delta t}\mathbf{f}_{\text{ext}}$
$\mathbf{g}_1$
$\mathbf{f}_{\text{int},1}$
$^{t}\mathbf{f}_{\text{ext}}$
$\Delta\mathbf{u}_0$
$\delta\mathbf{u}_1$
$\Delta\mathbf{u}_1$
u

Figure 3-12. Newton-Raphson iterative procedure [115]

In lieu of the implicit iterative procedure suggested here, it is also possible to obtain a static or quasi-static solution using explicit central differences [28] with dynamic relaxation. In this case, the solution is approached as a steady-state resolution of the equation of motion using artificial damping.

An important advantage of this approach is that the requirement for assembly of the tangent stiffness at each increment is eliminated. However, Lagrange multipliers cannot be incorporated in the same way as previously [1], although it is possible to enforce displacement constraints directly on low-order NMM elements as discussed in section 3.8.4. An additional disadvantage is that mass scaling, which is undertaken to recover numerical stability, affects the solution path.

The above considerations were discussed in the context of negligible inertia forces. In applications where inertia effects are significant, the solution can be approached using either implicit or explicit schemes [28, 134].

In general, implicit approaches of highly nonlinear or highly dynamic problems (or both) may suffer from relatively large solution times, while explicit schemes may suffer from loss of stability. The choice of an optimal solution strategy depends ultimately on the physics of the particular problem under consideration, the presence and extent of constitutive and geometrically nonlinear effects, applied loads and their duration and the required level of accuracy, as well as project constraints.

3.10 Computational implementation

NMM, as it has been developed in all subsequent parts of the thesis has been implemented using MATLAB [71] as a computing environment and GiD [46] as a platform for pre- and post-processing (Figure 3-13).

MATLAB is a high-level programming language which also integrates a large library of pre-compiled and efficient mathematical functions, data visualisation and data analysis tools.

GiD is a customizable graphical interface for modelling, data input and results visualisation for different types of numerical techniques. It integrates sophisticated meshing algorithms and the possibility to program extensions using the Tcl/Tk language [111].

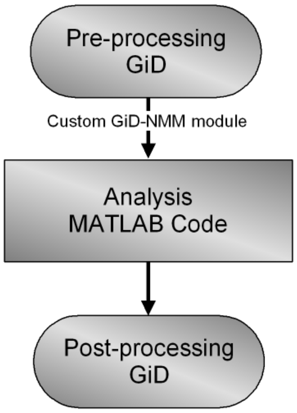


Figure 3-13. Simplified flowchart of analysis process

Pre-processing (discretisation, material properties, definition of boundary conditions, input of initial order of displacement polynomials at different parts of the domain, solution parameters) takes place in GiD. Using a custom module programmed with Tcl/Tk extensions, the complete problem input is translated into a format recognisable by the MATLAB NMM code.

The pre-processing process is in principle similar to the process undertaken to create a finite element model. However, the order N_i which defines the form of the displacement polynomials associated with nodes i can also be defined interactively for any individual node as an additional nodal property. If N_i is not prescribed then a default value of 0 is assumed (constant displacement functions). In an adaptive simulation, the prescribed (or default) values of N_i are considered as initial values.

The MATLAB code invokes the input file and performs analysis (assembly of the discretised system, enforcement of constraints, solution, and adaptivity). For debugging and speed purposes, the code typically stores all matrices and vectors in memory although optionally a swap file can be employed for analyses with large numbers of degrees of freedom. Once results have been obtained, the solution is saved in a format suitable for post-processing in GiD.

Results are typically saved per pseudo-time step within the MATLAB code. Nodal stress and strain averaging is undertaken during this stage, although it can also be undertaken during post-processing (in GiD). Furthermore, preparation of the post-processing mesh to visualise discontinuities in continuous-discontinuous problems (removal of overlapping regions as discussed in Chapter 5) is also undertaken during this stage.

Visualisation and analysis of results (deformation, contour plots, graphs) is undertaken using the post-processing module of GiD, once the results file prepared by MATLAB has been invoked.

3.11 Remarks regarding NMM and FEM

In previous parts it was advocated that there are similarities between NMM and other numerical methods, namely FEM, partition of unity methods, DDA and meshless methods. The principal conceptual connections between these methods and NMM arise from the way that the approximation field is constructed: the form of shape functions and the associated unknown functions. This section provides a closer examination on similarities between NMM and the most popular of these techniques: FEM.

Subsequent to the introduction of the generalised NMM displacement function (section 3.4), it was discussed that when zero-order displacement functions are employed on all covers associated with a simplex element, then the element is identical to a constant-strain finite element. Furthermore, the concept of *h*-enhancement of the approximation (by increase of the total number of nodes or decrease of element size) was advocated (similar to FEM), although it was also mentioned that, potentially, in NMM a better approximation can be also obtained by increasing the order N_i of the associated displacement polynomials.

Similar to FEM, the complete NMM domain has to be discretised with a finite number of elements, although covers may or may not be effective on discretised regions of the domain. Also, in both methods the approximation within an element is constructed as a form of:

$$\mathbf{u} = \mathbf{T}\,\mathbf{a} \tag{3.73}$$

where $\mathbf{T}$ is a shape function matrix and $\mathbf{a}$ is a vector of unknowns.

In NMM, $\mathbf{T}$ is a function of the weight function multiplied by the positional terms of the associated displacement polynomials. In essence this is also true for FEM, although FEM does not explicitly identify weighting functions. The NMM weighting function is typically a linear function of position and the same function is used for any order of the trial function. Furthermore, both NMM and FEM shape functions are partitions of unity [133, section 3.3].

In the simplest possible NMM case of simplex elements with zero-order displacement functions, $\mathbf{a}$ is populated by constants (since the displacement functions are zero-order), hence $\mathbf{T}$ is identified by the weight function alone which, as mentioned previously is a linear function of position. As a result, the unknowns of $\mathbf{a}$ are the nodal displacement unknowns.

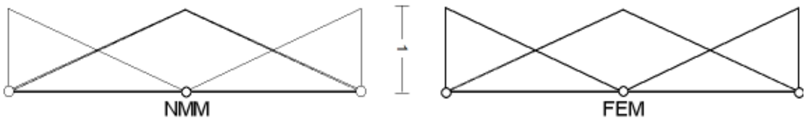


Figure 3-14. Shape functions of two one-dimensional elements in NMM with zero-order displacement functions (left) and standard FEM (right)

This is also true in FEM. If linear simplex elements are employed, then the shape functions are linear functions of position [133]. Hence $\mathbf{T}$ is populated by linear terms similar to NMM, and $\mathbf{a}$ consists of nodal displacement unknowns.

For example, in the case of a three-node constant strain triangle, in FEM the variation of the displacement function in each direction is defined by a linear expansion with three terms [133]:

$$u(x_1, x_2) = \beta_1 + \beta_2 x_1 + \beta_3 x_2 \tag{3.74}$$

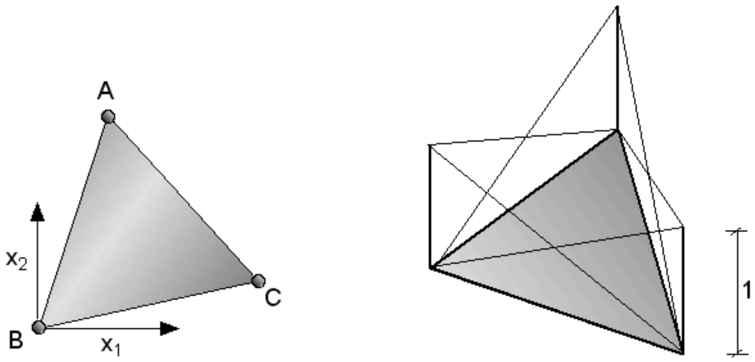


Figure 3-15. Three-node two-dimensional element and shape functions

or:

$$u = \mathbf{P}\boldsymbol{\beta} \tag{3.75}$$

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \tag{3.76}$$

If the coordinates of nodes A, B and C (Figure 3-15) are substituted in Equation (3.74), the following system can be constructed to determine the coefficients β:

$$\begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix} = \begin{bmatrix} 1 & x_{1,A} & x_{2,A} \\ 1 & x_{1,B} & x_{2,B} \\ 1 & x_{1,C} & x_{2,C} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{3.77}$$

or alternatively:

$$\mathbf{u} = \mathbf{C}\boldsymbol{\beta} \tag{3.78}$$

Therefore, equation (3.75) can be alternatively expressed as:

$$u = \mathbf{P}\mathbf{C}^{-1}\mathbf{u} \tag{3.79}$$

or:

$$u = \mathbf{T}\,\mathbf{u} \tag{3.80}$$

$$\mathbf{T} = \mathbf{P}\mathbf{C}^{-1} \tag{3.81}$$

In NMM the variation of the displacement function within an element is defined as the product sum of weighting functions and cover displacement functions (section 3.4). In zero-order NMM, since the cover displacement functions are constant, the displacement variation will be governed by the weighting function:

$$w\left(x_1, x_2\right) = \beta_1 + \beta_2 x_1 + \beta_3 x_2 \tag{3.82}$$

or:

$$w = \mathbf{P}\boldsymbol{\beta} \tag{3.83}$$

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \tag{3.84}$$

If nodal coordinates are substituted in Equation (3.82), the coefficients β can be determined from:

$$\begin{bmatrix} w_A \\ w_B \\ w_C \end{bmatrix} = \begin{bmatrix} 1 & x_{1,A} & x_{2,A} \\ 1 & x_{1,B} & x_{2,B} \\ 1 & x_{1,C} & x_{2,C} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{3.85}$$

or:

$$\mathbf{w} = \mathbf{C}\boldsymbol{\beta} \tag{3.86}$$

$$\boldsymbol{\beta} = \mathbf{C}^{-1}\mathbf{w} \tag{3.87}$$

Since:

$$u = w\,\mathbf{u} \tag{3.88}$$

It follows that:

$$u = \mathbf{P}\boldsymbol{\beta}\,\mathbf{u} \tag{3.89}$$

However, using the definition given in section 3.3 and the Kronecker delta, the weighting functions equal unity at the centre of covers and zero at centres of neighbouring covers. Therefore, substituting equation (3.87) into (3.89) yields:

$$u = \mathbf{P}\mathbf{C}^{-1}\mathbf{I}\mathbf{u} \tag{3.90}$$

Hence,

$$\mathbf{T} = \mathbf{P}\mathbf{C}^{-1} \tag{3.91}$$

which is identical to equation (3.80) and the FEM shape functions in (3.81).

FEM can improve the approximations not only by reducing the element size (introducing more nodes), but also by introducing new nodes in the same element using standard shape functions. Since the points that determine the variation of the displacement function increase, consequently equation (3.74) and $\mathbf{P}$ expand and adopt more complicated polynomial forms. As a result, the shape functions in equation (3.81) change and the computation process has to be repeated.

In NMM, rather than introducing new nodes, additional unknowns can be introduced in the displacement function, at existing nodal supports with the use of higher-order polynomials. The same weighting functions are preserved for any order of the displacement function.

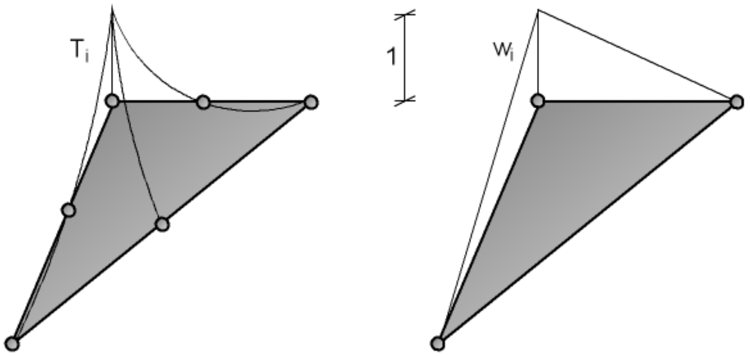


Figure 3-16. Higher-order triangular element: Left: FEM with standard shape functions. Right: NMM with linear weighting functions

This approach is conceptually similar to the hierarchical approximation approach in FEM and it is associated with distinct advantages with regard to the integration process, as will be discussed later, and conditioning of higher order elements in steady-state problems [133]. The similarities of hierarchical FEM with higher-order NMM are discussed later in more detail.

Therefore, the principal conceptual difference between NMM and FEM with standard shape functions arises from the way elements are constructed in each case. Whereas traditionally, in FEM a better approximation can be obtained by decreasing the element size or introducing new nodes in elements, in NMM a better approximation can be achieved either by decreasing the element size (similar to FEM) or by introducing higher-order polynomial displacement functions, without introducing new nodes.

In this latter case essential boundary conditions cannot be enforced directly as in FEM and instead Lagrange multipliers, penalty constraints or coupling techniques have to be considered, although an elegant alternative is that of local enforcement with zero-order displacement functions as proposed in Chapter 7.

Furthermore, in NMM strong discontinuities can be modelled by exploiting the partition of unity property as discussed in the following chapter, wherein no *a priori* assumptions are required regarding the potential discontinuity path. On the other hand in FEM discontinuities are traditionally modelled using remeshing techniques or interface elements which are defined *a priori*.

In Chapter 5 it will be shown that the same discontinuous-modelling concepts employed in NMM can also be adopted in FEM. In fact, the development of XFEM [73, 74, 75], which appears as an extension of FEM and advocates similar concepts for modelling of discontinuities without remeshing, is a strong example that this is possible.

3.12 Concluding remarks

The Numerical Manifold Method has been introduced and extended for the analysis of continuous and discontinuous problems in solid mechanics. In particular, the improvements developed in this chapter can be summarised as follows:

1. NMM was constructed in a variational form for three-dimensional domains
2. The approximating function was extended for a generalised case of higher-order deformability within simplex elements. The resulting system matrices may appear more involved than the original formulations by Shi [96] and Chen [21] for zero and first-order functions, but they are cast in a general form for one-, two- and three-dimensional spaces, for any arbitrary order of displacements functions.
3. Essential boundary conditions were satisfied using penalty constraints, Lagrange multipliers and a technique based on projection matrices. In addition, the possibility to enforce essential boundary conditions directly (similar to FEM) in higher-order approximations was introduced.

Furthermore, the concepts of local enhancement of the approximation field and modelling strong discontinuities using partitioned covers, without remeshing, were introduced. These concepts can be particularly significant in failure simulations, in which the ability to represent evolving discontinuities in robust, efficient and accurate ways is generally desired.

In the final part of this chapter, parallels between NMM and FEM were drawn. Comparisons point towards conceptual similarities as, under specific conditions, the two methods appear to arrive at the same conclusions although cast from relatively different angles. In addition, NMM appears to materialise as an integrated array of several powerful modelling aspects (partition of unity, hierarchical enhancement), although several of these aspects appear analogous to traditional and more recent developments in finite element techniques.

4 Higher-order approximation and hierarchical local enhancement

4.1 Introduction

The basis of NMM for solids in two and three dimensions was laid in Chapter 3, while it was advocated that the solution can be potentially improved by expanding the polynomial functions used to approximate the displacement field up to any arbitrary level, without introducing new nodes and hence without undertaking remeshing.

The problems of improvement, convergence and adaptivity are intrinsic and universal in numerical approximation techniques, whether continua or discontinua are considered, and whether discontinuities are introduced using remeshing or other strategies.

In principle, a numerical solution can be improved by increasing the number of unknowns employed to define the approximation field, so that the trial field becomes a closer representation of the actual field. Assuming that the trial field is capable of reproducing the displacement form of the continuum (or discontinuum), then with refinement or enrichment of the approximation with additional unknowns it may be possible to obtain the exact solution.

This can be achieved in two ways:

1. By adapting the level of discretisation. In mesh-dependent methods this can be undertaken via remeshing or via modification of the existing mesh without changing the number of elements, nodes or connectivity. The former is referred to as *h*-refinement and the latter *r*-refinement.
2. By use of higher-order basis functions. Instead of introducing more unknowns by increasing the number of nodes, the approximation can be enhanced by expanding the local basis functions used to approximate the unknown field (*p*-refinement). This is typically undertaken in a hierarchical manner.

Both approaches are equally applicable in NMM. However, in order to preserve the main benefit obtained by the partition of unity approach of NMM, which is the introduction of discontinuities without remeshing as discussed in Chapter 5, there is a distinct advantage in investigating the ability to improve the approximation using higher-order functions. Furthermore, the possibility for hierarchical improvement of the approximation field without remeshing appears to be a particularly attractive option for adaptivity, as it entails almost identical implementations for problems of any spatial dimension and mesh structure.

This chapter examines the following:

1. The ability to improve the approximation without remeshing

2. The ability to improve the approximation locally, without remeshing
3. The ability to improve the approximation adaptively, without remeshing

Although the foundations for the first remark above have been laid [21, 65] in the context of NMM, the practical implementation of the ability to improve the approximation up to any arbitrary level has not been demonstrated in literature. To the best of the author's knowledge, the second and third remarks have not been examined whatsoever in the context of NMM.

This chapter begins with the development of two alternative strategies for constructing higher-order systems in a general three-dimensional case. Both strategies are illustrated with algorithmic examples. A strategy for local hierarchical enhancement of the approximation is discussed and remarks regarding deformability and boundary discretisation are made.

The section then moves to discuss the exploitation of local higher-order enhancement using *p*-adaptivity and error indicators. A single element benchmark test is devised to investigate convergence and higher-order deformability. Furthermore, the global and local higher-order enhancement and adaptivity strategies discussed here are illustrated using numerical examples.

4.2 *Discretised higher-order system*

In Chapter 3, it was shown that the general discretised system can be cast in a form of the familiar relationship:

$$\mathbf{K}\,\mathbf{a} = \mathbf{f} \tag{4.1}$$

where:

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega \tag{4.2}$$

and:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \ldots & \mathbf{B}_n \end{bmatrix} \tag{4.3}$$

where n is the number of nodes associated with an element.

However, in contrast to traditional finite element techniques, the unknown vector $\mathbf{a}$ can consist of terms that are not necessarily nodal displacements, unless the associated nodal displacement functions are monomials which are also independent of position.

Furthermore, it was discussed that although the approximation can be enhanced by increasing the number of nodes associated with an element or the complete domain (similar to FEM with standard shape functions) while the number of unknowns per node remains constant, it is also possible to enrich the approximation by increasing the order of displacement function polynomials associated with nodes. In this latter case, the number of unknowns per node increases, and the unknowns have no direct physical interpretation although they are related to displacement degrees of freedom via the cover displacement function.

The cover displacement functions can be of any general polynomial order N. It is worthwhile to note that the displacement functions at a particular node associated with a given element can be different from the displacement functions of another node associated with the same element. Furthermore, the displacement function of a particular node in a given direction could be different from the displacement function employed in another direction.

Using Equation (3.24), it is observed that by increasing the order N_i of the displacement function polynomial associated with node i, the number of unknowns increases linearly in one dimension, quadratically in two-dimensions and in an exponential manner in three-dimensions (Figure 4-1). For example, in two dimensions, the number of unknowns per node increases from two (for the zero-order case), to 30 unknowns for the fourth-order case. This can be viewed crudely as discretising an area which is initially represented by a single three-node constant-strain triangular element (six unknowns), with 14 three-node constant-strain triangular elements.

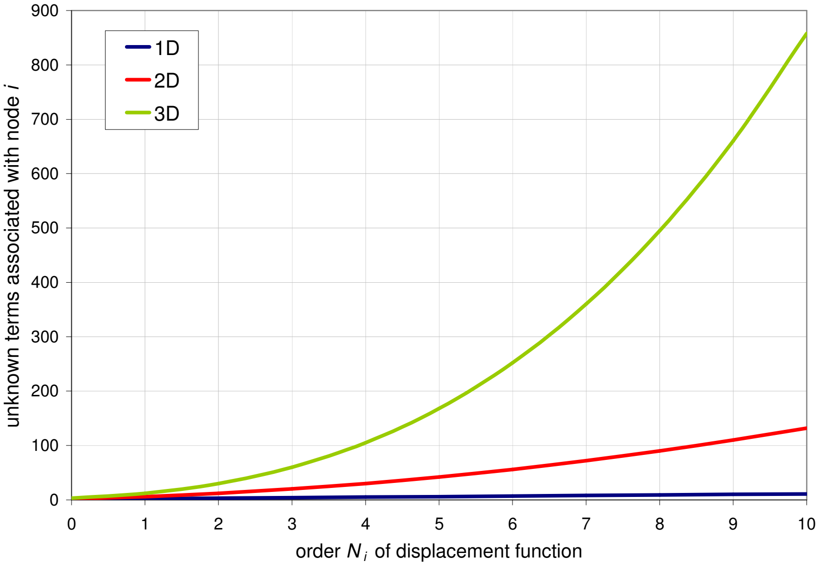


Figure 4-1. NMM unknowns for different orders (N_i) of displacement polynomials for the one-, two- and three-dimensional case, for a given node i

When higher-order displacement functions are used, each sub-matrix $\mathbf{B}_i$ in equation (4.3) can consist of several terms of nonlinear functions of the spatial components (see equation (3.41)) and as a result each term of the stiffness matrix $\mathbf{K}$ in equation (4.2) can be an expansion of tens, or even several thousands of nonlinear sub-terms. Clearly, for any general higher-order case, the analytical derivation and hard-coding of the complete discretised system is limited and impractical.

The following sections develop two alternative methods with which higher-order systems can be constructed algorithmically. This is achieved without making simplifications, so that the resulting systems are exactly those which would be derived by the analytical process. It is implied that an integral of the type:

$$\iiint_V x_1^{n1} x_2^{n2} x_3^{n3} dx_1 dx_2 dx_3 \qquad \forall \left(n_1, n_2, n_3\right) \in \mathbb{N}^3 \tag{4.4}$$

is available and the reader is referred to Chapter 6 where the analytical and algorithmical derivation of exact integrals of this type is explained.

4.3 Sub-matrix method

One possible way of computing the stiffness matrix of a general higher-order system is by using the sub-matrix method. A variant of this approach was originally implemented by Lu [65] for the two-dimensional case for problems that are of global order N. Here, the approach is extended in three-dimensions and it is generalised for problems in which not all nodal displacement functions are of the same order (section 4.7).

The sub-matrix method relies in explicit hard-coding of sub-matrices of the global stiffness matrix. Once the sub-matrices are available, the global stiffness matrix is assembled algorithmically.

For clarity, a detailed explanation of the sub-matrix method is omitted from this chapter and presented in Appendix D.

4.4 Multi-dimensional matrix method

An alternative method for constructing the stiffness matrix of an arbitrary higher-order problem is proposed here, which can be undertaken utilising products of multi-dimensional arrays. The multi-dimensional array products are used to assemble the matrix which defines the stress-strain relation, and therefore the stiffness matrix. Multi-dimensional array capabilities are available in most popular programming codes, such as MATLAB, C++ and FORTRAN.

A key difference between this approach and the sub-matrix method is that there is no derivation and requirement to hard-code explicitly parts of the $\mathbf{B}$ matrix. As a result, the implementation of this method is less laborious than the sub-matrix method. Also, implementation of the multi-dimensional matrix approach for a generalised one-, two- and three-dimensional case entails minimal differences. However, the approach discussed here can be more demanding in terms of computational resources and it can also be more difficult to implement than the sub-matrix method.

Similar to the sub-matrix method, from the definition of the weighting function, each term of the sub-matrix $\mathbf{B}_i$ that contains a derivative with respect to $\mathbf{x}$ can be expressed in the general form given in Equations (12.7), (12.8) and (12.9).

Since integration can be undertaken explicitly (Chapter 6) and in order to make the computer implementation easier, the terms of sub-matrices $\mathbf{B}_i$ can be stored in multi-dimensional arrays $\overline{\mathbf{B}}_i$. Each matrix $\overline{\mathbf{B}}_i$ has dimensions $r \times p(N_i+1) \times (N_i+1) \times (N_i+1) \times (N_i+1)$, where r is the number of strain components.

The first two dimensions of $\overline{\mathbf{B}}_i$ correspond to the location of terms of $\mathbf{B}_i$ whereas the third dimension of $\overline{\mathbf{B}}_i$ corresponds to the exponents of x_1 terms, the fourth dimension corresponds to the exponents of x_2 terms and so on. For example, for a two-dimensional problem ($p=2$) with three strain components ($r=3$), $\overline{\mathbf{B}}_i$ will be $3 \times 2(N_i+1) \times (N_i+1) \times (N_i+1)$.

For clarity, further explanation of the approach is provided in Appendix E. It is worth noting that multiplication of the multi-dimensional arrays used to evaluate element stiffness matrices is an important algorithmical challenge associated with the approach. Appendix F explains how to implement this.

4.5 Boundary discretisation

Enrichment with higher-order displacement functions can yield improved element deformability, which consequently results in non-homogeneous strain state within higher-order elements. Due to this reason, enforcement of constraints using any method requires particular care to ensure that boundary edges or surfaces between restrained nodes are also adequately restrained. This issue is treated in Chapter 7.

4.6 Higher-order deformability

The enhanced element deformability resulting from enrichment with higher-order displacement functions can produce a better correlation between the approximation and the actual displacement field, and potentially yield a better solution (Figure 4-2). Furthermore, the stress and strain fields are also enhanced.

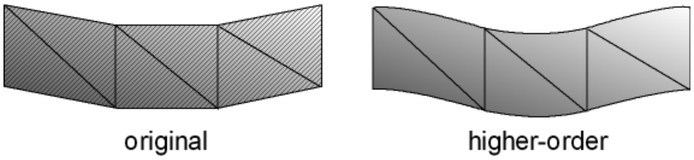
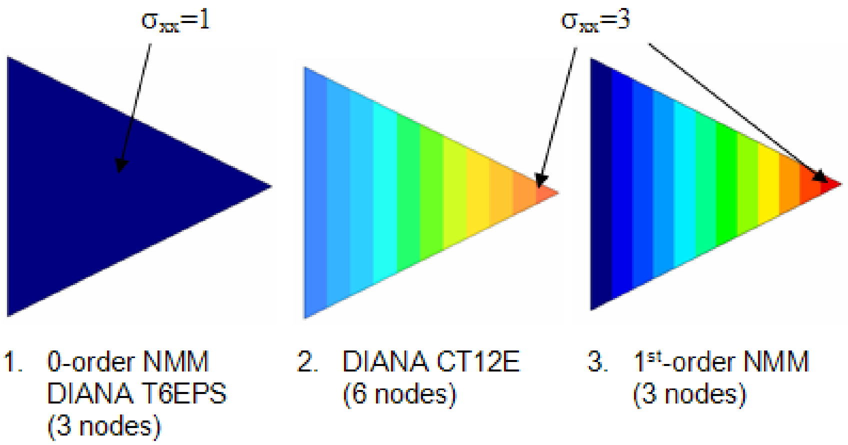


Figure 4-2. Original and higher-order deformability of a beam constrained at both ends

For example, a constant-strain element which is enriched with higher-order displacement functions at any of its nodes will produce a variable strain state. The results can be comparable to the FEM situation where additional nodes are introduced in the element while the standard shape functions are retained (Figure 4-3). Here however, the number of nodes remains constant and the displacement and strain fields can be enhanced up to any arbitrary level without undertaking any form of remeshing.



1. 0-order NMM DIANA T6EPS (3 nodes)
2. DIANA CT12E (6 nodes)
3. 1st-order NMM (3 nodes)

Figure 4-3. Stress contours of: 1- Single three-node triangular plane-strain element (DIANA element T6EPS and NMM with 0-order displacement functions yield exactly the same constant stress field). 2- Single six-node triangular isoparametric plane-strain finite element test (DIANA element CT12E). 3- Single three-node triangular plane-strain NMM element with 1st -order displacement functions. In all cases the two left-hand nodes are fixed whereas a point load of 0.5 is applied on the right-hand node in the horizontal direction. E = 1000, $\nu = 0$.

In order to visualise better the enhanced deformability of higher-order elements it is useful to employ a secondary mesh of triangles within each element during the post-processing stage, thereby enabling the use of standard FE post-processing software to be utilised. Displacements at these additional nodes and stresses within the additional elements can be determined *a posteriori*. Evidently, if the mesh is relatively fine, these *a posteriori* considerations to enhance visualisation of deformability are not necessary.

4.7 *Algorithmical issues of hierarchical local enhancement*

The difference between global and local enhancement is that in the former case, the displacement polynomials of all nodes of a given problem are of the same order N, while in the latter case the displacement polynomials of each node i can be of different order N_i.

From the point of view of practical implementation, this means that the contribution to the stiffness matrix or the force vector from the degrees of freedom of a node associated with displacement polynomials of order N_i, can be different for each node i and will be governed by the number of unknowns (defined as m_i previously).

While the derivation of local contributions associated with any arbitrary order N_i (as functions of shape matrices $\mathbf{T}_i$ and their derivatives) has been considered by the theoretical and implementational considerations of previous sections, the only remaining question is how to assemble such dissimilar local contributions to the global system.

It is known that, for example, the stiffness contribution of node i in a three-dimensional case is of $(3m_i)\times(3m_i)$ dimensions (three times the number of unknowns in each direction). The stiffness contribution of node $i+1$ is of $(3m_{i+1})\times(3m_{i+1})$ dimensions. If the same problem is simply re-numbered so that i becomes $i+2$, the position of contributions associated with that node in the global system changes.

In general, there are $n!$ combinations, where n is the total number of nodes, with which this re-numbering can be undertaken. Therefore derivation of a direct analytical expression for the identification of the location of contributions of each node in any general case is impractical. However, this issue can be approached algorithmically in an iterative manner.

The algorithm of Figure 4-4 offers such an example. This example returns the starting row and column of a general discretised two-dimensional system, where the stiffness or force contributions from the degrees of freedom of nodes enriched with displacement polynomials of any arbitrary order have to be included.

```
% It is assumed that this algorithm is located within a loop of the form:

% for i=1:3

%         for j=1:mi

%                 for r=1:3

%                         for l=1:mr

% where i & r are nodes of a given element, j and l are degrees of freedom
% associated with nodes i & j respectively. Furthermore, node i has mi
% unknowns and is associated with polynomials of Ni order whereas node r
% has mr unknowns and is associated with polynomials of Nr order

Row = 0;

for ii=1:Ni

    Ntem = … % calculate order of polynomial associated with ii

    mtem = (Ntem+1)*(Ntem+2)/2;

    if (ii==Ni)

            DOF = 2*j-1;

    else

            DOF = 2*mtem;
    end
    for jj=1:DOF
        Row = Row + 1;
    end
end

Column = 0;
for rr=1:Nr
    Ntem = … % calculate order of polynomial associated with rr
    mtem = (Ntem+1)*(Ntem+2)/2;
    if (rr==Nr)
        DOF = 2*l-1;
    else
        DOF = 2*mtem;
    end
    for ll=1:DOF
        Column=Column+1;
    end
end
```

Figure 4-4. Pseudo-algorithm for the calculation of the starting location of local contributions of nodes associated with displacement functions of any arbitrary order in a general two-dimensional system

4.8 Adaptivity

Numerical approximations are intrinsically associated with errors and it is often desired to minimise inaccuracies by improving the solution using some form of estimators, which provide estimates of the relative error of the solution, or indicators, which provide cruder estimates of relative errors, or provide indication of where the solution has to be improved rather than how much.

Early in the chapter it was discussed that the NMM approximation can be improved using *h*-refinement strategies, similar to traditional finite element techniques, or using *p*-refinement by adapting the order of displacement polynomials that define the approximation field. This latter approach can in principle be undertaken by:

1. Increasing the displacement polynomial order uniformly throughout the domain (global refinement)

2. Increasing hierarchically the displacement polynomial order in a local level (local enhancement).

A distinct advantage of the latter case is that, since the approximation can be improved locally by increasing the order of displacement polynomials, then with the use of error estimators or error indicators it may be possible to improve the solution by simply adapting locally the order of nodal displacement functions from N_i to $N_i + 1$, or $N_i + N_{es}$, where N_{es} is the enhancement step.

Therefore, the approximation is potentially improved locally without undertaking any remeshing, therefore with optimal pre-processing effort. Within the developed higher-order framework discussed in previous sections, this form of adaptivity is relatively straightforward to implement.

It is worthwhile to note that since this type of adaptive strategy is based only on local adaptation of the order of displacement polynomials (by increasing N_i based on certain criteria), the procedure is identical for problems of any spatial dimension, mesh structure and element type unlike *h*-refinement strategies.

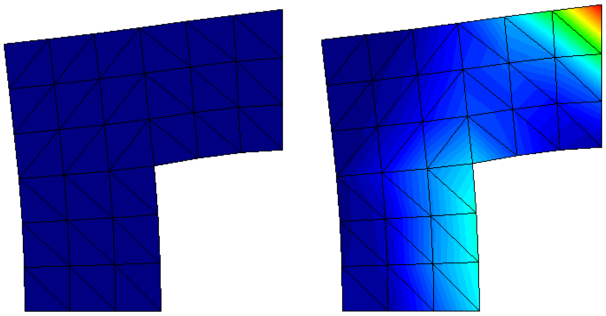


Figure 4-5. Analysis of L-shaped domain with singularity. Step 1: contours of zero-order displacement functions (left) and major principal stress (right). The bottom horizontal edge is fully fixed whereas the right-hand vertical edge is allowed to translate in the vertical direction only. Pressure acting upwards is applied on the top horizontal edge.

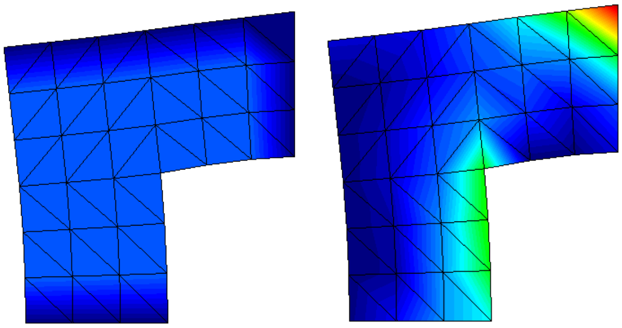


Figure 4-6. Analysis of L-shaped domain with singularity. Step 2: contours of first-order displacement functions around the singularity (left) and major principal stress (right). The bottom horizontal edge is fully fixed whereas the right-hand vertical edge is allowed to translate in the vertical direction only. Pressure acting upwards is applied on the top horizontal edge.

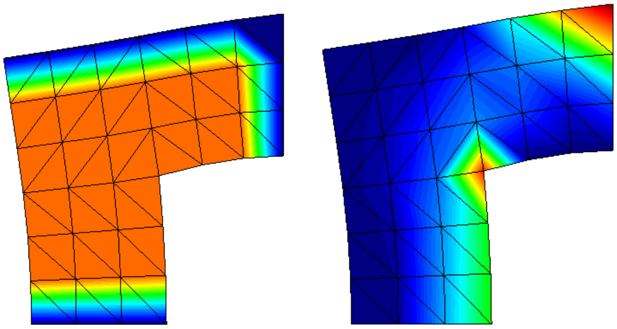


Figure 4-7. Analysis of L-shaped domain with singularity. Step 5: contours of fourth-order displacement functions around the singularity (left) and major principal stress (right). The bottom horizontal edge is fully fixed whereas the right-hand vertical edge is allowed to translate in the vertical direction only. Pressure acting upwards is applied on the top horizontal edge.

Here, a simple automated *p*-adaptive procedure has been implemented whereby error indicators are used to provide an estimate of the relative local error and hence, an indication of where the approximation requires improvement in order to minimise error. The adopted error indicators are based on:

1. Stress criteria based on local errors of individual components of stress tensors compared to a smoothed solution.
2. Displacement criteria based on local errors of two consecutive solutions.

In the first case, the error approximation technique is based on the standard Zienkiewicz-Zhu technique for error-estimation in finite elements [133].

Similar to finite elements, in NMM the continuity assumption used in displacement-based formulations results in a continuous displacement field across elements, but a discontinuous stress field. To obtain more acceptable stresses, averaging of the element nodal stresses can be undertaken. The stress error vector at each node of the element considered is defined as:

$$\Delta\boldsymbol{\sigma}_i^e = \boldsymbol{\sigma}_i^a - \boldsymbol{\sigma}_i^e \tag{4.5}$$

where $\Delta\boldsymbol{\sigma}_i^e$ is the stress error vector at node i of element e, $\boldsymbol{\sigma}_i^a$ is the averaged stress vector at node i and $\boldsymbol{\sigma}_i^e$ is the calculated stress vector of node i of element e.

Ideally, if the stress field is smooth then stress at a particular node would be the same for each element associated with that node, so that the averaged stress would be the same as the calculated stress. In this case, the error would be zero. If $\Delta\boldsymbol{\sigma}_i^e$ exceeds a prescribed limit, then in order to improve the approximation the order of the displacement function associated with that node can be increased from N_i to N_i+1.

The displacement-based approach is similar, but requires initially two solutions to be derived: the original solution and an improved solution so that relative error can be indicated. In this case:

$$\Delta\mathbf{u}_i = \mathbf{u}_i^{N_i+1} - \mathbf{u}_i^{N_i} \tag{4.6}$$

where $\Delta\mathbf{u}_i$ is the displacement error vector at node i, $\mathbf{u}_i^{N_i+1}$ is the displacement vector calculated using a displacement function which is increased from the original order N_i of that node by one, whereas $\mathbf{u}_i^{N_i}$ is the displacement vector calculated using a displacement function of order N for node i.

Based on the assumption that an enhanced solution will provide a better approximation, the second solution is obtained by increasing the order of displacement functions uniformly by a single order. If the prescribed error is exceeded locally, then the order of the associated displacement function (of the original solution) is increased from N_i to N_i+1 and a new solution is obtained.

Combinations of the above two criteria are possible. Furthermore, it is possible to prescribe a step function for increasing polynomials by two or more steps to achieve faster convergence. Options for prescribing automatically boundaries of enriched problems with zero-order polynomials in order to enforce displacement continuity are also available. Similarly, within the context of a discontinuous modelling framework, it is possible to enhance automatically only nodes within a prescribed radius in the vicinity of a discontinuity tip if this is required, in order to improve initiation and orientation estimators (Figure 4-8).

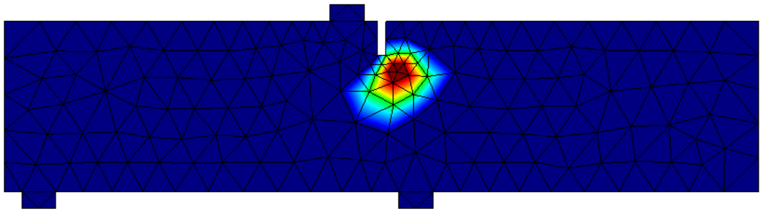


Figure 4-8. SEN beam model with enhanced displacement functions in the vicinity of the originating discontinuity. The coloured region illustrates the enhancement zone while blue represents the region associated with zero-order displacement functions.

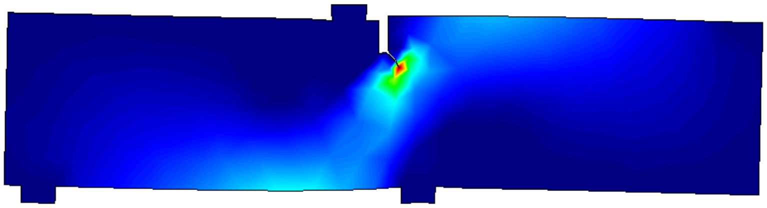


Figure 4-9. Major principal stress contours of refined SEN model.

Other *a posteriori* error indicators or estimates can also be used to drive the adaptive process or measure the error of the calculated solution. One such example is the energy norm error estimator [133] which is used in examples presented later in this chapter. For elasticity problems, the energy norm error for each element can be written as:

$$\left\| e^e \right\| = \left[\int_{\Omega^e} \left\{ \Delta\boldsymbol{\sigma}^e \right\}^T \mathbf{E}\, \Delta\boldsymbol{\sigma}^e \, d\Omega \right]^{\frac{1}{2}} \tag{4.7}$$

where $\Delta\boldsymbol{\sigma}^e$ is the stress error vector of element e, evaluated from all $\Delta\boldsymbol{\sigma}_i^e$ of this element, $\mathbf{E}$ is the elasticity matrix and $\boldsymbol{\Omega}^e$ is the element volume. The energy error of the complete domain can be calculated as:

$$\left\| e \right\| = \sum_{i=1}^{m} \left\| e^i \right\| \tag{4.8}$$

where m is the total number of elements. Furthermore, the relative energy norm error can be calculated as:

$$\eta = \frac{\|e\|}{\|u\|} \times 100 \; (\%) \tag{4.9}$$

where $\|u\|$ is the energy norm of the calculated solution.

A benchmark problem which utilises the implemented adaptive strategy is illustrated in section 4.12.

Clearly, there are cases in which convergence using any criteria is bound to fail. For example, where singularities exist or where point loads are applied (Figure 4-7). Furthermore, it is worthwhile to note that the use of interpolation with high-order polynomials at equidistant points can introduce errors as the solution tends to oscillate with increasing polynomial orders at interpolation intervals [87]. This is known as 'Runge's phenomenon' (Figure 4-10). Potential remedies are the use of spline curves or the use of Chebyshev nodes that become increasingly closer near boundaries.

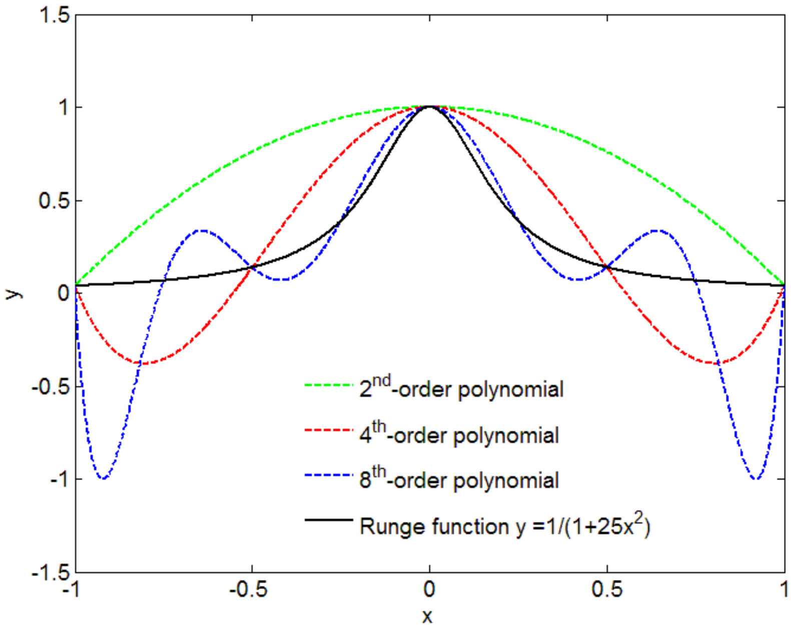


Figure 4-10. Runge's phenomenon: interpolation of Runge's function with high-order polynomials

Therefore, although the type of adaptivity introduced here can in principle improve a solution with reduced pre-processing effort, it requires an active level of attention and engineering judgement in order to avoid spurious results.

4.9 *Single element benchmark test*

In order to examine the performance and convergence of elements enhanced with higher-order displacement functions, the following benchmark test is considered. The test consists of a single three-node triangular plane-stress element (Figure 4-11).

Nodes two and three are restrained in the horizontal (x) direction, while node three is also restrained in the vertical (y) direction. A force equal to 0.5 is applied at node one in the positive x-direction. The elastic modulus is taken as 100, whereas Poisson's ratio is assumed to be zero.

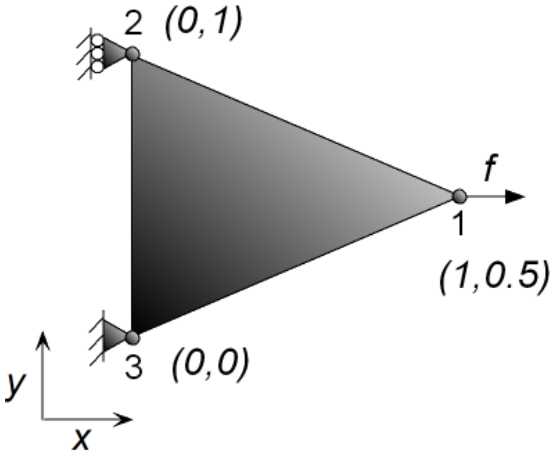


Figure 4-11. Single element test

The number of unknowns is:

$$3\times\sum_{i=1}^{3}\left(N_i+1\right)\left(N_i+2\right) \tag{4.10}$$

where $N_i \in \mathbb{N}$ is the order of displacement polynomials associated with nodes 1, 2 and 3.

A closed-form solution can be sought by perceiving the problem as a limiting case of a tapered bar problem (Figure 4-12), in which the tip cross-sectional area tends to zero.

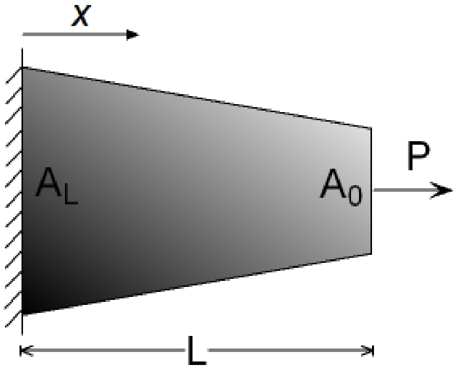


Figure 4-12. Tapered bar problem with end load P

From the basic definition of stress, the axial stress at any position x along the bar is:

$$\sigma_x = \frac{P}{A_x} \tag{4.11}$$

Assuming that the cross-sectional area A_x varies linearly from the constrained end A_L to the free end A_0, (4.11) can be re-written as:

$$P = \sigma_x \left(A_0 + \frac{A_L - A_0}{L} x \right) \tag{4.12}$$

Using Hooke's law, the strain along x can be derived from:

$$\varepsilon_x = \frac{\sigma_x}{E} = \frac{P}{EA_x} \tag{4.13}$$

Since one end is fully constrained and strain is a derivative of displacement, the displacement at x can be given as the integral of ε_x from 0 to x. Therefore, substituting (4.12) in (4.13) and integrating yields:

$$u(x) = \frac{P}{E} \int_0^x \frac{1}{A_0 + \frac{A_L - A_0}{L} x} dx \tag{4.14}$$

It can be shown that (4.14) is equal to:

$$u(x) = \frac{P}{aE}\left[\ln\left| A_0 + \frac{A_L - A_0}{L} x \right| - \ln\left| A_0 \right| \right] \tag{4.15}$$

where $a = (A_L - A_0)/L$.

From (4.13) and (4.15) it can be observed that if $A_0 = 0$ and x approaches L, strain and displacement approaches infinity; i.e. the tip of the problem is a singularity. For small values of A_x the displacement within the domain will not be affected due to St. Venant's effect.

The solution of the zero-order case is identical to the constant strain finite element case. This is shown here for reference.

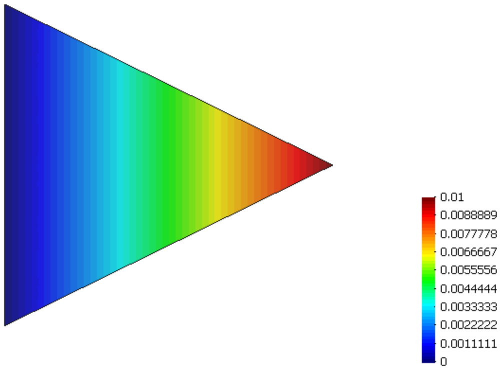


Figure 4-13. Zero-order element: deformed shape and contours of displacement along the horizontal axis

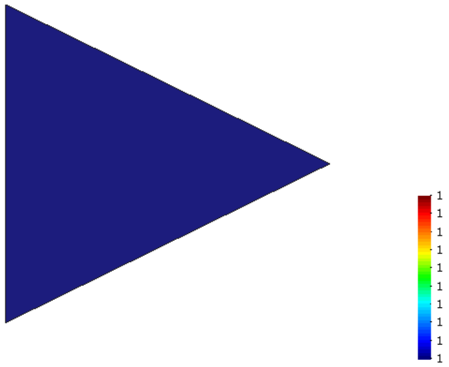


Figure 4-14. Zero-order element: deformed shape and contours of stress in the horizontal axis

Node	u_x	σ_{xx}
1	0.01	1
2	0	1
3	0	1

Table 4-1. Zero-order test: nodal results

If the order of displacement polynomials of nodes one, two and three is increased uniformly by a single order of magnitude, a first-order solution is recovered (Figure 4-15).

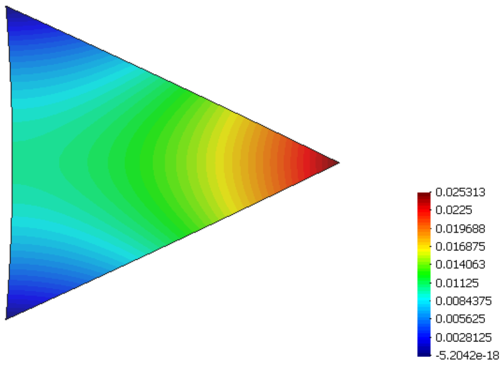


Figure 4-15. First-order element: deformed shape and contours of displacement along the horizontal axis

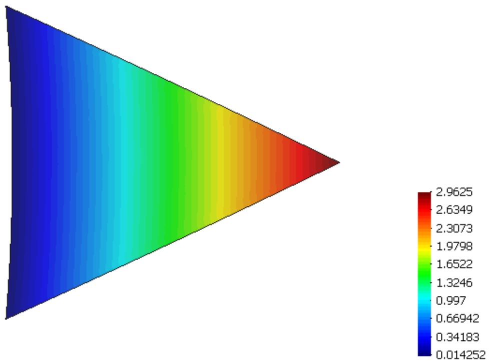


Figure 4-16. First-order element: deformed shape and contours of stress in the horizontal axis

Node	u_x	σ_{xx}
1	0.025	3
2	0	1E-12

3	0	2.2E-12

Table 4-2. First-order test: nodal results

Similarly, it can be shown that for a second-order element the solution is:

Node	u_x	σ_{xx}
1	0.04	6
2	0	3.26
3	0	3.26

Table 4-3. Second-order element: nodal results

Evidently, the increased order of displacement polynomials in the higher-order cases leads to an increasing strain state within the same three-node domain.

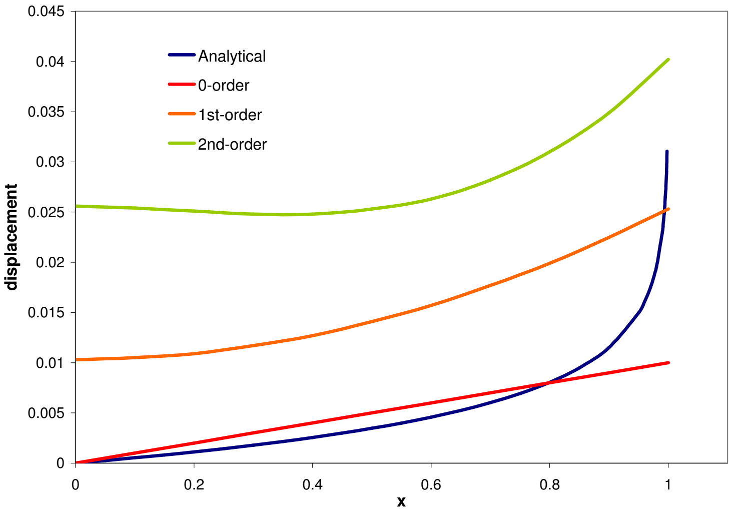


Figure 4-17. Variation of displacement along x

From Figure 4-17 it can be observed that higher-order enrichment affects the boundary between points were essential conditions are imposed. Thus the problem is restated such that sufficient constraints are imposed to ensure that no horizontal displacement occurs along the left hand edge of the element. The updated displacement graph is displayed in Figure 4-18. The displacement approximation in this case is an improved representation of the analytical field, although convergence appears to become slow as x approaches the singularity. Figure 4-19 to Figure 4-26 illustrate the deformed element, horizontal displacement distribution and stress distribution for the zero-order to fourth-order case.

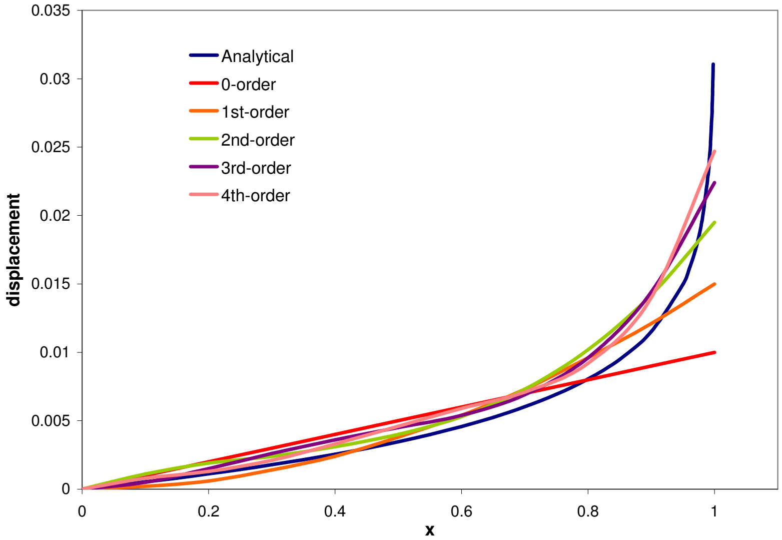


Figure 4-18. Variation of displacement along x, with enforced displacement continuity along the constrained boundary

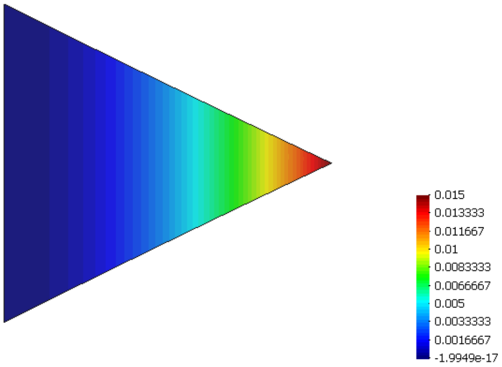


Figure 4-19. First-order element: deformed shape (×2) and contours of displacement along the horizontal axis

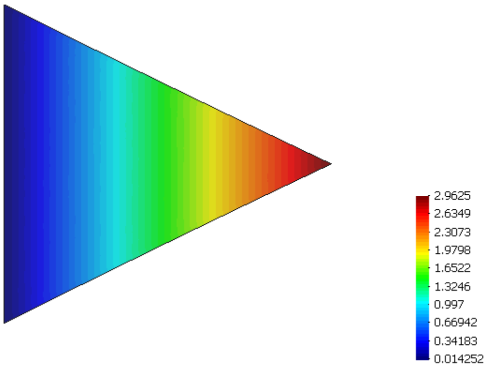


Figure 4-20. First-order element: deformed shape (×2) and contours of stress along the horizontal axis

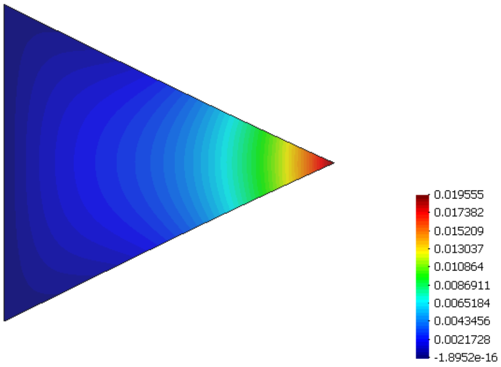


Figure 4-21. Second-order element: deformed shape (×2) and contours of displacement along the horizontal axis

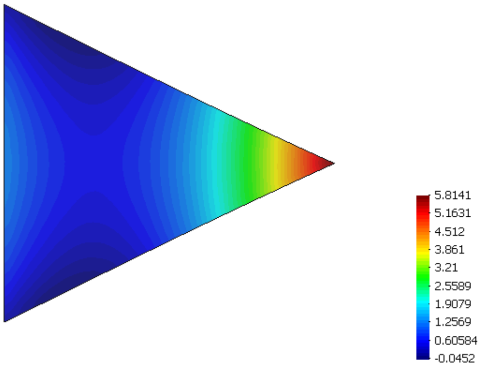


Figure 4-22. Second-order element: deformed shape (×2) and contours of stress along the horizontal axis

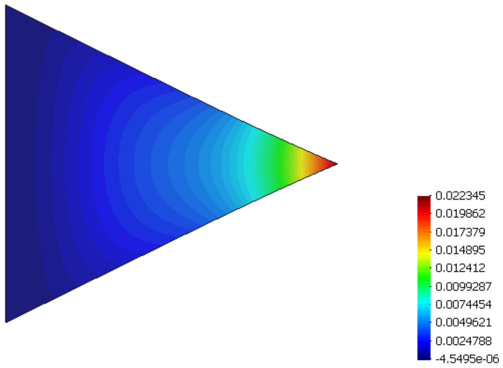


Figure 4-23. Third-order element: deformed shape (×2) and contours of displacement along the horizontal axis

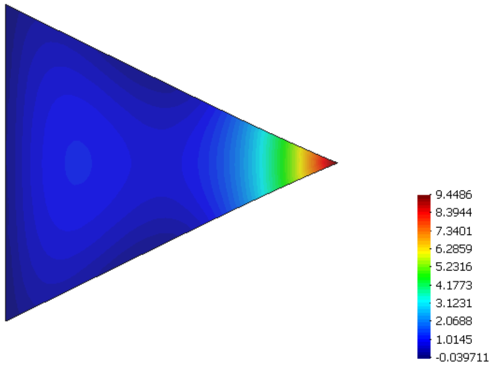


Figure 4-24. Third-order element: deformed shape (×2) and contours of stress along the horizontal axis

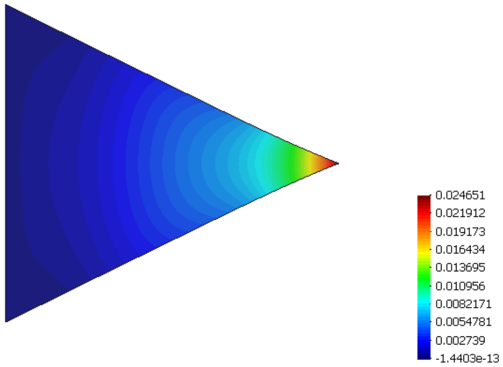


Figure 4-25. Fourth-order element: deformed shape (×2) and contours of displacement along the horizontal axis

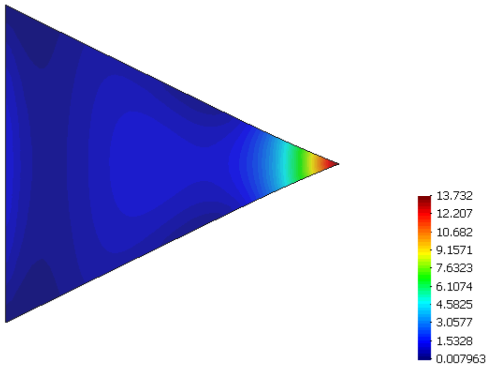


Figure 4-26. Fourth-order element: deformed shape (×2) and contours of stress along the horizontal axis

4.10 *Example: Constrained beam in uniaxial tension*

In order to demonstrate the convergence characteristics of enhancement with higher-order displacement functions, a number of illustrative problems is presented. The first problem represents a one-dimensional beam restrained at both ends and loaded at its mid-point with a point load (Figure 4-27). The elastic modulus, element length L and cross-sectional area are taken as 1. The force load is taken as 4/3.

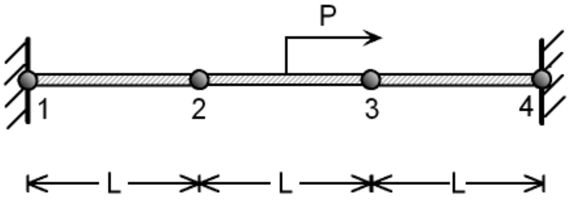


Figure 4-27. Problem definition and discretisation

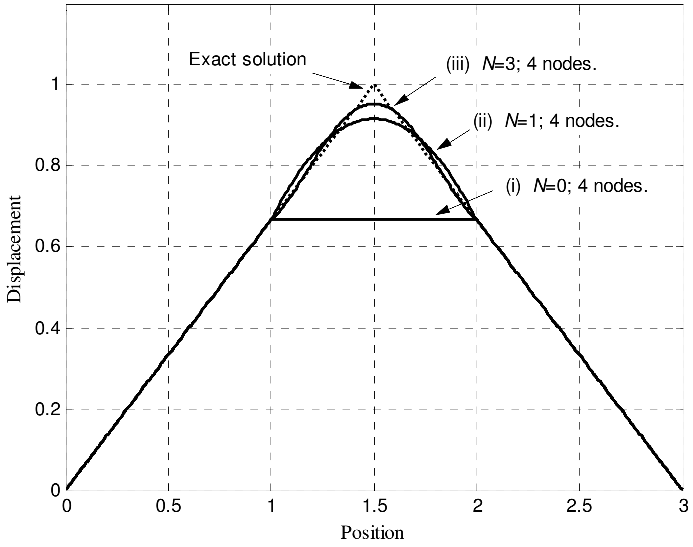


Figure 4-28. Displacement versus position for different orders of displacement polynomials

Figure 4-28 illustrates the displacement versus position graphs resulting from various analyses. It is worthwhile to note that a bifurcation point in the strain field exists at the mid-point of the beam due to the chosen loading arrangement. In the particular case that a node exists at the loading location in the centre of the beam, the NMM solution will be exact.

However, in the case that a node does not exist at the loading point, the discontinuity in the strain field will be difficult to capture using continuous displacement functions. Therefore, to make the problem interesting, the beam has been discretised crudely into three equal elements and four nodes, ensuring that loading is not applied on a node.

The remaining graphs in Figure 4-28 represent various solutions corresponding to different orders of the displacement function. Graph (i) shows the solution using zero-order displacement functions. In this case, the analysis cannot resolve the strain discontinuity at the centre of the beam and the result is zero strain in the central element. The same solution is also representative of the FEM solution using three linear elements.

Graph (ii) shows the result for the same number of nodes (and elements) as graph (i) but using first-order displacement functions associated with each node. The resolution of displacement is now somewhat improved and tends to a more accurate representation of the exact solution. The resolution can be enhanced further by using higher-order displacement functions. For example, graph (iii) displays the result from a third-order analysis with four nodes. Once again, it can be seen that the solution is approaching the exact solution.

It is interesting to note that a second-order solution offers no improvement upon the equivalent first-order analysis. This is due to the symmetric one-dimensional nature of the particular problem considered: additional degrees of freedom associated with the second-order displacement functions provide no enhancement to the first-order case. If the loading point is moved off-centre then an improvement on the first-order analysis would be observed for this particular case.

The same problem is now revisited considering various local enhancement strategies (Figure 4-29). The beam is discretised, as before, into three equal elements and four nodes, with no node positioned at the centre.

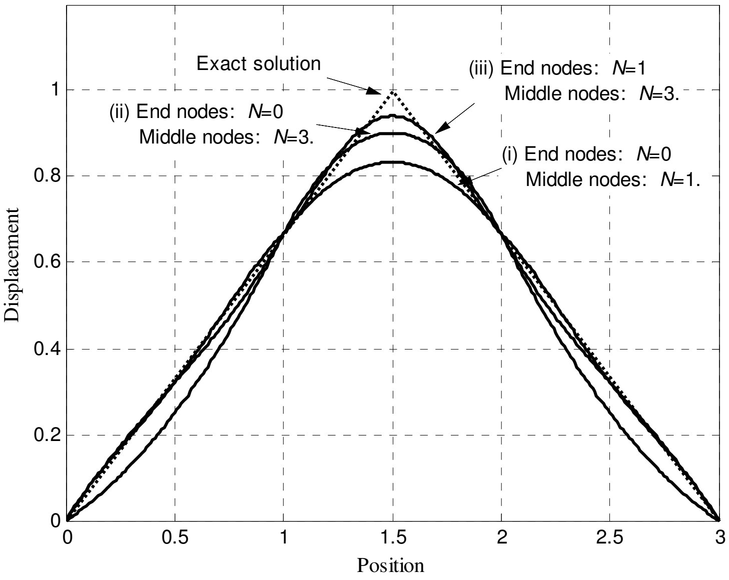


Figure 4-29. Displacement versus position for different local enhancement strategies

Graph (i) in Figure 4-29 illustrates the result from an analysis with zero-order displacement functions at the two end-nodes, and first-order functions at the two remaining nodes. This solution does not offer an improvement to the uniform first-order case (Figure 4-28, graph ii), but it offers a better approximation than the model with uniform zero-order functions (Figure 4-28, graph i).

The approximation is improved further by increasing the displacement polynomials associated with the two internal nodes to third-order (graph ii). Finally, graph (iii) illustrates the solution which corresponds to enhancement of the end-node displacement polynomials to first-order, while adopting third-order functions at all other nodes. It is interesting to note that perturbations are observed in the displacement field in the outer two elements. Once again, for this particular problem introducing second-order functions offers no improvement to the equivalent first-order case.

4.11 *Example: Semi-infinite plate with hole*

The following problem of a semi-infinite membrane subject to uniaxial tension is considered in order to investigate the effect of global and local enhancement with higher-order displacement polynomials. The problem consists of a 400×200 mm membrane of thickness 1 mm (Figure 4-30) subject to a uniform axial pressure of 100 MPa. A traction-free circular hole of 20 mm radius is situated in the middle of the membrane to weaken the section. The elastic modulus and Poisson's ratio are taken as 100,000 MPa and 0.3 respectively. The problem is idealised in plane-stress.

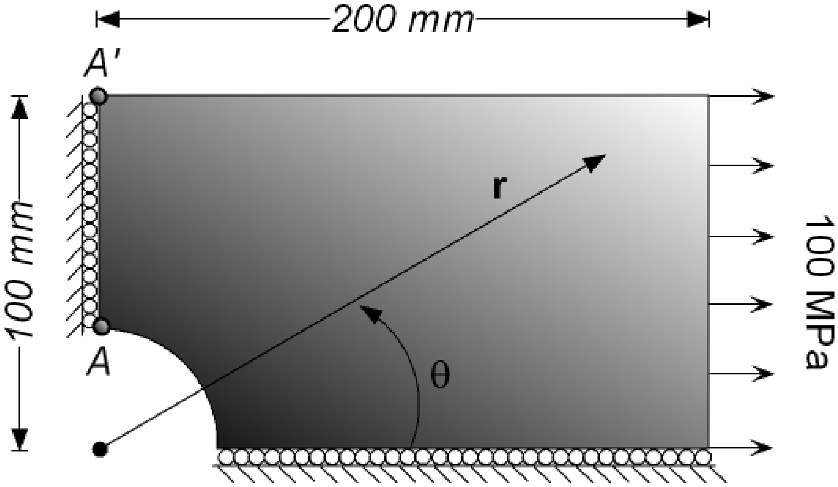


Figure 4-30. Definition of quadrant idealisation of semi-infinite perforated plate; (r, θ) are polar coordinates

Normally, due to two-fold symmetry it is sufficient to model only a quadrant of the problem. However, since displacement continuity at constrained boundaries is not automatically enforced (see section 4.9 and Chapter 7) when higher-order polynomials are employed at the boundary, the symmetry case will be incorrect as it is desired to enrich elements in the vicinity of the hole.

This pathology associated with higher-order boundaries can be successfully treated by enforcing a constant-strain state along boundaries using zero-order displacement functions (Chapter 7). However, since it is desired to enrich the region in the vicinity of the hole with higher-order functions, here it is necessary to consider the full model and enforce essential boundary conditions with zero-order functions on the actual boundary.

A relatively coarse discretisation comprising 278 three-node triangular elements is considered (Figure 4-31). This corresponds to 164 nodes, or 328 degrees of freedom from the zero-order case which is equivalent to a finite element model comprising of constant strain triangles.

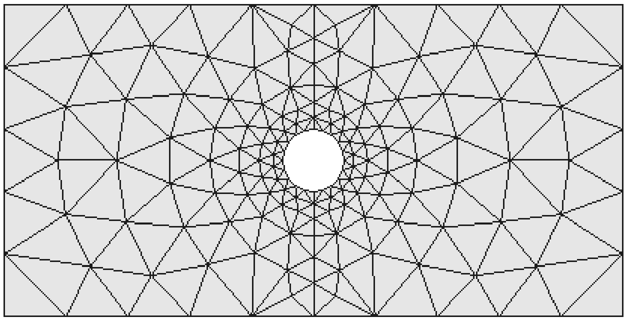


Figure 4-31. Problem discretisation

Traction conditions are imposed on the boundaries of the model of Figure 4-31 based on the analytical solution [114], while the hole remains traction-free:

$$\sigma_{xx} = P\left[1-\frac{a^2}{r^2}\left(\frac{3}{2}\cos 2\theta+\cos 4\theta\right)+\frac{3a^4}{2r^4}\cos 4\theta\right] \tag{4.16}$$

$$\sigma_{yy} = P\left[-\frac{a^2}{r^2}\left(\frac{1}{2}\cos 2\theta-\cos 4\theta\right)-\frac{3a^4}{2r^4}\cos 4\theta\right] \tag{4.17}$$

$$\sigma_{xy} = P\left[-\frac{a^2}{r^2}\left(\frac{1}{2}\sin 2\theta+\sin 4\theta\right)+\frac{3a^4}{2r^4}\sin 4\theta\right] \tag{4.18}$$

where (x, y) are the horizontal and vertical Cartesian coordinates respectively, (r,θ) are polar coordinates originating from the centre of the traction-free hole with θ measured counter-clockwise from x (see Figure 4-30), P is the applied tensile stress and a is the hole radius. Displacements are calculated from:

$$\begin{aligned} u_x = \frac{a}{8\mu}\Bigg[&\frac{r}{a}(\kappa+1)\cos\theta+2\frac{a}{r}\left((1+\kappa)\cos\theta+\cos 3\theta\right) \\ &-2\frac{a^3}{r^3}\cos 3\theta \end{aligned} \tag{4.19}$$

$$\begin{aligned} u_y = \frac{a}{8\mu}\Bigg[&\frac{r}{a}(\kappa-1)\sin\theta+2\frac{a}{r}\left((1-\kappa)\sin\theta+\sin 3\theta\right) \\ &-2\frac{a^3}{r^3}\sin 3\theta \end{aligned} \tag{4.20}$$

where $\mu = E/\left[2(1+\nu)\right]$, $\kappa = (3-\nu)/(1+\nu)$ for plane stress, while E and ν is the elastic modulus and Poisson's ratio respectively.

The mesh is created deliberately coarse in order to examine the effect of approximation improvement via *p*-refinement alone using zero, first, second, third and fourth-order displacement polynomials. However, along those boundaries where essential boundary conditions and loads are applied, nodes will be deliberately associated with zero-order polynomials at all cases considered – the reason for this will be discussed in full in Chapter 7.

Prior to analysing the problem using higher-order displacement functions, it is worthwhile to examine NMM convergence using *h*-refinement with remeshing and zero-order displacement functions (in essence finite element analysis with constant-strain triangles). Four additional discretisations from the original of Figure 4-31 are considered, with the mesh becoming increasingly refined primarily around the hole (Figure 4-32):

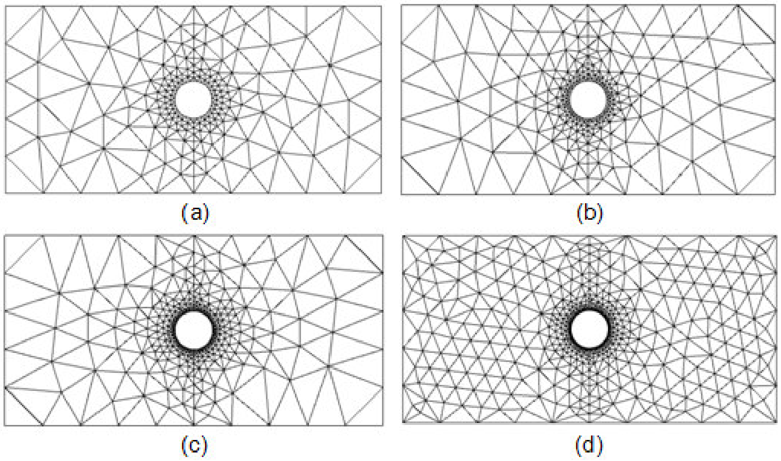


Figure 4-32. *h*-refinement using remeshing and zero-order displacement functions: (a) 215-node mesh (b) 264-node mesh (c) 430-node mesh (d) 638-node mesh

Indicatively, contours of displacement along the horizontal axis and major principal stress are illustrated in Figure 4-33 and Figure 4-34 respectively for the original mesh (Figure 4-31).

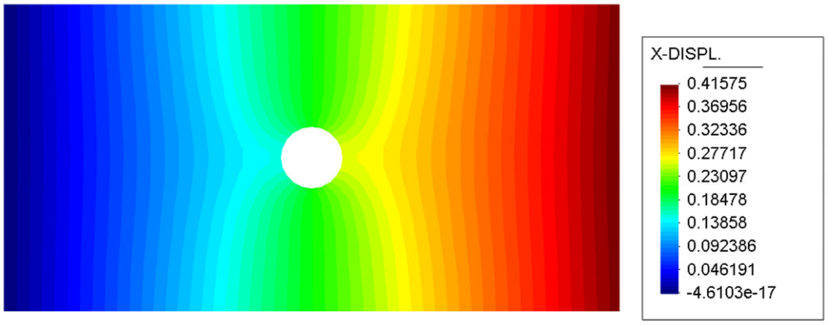


Figure 4-33. Contours of displacement along the x-axis; zero-order/constant-strain analysis using the original mesh

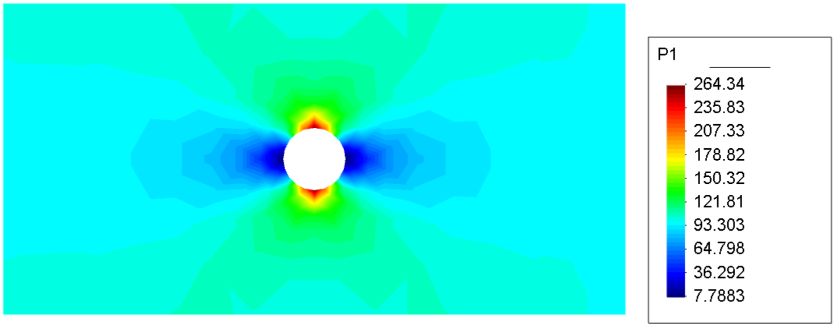


Figure 4-34. Smooth contours of major principal stress; zero-order/constant-strain analysis using the original mesh

Results of σ_{xx} at location A (top of the hole) for each of the five discretisations considered are presented in Figure 4-35 against the analytical solution. Furthermore, convergence of the energy norm error is illustrated in Figure 4-36.

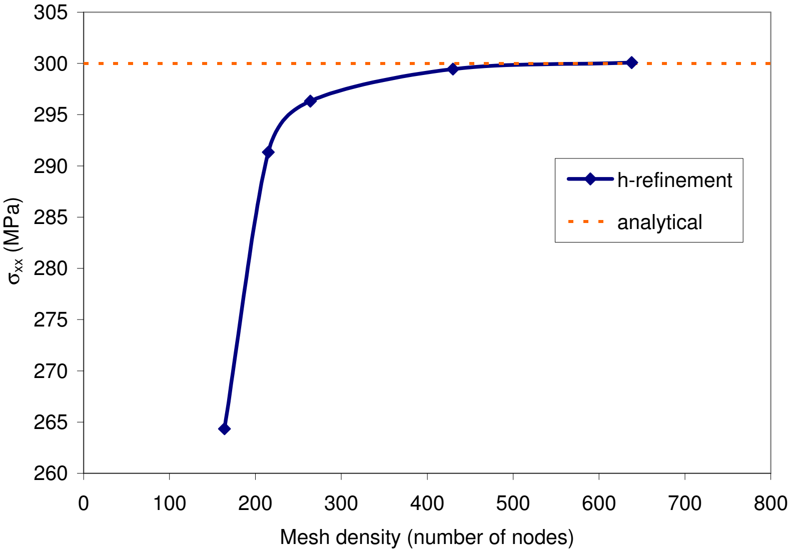


Figure 4-35. Results of σ_{xx} versus mesh density from *h*-refinement study using remeshing and zero-order displacement functions

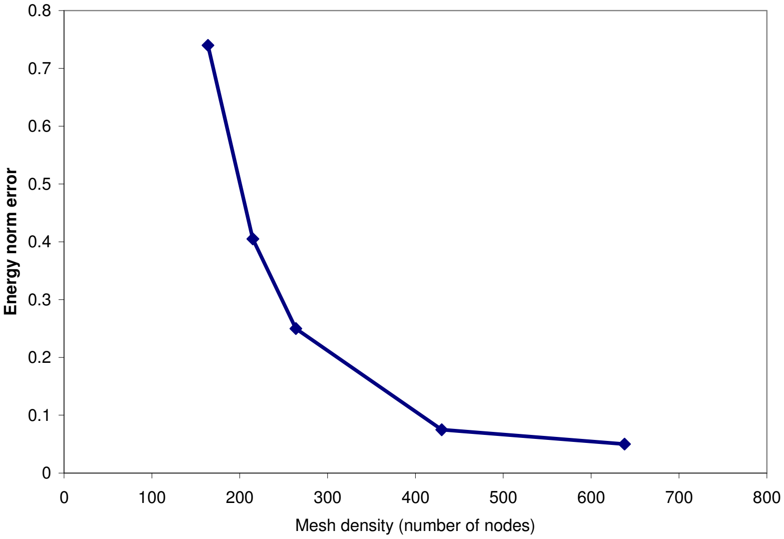


Figure 4-36. Energy norm error versus mesh density from *h*-refinement study using remeshing and zero-order displacement functions

From Figure 4-35 and Figure 4-36 it is evident that using *h*-refinement with remeshing and zero-order displacement functions, the solution eventually converges to the analytical solution with 1,276 degrees of freedom approximately.

A second experiment is conducted using only the original mesh (Figure 4-31, 164 nodes) while increasing the order of displacement functions uniformly throughout the mesh. The results are presented in Figure 4-37 and Figure 4-38 against the analytical solution.

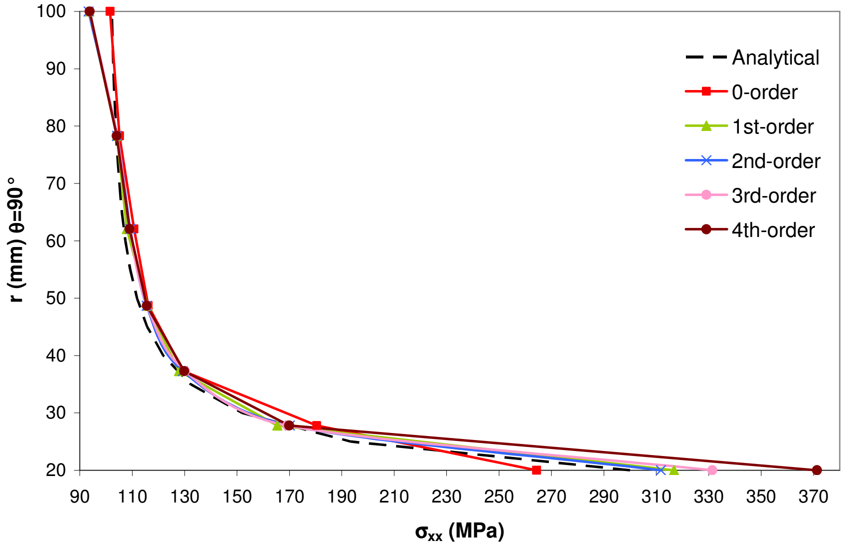


Figure 4-37. Uniform *p*-refinement: variation of σ_{xx} along the line A-A' for various orders of the displacement function polynomials

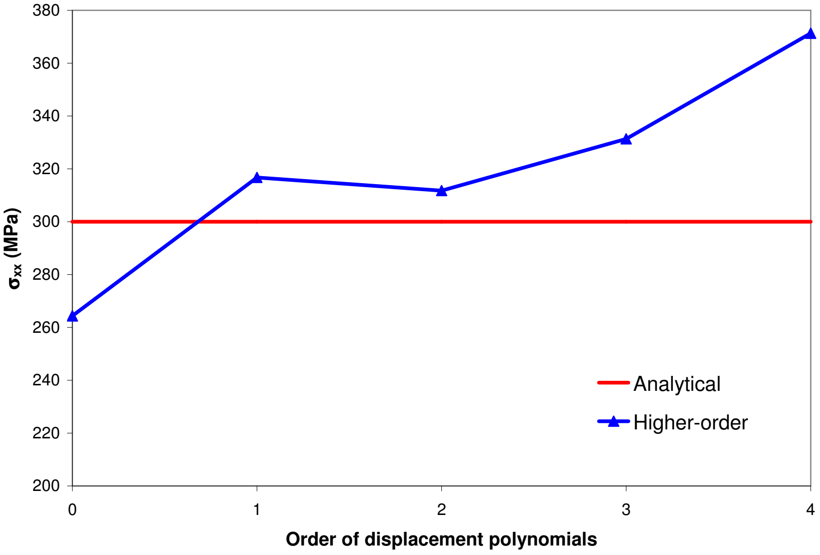


Figure 4-38. Uniform *p*-refinement: σ_{xx} at point A versus the order of displacement functions

Evidently, the zero-order solution yields a relatively poor solution in the vicinity of the hole but displays a closer match to the analytical solution towards the top boundary of the plate. On the other hand, the higher-order approximations yield better correlation to the analytical profile between the boundaries of the problem, but the solution appears to diverge at the boundaries and particularly in the vicinity of the hole (Figure 4-38). This can also be confirmed by the energy norm error for each case considered as illustrated in Figure 4-39. A potential cause for this behaviour is Runge's phenomenon (section 4.8).

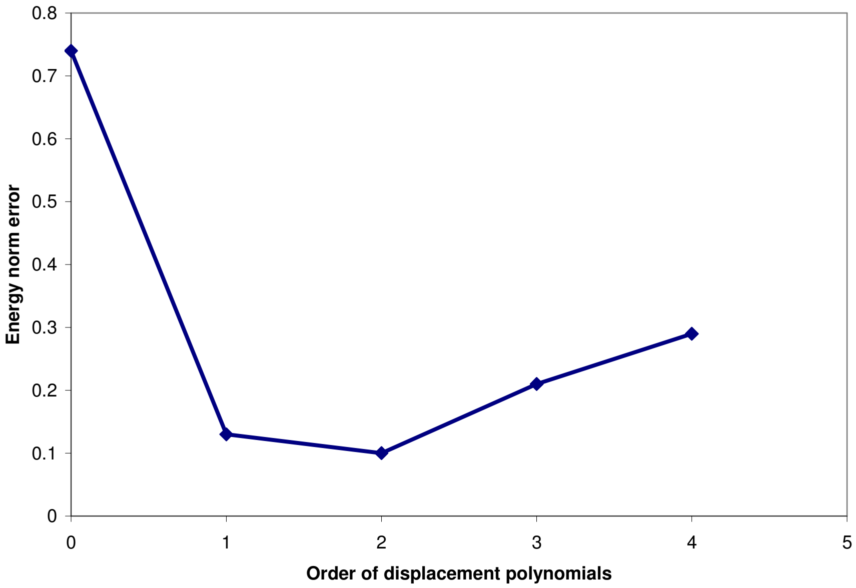


Figure 4-39. Energy norm error at A versus the order of global *p*-refinement

It is important to re-state that the same mesh was utilized in all five analyses; the only difference was the uniform increase of the order of displacement functions, resulting in 2 degrees of freedom per node in the first case and 6, 12, 20 and 30 degrees of freedom per node in the second, third, fourth and fifth case respectively (Table 4-4).

Solution	Total number of unknowns
0-order	328
1st-order	936
2nd-order	1848
3rd-order	3064
4th-order	4584

Table 4-4. Total number of unknowns per solution case

Although no remeshing is taking place, it is evident that uniform-global refinement may not yield correct results or may not be efficient in a situation such as this. However, the order of cover displacement functions can be increased alternatively for a selected number of nodes, thereby only enhancing the level of approximation locally, at areas of interest with minimal computational expense and no remeshing. If the selection of nodes and the level of *p*-refinement are chosen carefully, then convergence can be improved without remeshing and it is possible to recover the correct solution. This is demonstrated in the following example.

The same problem is now revisited considering various local enhancement strategies. Only two nodal zones in the vicinity of the hole (zones II and III as indicated in Figure 4-40) are enriched with higher-order displacement polynomials, while the rest of the domain, including the boundary defined by the hole (zone I), are enforced zero-order displacement functions at all cases.

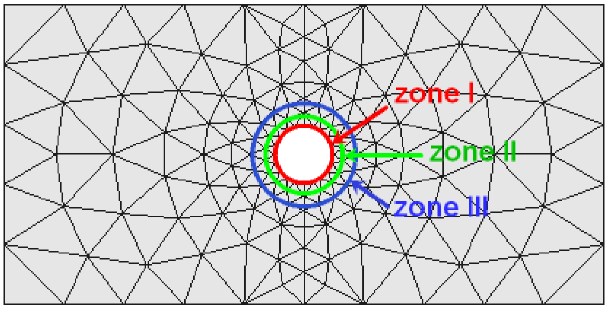


Figure 4-40. Local enhancement zones

In the first analysis, only nodal zone II is associated with first-order displacement polynomials, while the rest of the nodes are associated with zero-order polynomials. In the second analysis, nodal zones II and III are associated with second and first-order displacement polynomials respectively. Similarly, in the third analysis zones II and III are associated with third and second-order displacement polynomials respectively. A fourth analysis is undertaken in which both zones II and III are associated with third-order displacement polynomials. This is summarized in Table 4-5. The choice of displacement polynomials is also shown indicatively for the fourth analysis in Figure 4-41. The results for σ_{xx} at point A from each analysis are summarized in Figure 4-42. The energy norm error at A for each case considered is illustrated in Figure 4-43.

Analysis	Zone I	Zone II	Zone III
1	0-order	1st-order	0-order
2	0-order	2nd-order	1st-order
3	0-order	3rd-order	2nd-order
4	0-order	3rd-order	3rd-order

Table 4-5. Order of cover displacement polynomials associated with different nodal groups

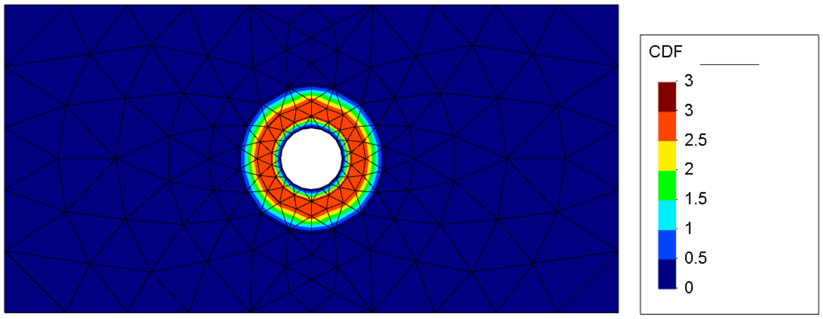


Figure 4-41. Contour plot of the distribution of higher-order polynomials in the discretised domain for the fourth case of local enhancement

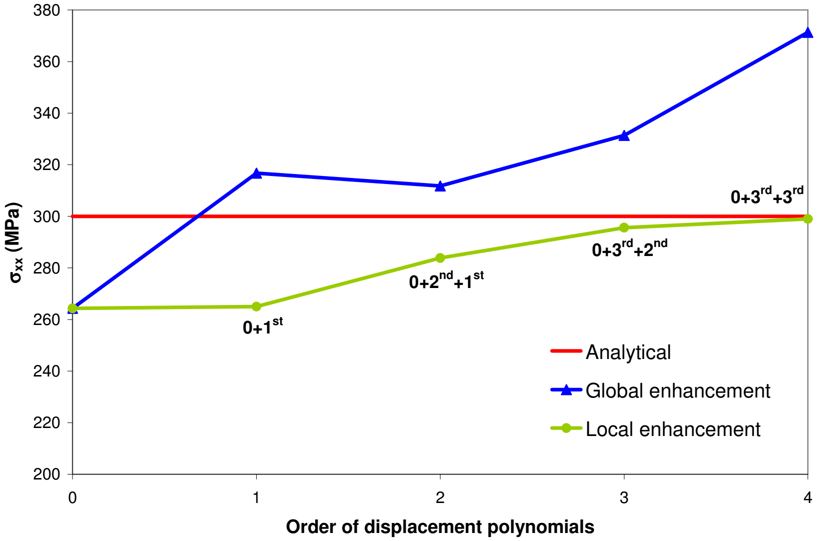


Figure 4-42. σ_{xx} at point A versus the order of displacement polynomials. Comparison between global and local enhancement.

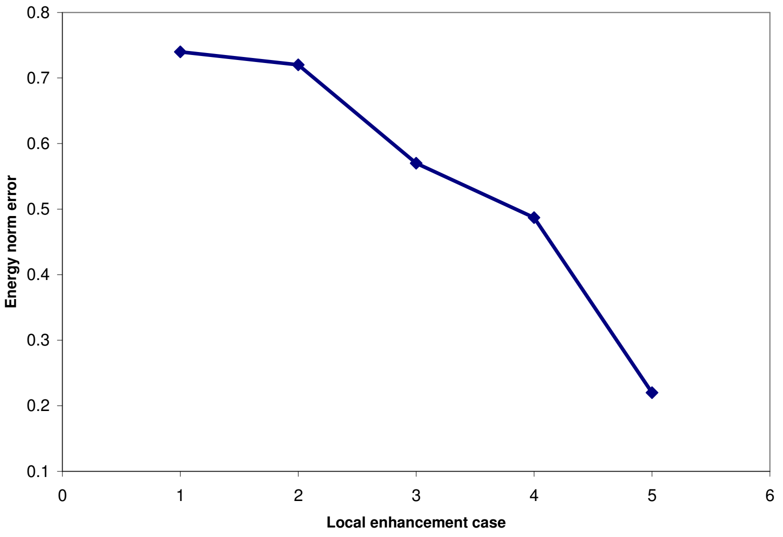


Figure 4-43. Energy norm error at A for each case of local enhancement

Solution	Total number of unknowns
1 (0+1st-order)	400
2 (0+2nd+1st-order)	588
3 (0+3rd+2nd-order)	852
4 (0+3rd +3rd-order)	1012

Table 4-6. Total number of unknowns per local enhancement case

Evidently, the use of local enhancement can achieve some control of convergence. In this case, the correct solution was achieved with a gain in computational cost when compared to the case with *h*-refinement (1012 versus 1276 degrees of freedom) and a significant gain in pre-processing effort (a single mesh was used with *p*-refinement instead of five in the case of *h*-refinement). However, it is clear that particular care needs to be taken with boundaries to avoid divergence issues. It is worthwhile to note that it is possible to automate the local enhancement process in an adaptive manner. This is demonstrated with the following example.

4.12 Example: Timoshenko beam and adaptive local enhancement

A cantilever beam problem is employed to illustrate adaptive local enhancement with higher-order NMM. The process is implemented as discussed in section 4.8. The beam is fixed on its left-hand end whereas a distributed load of 1 is applied on the free end. The elastic modulus and Poisson's ratio are taken as 100 and 0 respectively. The beam length is taken as 5, while the breadth and depth of the section are taken as 1.

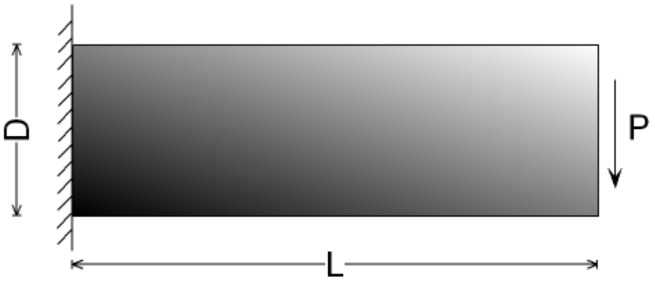


Figure 4-44. Problem definition

Two steps of adaptive enrichment are carried out using simple stress error indicators. The left and right-hand boundaries where essential boundary conditions and loads are prescribed respectively are automatically enforced with zero-order displacement polynomials to maintain continuity (Chapter 7). Essential boundary conditions are enforced using projection matrices.

The complete process is automated and can be undertaken for any order of displacement polynomials. User parameters that drive the enrichment process are stress-based norm tolerances, a maximum allowed number of enhancement steps, a maximum allowable order of displacement polynomials (to avoid excessive cost or potential divergence issues associated with the presence of singularities or point loads), and a prescribed magnitude of enhancement per step, which represents the increase of the order of displacement polynomials.

A relatively coarse structured mesh is adopted. The mesh comprises only 40 three-node triangles (Figure 4-45). This discretisation remains unaltered during the solution process.

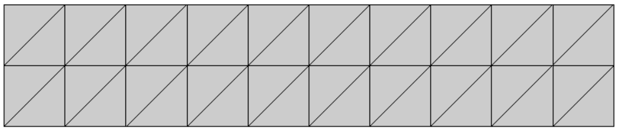


Figure 4-45. Problem discretisation

Analytical solutions for displacements and stresses for this problem are given by Timoshenko [113]:

$$u_x = -\frac{P}{6EI}\left(y-\frac{D}{2}\right)\left[\left(6L-3x\right)x+\left(2+\nu\right)\left(y^2-Dy\right)\right] \tag{4.21}$$

$$u_y = \frac{P}{6EI}\left[3\nu\left(y-\frac{D}{2}\right)^2\left(L-x\right)+\frac{1}{4}\left(4+5\nu\right)D^2x+\left(3L-x\right)x^2\right] \tag{4.22}$$

$$\sigma_{xx} = -\frac{P}{I}(L-x)\left(y-\frac{D}{2}\right) \tag{4.23}$$

$$\sigma_{yy} = 0 \tag{4.24}$$

$$\sigma_{xy} = -\frac{P\,y}{2I}(y-D) \tag{4.25}$$

Based on the above closed-form relationships, the target solution at the tip is calculated as:

$$u_{y.\max} = -5.1 \tag{4.26}$$

$$\sigma_{xx.top\ fixed\ end} = 30 \tag{4.27}$$

Results of the adaptive solution are illustrated below for each step of the process. Note that step 1 is the initial solution using FE constant-strain triangles / 0-order simplex NMM elements. Steps 2 and 3 are the two adaptivity steps.

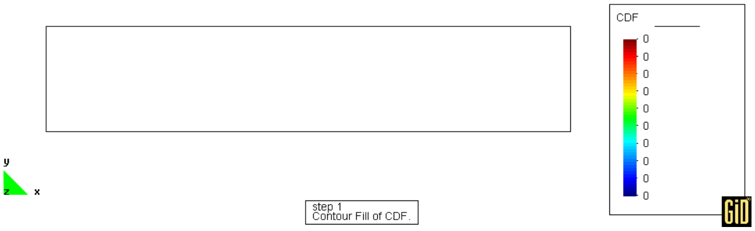


Figure 4-46. Step 1 (initial solution): contours of the order of displacement polynomials

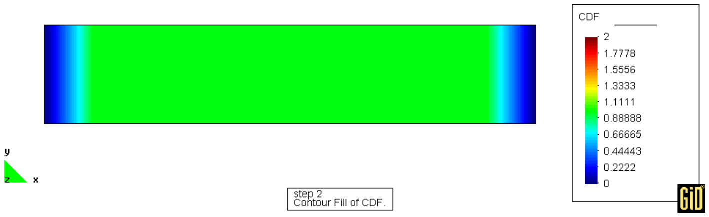


Figure 4-47. Step 2 (first adaptivity step): contours of the order of displacement polynomials

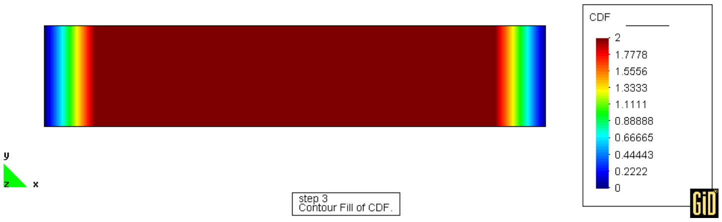


Figure 4-48. Step 3 (second adaptivity step): contours of the order of displacement polynomials

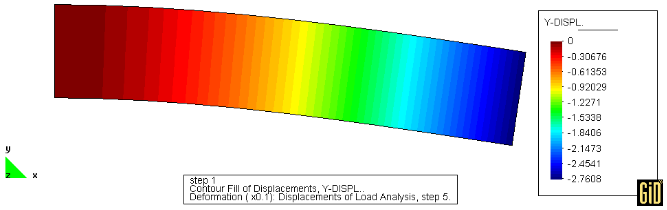


Figure 4-49. Step 1 (initial solution): deformed shape and contours of displacement in the vertical axis

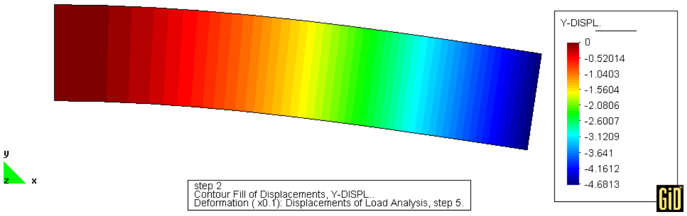


Figure 4-50. Step 2 (first adaptivity step): deformed shape and contours of displacement in the vertical axis

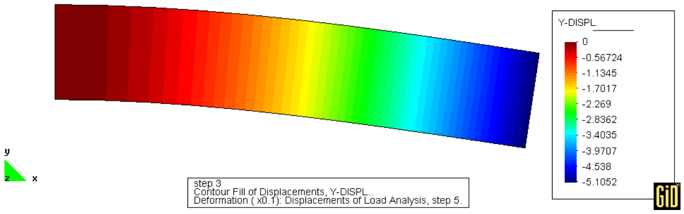


Figure 4-51. Step 3 (second adaptivity step): deformed shape and contours of displacement in the vertical axis

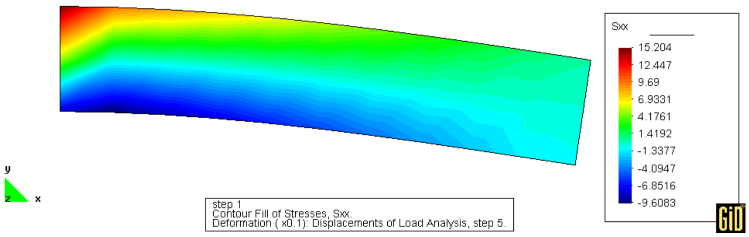


Figure 4-52. Step 1 (initial solution): deformed shape and smooth contours of σ_{xx}

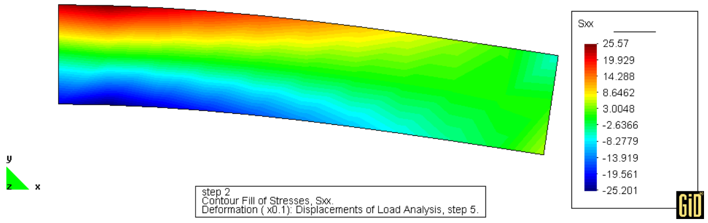


Figure 4-53. Step 2 (first adaptivity step): deformed shape and smooth contours of σ_{xx}

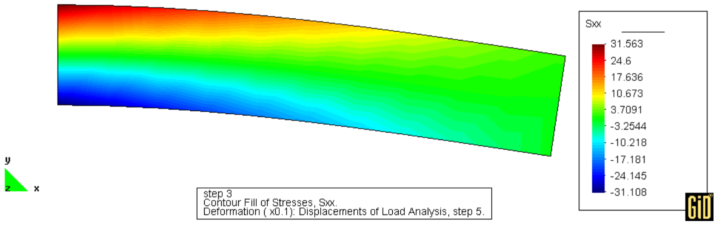


Figure 4-54. Step 3 (second adaptivity step): deformed shape and smooth contours of σ_{xx}

Step	dofs	$u_{y.\max}$	$\sigma_{xx.\max}$
1	66	-2.76	15.2
2	174	-4.67	23.4
3	336	-5.1	31.5

Table 4-7. Summary of results per step of the adaptive simulation

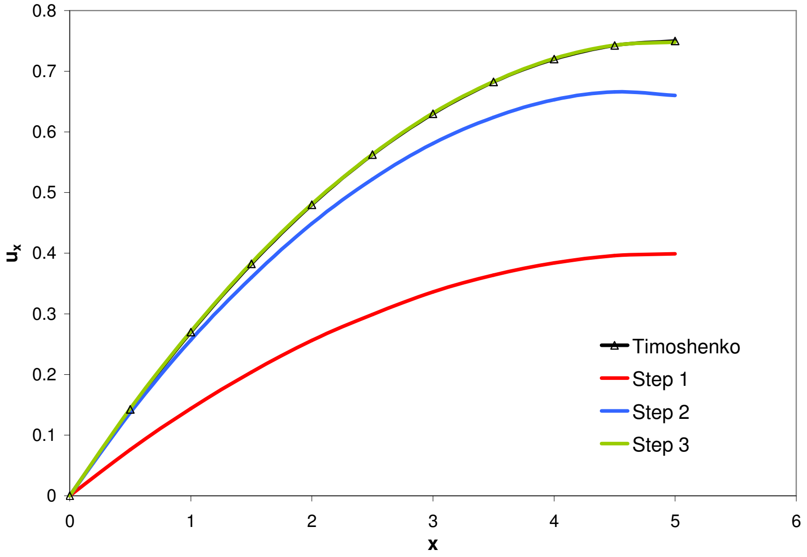


Figure 4-55. u_x versus position for each adaptive step

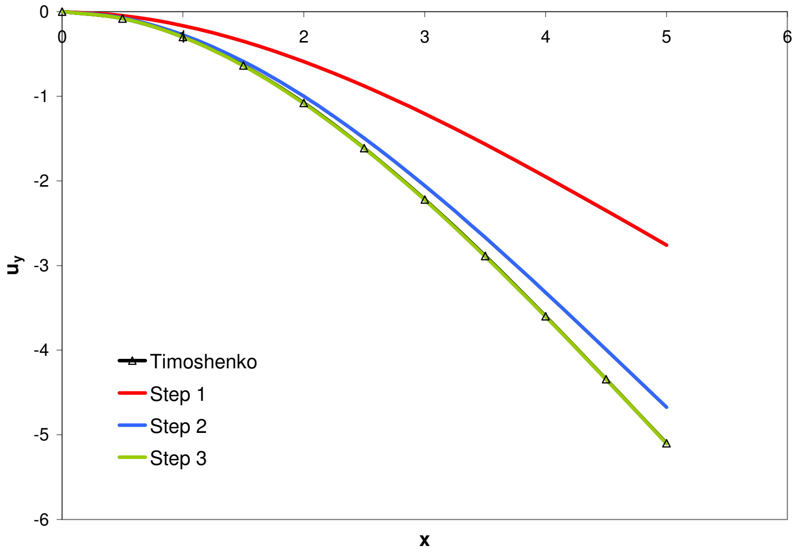


Figure 4-56. u_y versus position for each adaptive step

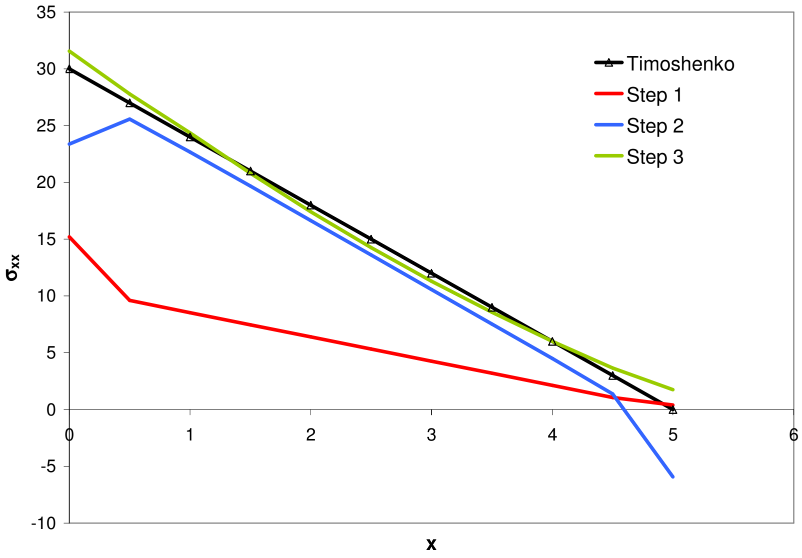


Figure 4-57. σ_{xx} versus position for each adaptive step

From Figure 4-47 and Figure 4-48 it is evident that during the adaptive process the order of polynomial displacement functions is increased uniformly throughout the domain (the left and right hand boundaries are enforced with zero-order functions at all times as discussed above). This would be expected in this particular case due to the coarse discretisation adopted and the symmetric nature of the problem (high tensile and compressive stress gradients at the top and bottom part of the beam). The first enrichment step (step 2) increases displacement polynomials to second-order (Figure 4-47), while the second enrichment step (step 3) increases polynomials to third-order (Figure 4-48).

From Figure 4-55 and Figure 4-56 it is evident that already from the first enrichment step (step 2) the approximation achieves good correlation to the closed-form solutions, whereas the second enrichment step (step 3) achieves even better match to the closed-form displacement profiles. The solution for displacement in the vertical direction matches the target solution of -5.1 exactly. The energy norm error and relative energy norm error are presented in Figure 4-58 and Figure 4-59 respectively, as calculated at the upper left element where the peak tensile stress along the axis of the beam occurs.

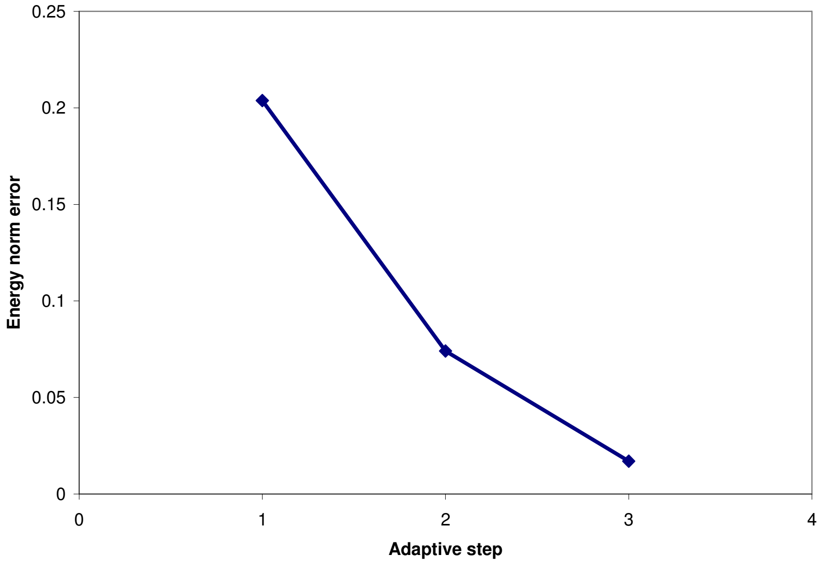


Figure 4-58. Energy norm error at the uppermost left element for each adaptive step

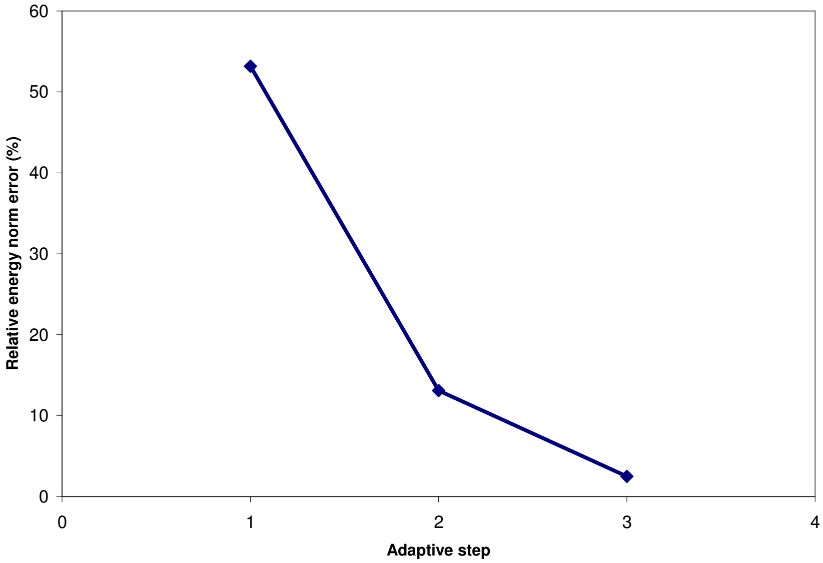


Figure 4-59. Relative energy norm error at the uppermost left element for each adaptive step

The approximated stress profile (Figure 4-57) of the first step (zero-order/constant-strain finite element case) indicates poor correlation to the analytical solution. Furthermore, it can be observed that there is an abrupt change of stress between the first two nodes at the top left of the beam, since the discretisation is too coarse. A similar effect is also observed in the first enhanced solution (step 2). In this particular case the solution also appears to oscillate at the right boundary where stress along the horizontal axis is expected to be zero, although the boundary is enforced with zero-order displacement functions.

However, the first adaptive step improves the solution significantly elsewhere comparing to the uniform zero-order case (step 1). At the top of the fixed end in particular, the relative energy error reduces from 53% in the uniform constant-strain case to 13% (Figure 4-59). The solution of the second step (step 3) achieves much better correlation to the analytical stress solution, with only 2.5% relative energy error and approximately 5% error from the target closed-form solution of σ_{xx}.

4.13 Remarks regarding NMM and hierarchical FEM

Finite element methods with hierarchical shape functions [133], are conceptually similar with the idea of higher-order approximation in NMM. In both techniques, rather than introducing more nodes to construct elements with 'standard' shape functions, the approximating displacement function is expressed as a hierarchically increasing expansion of functions which are typically polynomials derived from the Pascal-like triangle of Figure 3-8.

From this concept, it also follows in both methods that the complete problem domain may be enhanced globally, or locally by increasing the order of displacement functions rather than undertaking any kind of mesh regeneration. Potential divergence points between NMM and hierarchical FEM can be related to the fact that NMM shape functions are traditionally expressed directly in the global coordinate system, and integration is undertaken explicitly using the simplex strategy discussed in Chapter 7.

It is also worthwhile to note that, as shown by Taylor [110], a form of *p*-refinement can be realized in finite elements based on the partition of unity (in essence XFEM) by incorporating higher-order polynomials into the finite-element approximation. This concept was also applied by Wells [127] to overcome volumetric locking during plastic flow.

Furthermore, in the context of XFEM and the introduction of discontinuities via partition of unity concepts, Moës [73] discussed the extension to higher-order elements and Mariani [69] employed cubic polynomial bases to reproduce the cusp-like shape of the process zone at the discontinuity tip of a cohesive crack. However, to the author's knowledge higher-order enhancement in XFEM has not been demonstrated in the generalised form presented here.

4.14 Concluding remarks

This chapter discussed and demonstrated strategies with which global and local enhancement of the approximation field may be carried out up to any arbitrary level, with no remeshing and potentially minimal computational expense. Furthermore, the concept of local hierarchical enhancement was demonstrated in the context of a *p*-adaptive strategy.

The local enhancement strategies presented here preserve the abilities to introduce discontinuities using the methodology discussed in Chapter 5 (the partition of unity remains unaffected), and to undertake integration explicitly using simplex integration (Chapter 6). These developments, coupled with the ability to introduce strong discontinuities explicitly, without remeshing, can be potentially tailored for complex problems associated with challenging remeshing issues. For example, the improvement of approximation using structured hexahedral meshes, the simulation of propagating cracks and the transition from continua to discontinua in general can be covered by the modelling framework developed here.

Similar to other techniques, there are particular cases in which convergence is not possible or cases in which adaptivity is bound to fail; for example this can happen at singularities or regions where point loads are applied. Furthermore, experiments indicate that special considerations are required in order to maintain convergence and continuity at boundaries where higher-order polynomials are employed. The latter issue is treated in Chapter 7 whereas the former can be potentially controlled using local enhancement as it was shown earlier. Consequently, the approximation process with hierarchically increasing displacement functions requires a special level of attention and engineering judgement in order to attain meaningful results.

The process of enhancement with hierarchically increasing displacement functions is inherently associated with a progressively increasing number of degrees of freedom. Although this is achieved with a minimisation of data preparation during pre-processing or when the approximation adapts (as no mesh regeneration is undertaken), the process may not be automatically associated with a reduction in computational effort comparing, for example, to *h*-refinement with remeshing. This is due to the progressively increasing cost of the formulation process, as element deformability increases in complexity.

Furthermore, depending on the initial level of discretisation, an optimally converged solution may not be necessarily one which is achieved using *p*-refinement alone, but a combination of *h*- and *p*-refinement, so that additional standard and higher-order degrees of freedom are situated closer to regions of interest. Finally, it is worthwhile to note that aspects of the higher-order strategies and issues discussed here can be potentially applicable in other techniques which are conceptually similar to NMM, such as DDA and extended finite elements.

5 Modelling of evolving displacement discontinuities

5.1 *Introduction*

Previously, it was discussed that quasi-brittle failure has been traditionally approached numerically using either continuum-based failure techniques (such as smeared crack concepts) or via the introduction of displacement discontinuities using discrete modelling techniques (Chapter 2).

The former provides a realistic description of strain localisation but does not fully resolve the complete failure mechanism of quasi-brittle materials and can result in severe numerical difficulties. The latter provides a better description of macroscopic traction-free cracks, but it traditionally involves *a priori* assumptions in discrete methods with regard to the location of discontinuous boundaries, or it is incorporated in standard finite elements via interface elements and computationally involved remeshing techniques.

The unification of continuous and discrete modelling techniques appears as a more natural and attractive progression for simulating localisation and failure, and this is what meshless and partition of unity based techniques have recently attempted to resolve. This chapter examines how the partition of unity property of NMM weight functions can be utilised to introduce arbitrary displacement discontinuities without the requirement for *a priori* assumptions, without the use of interface elements and without remeshing.

The basic idea behind the technique developed here is the incorporation of discontinuous shape functions into the approximation, without in essence augmenting the standard or higher-order basis functions, or the solution process. Hence, discontinuities are not limited to element boundaries but can be located anywhere in the mesh. Enrichment with additional degrees of freedom that represent the displacement jumps follow directly from the existing mesh topology.

Furthermore, the following aspects of discontinuous modelling are discussed in the context of NMM:

- How discontinuities can be initiated
- Where discontinuities are initiated
- What is the orientation of introduced discontinuities
- Where discontinuities stop
- How discontinuities can potentially interact

5.2 *Partition of unity*

In Chapter 3 it was discussed that NMM weight functions are partitions of unity, and therefore must satisfy and adhere to the following conditions:

$$\sum_{i=1}^{n} w_i(\mathbf{x}) = 1 \quad \forall\, \mathbf{x} \in \Omega^e \tag{5.1}$$

$$0 \le w_i(\mathbf{x}) \le 1 \quad \forall\, \mathbf{x} \in \Omega_i \tag{5.2}$$

$$w_i(\mathbf{x}) = 0 \quad \forall\, \mathbf{x} \notin \Omega_i \tag{5.3}$$

If an arbitrary discontinuity, which is represented by boundary Γ_d (Figure 5-1), is introduced anywhere in the arbitrary physical domain Ω, then the domain on either side of the discontinuity will be partitioned, or decomposed, into Ω^+ and Ω^- respectively, where $\Omega^+ + \Omega^- \subseteq \Omega$.

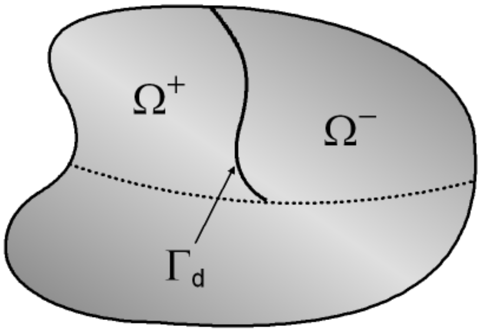


Figure 5-1. Example arbitrary physical domain Ω intersected by discontinuity Γ_d

At this stage, the definition of Ω^+ and Ω^- is arbitrary in order to introduce the concept; later it will be shown that whether a point lies to the 'left' or 'right' of a discontinuity can be determined by the location of that point and the geometrical definition of the discontinuity.

For (5.3) to hold, any weight functions intersected by the discontinuity will have to be modified in order to become discontinuous (or ineffective) in Γ_d. Specifically, weight functions in Ω^+ intersected by Γ_d have to become ineffective in Ω^-, and weight functions in Ω^- intersected by Γ_d have to become ineffective in Ω^+.

However, if weight functions intersected by the discontinuity become modified due to (5.3), equations (5.1) and (5.2) are invalidated. Therefore, in order to restore the partition of unity, additional supports (covers) may be introduced. The additional supports can either be located on Γ_d (hence remeshing is undertaken), or they can overlap existing supports on either side of Γ_d (Figure 5-2).

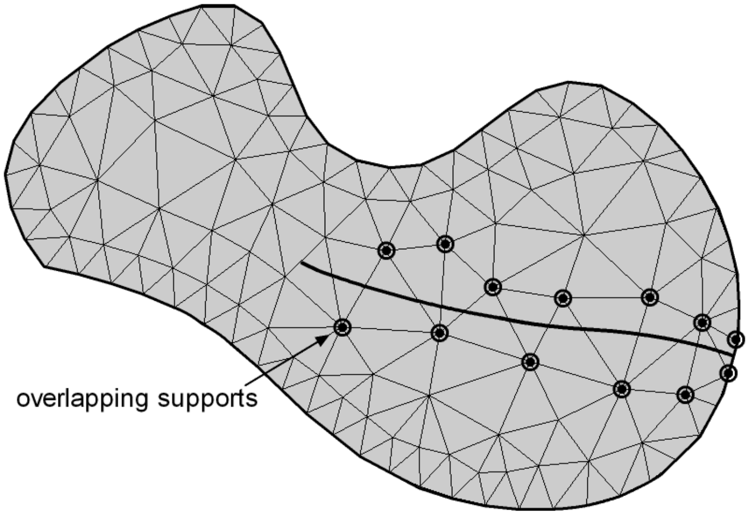


Figure 5-2. Arbitrary discretised domain with discontinuity and additional supports

In the latter situation, remeshing is not undertaken in the traditional sense, although additional degrees of freedom are introduced. In this case, the computational effort relies only on:

1. Tracking the discontinuous boundary
2. Identifying existing supports on either side of the discontinuity
3. Introducing additional supports on known locations and updating the element topology.

The concept described above can be illustrated in stages using the following simple one-dimensional single-element example defined by nodes i and j.

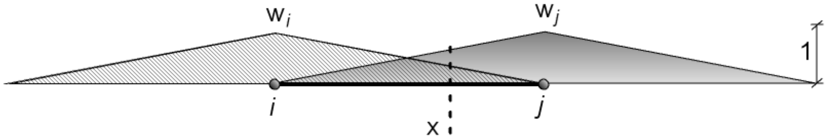


Figure 5-3. Phase 1: A discontinuity is introduced between nodes i and j of one-dimensional element ij.

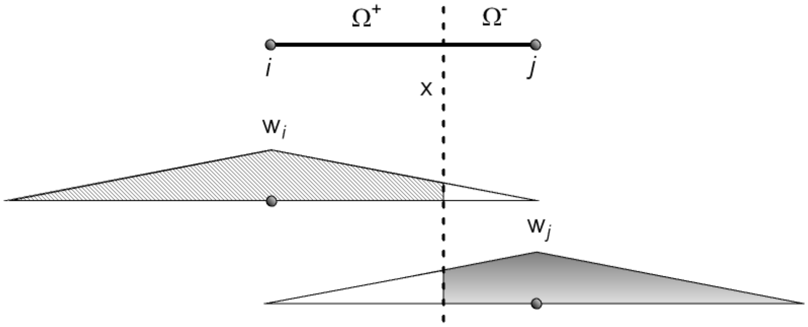


Figure 5-4. Phase 2: Weight functions become discontinuous. The partition of unity is violated between i and j .

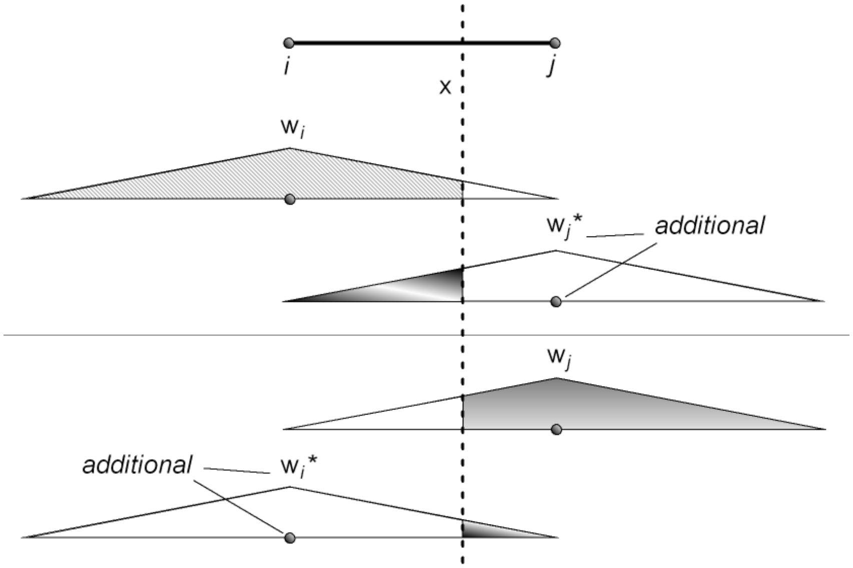


Figure 5-5. Phase 3: Identification of weight functions required to restore the partition of unity. Additional overlapping supports are introduced.

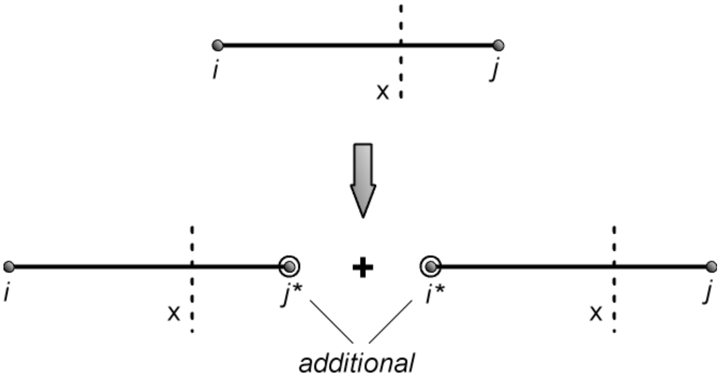


Figure 5-6. Phase 4: Connectivity of overlapping mesh. Partition of unity is restored.

Algorithmically, the partition of unity procedure described above can be significantly more simple and efficient than remeshing, since additional degrees of freedom that represent displacement jumps follow directly from the existing mesh topology.

Furthermore, the merits of the partition of unity approach can extend further in three-dimensional domains and elements with complicated shapes where remeshing algorithms become more involved, or under certain constraints they are not even possible. For example, it is not possible to remesh a structured hexahedral mesh intersected by arbitrary discontinuities unless intersected regions of the domain are discretised with different element types in an unstructured manner. In contrary, the approach described here is:

- Independent of element shape
- Independent of mesh structure

It is worthwhile to note that, since the procedure described up to this point relies on the partition of unity property only, it can be implemented in any technique that incorporates weight or shape functions that satisfy the partition of unity.

5.3 Kinematics of discontinuities

In this section, the introduction of discontinuities via the partition of unity concept is taken a step further by examining the displacement field in the vicinity of discontinuities.

Previously, it was discussed that where the cover (or support) of a node is completely intersected by a discontinuity, that node is duplicated by an additional node that is unconnected to its parent. For example, consider the plane element in Figure 5-7. The covers of nodes 1 and 2 are deemed to be cut completely by a discontinuity. These nodes are highlighted by circles.

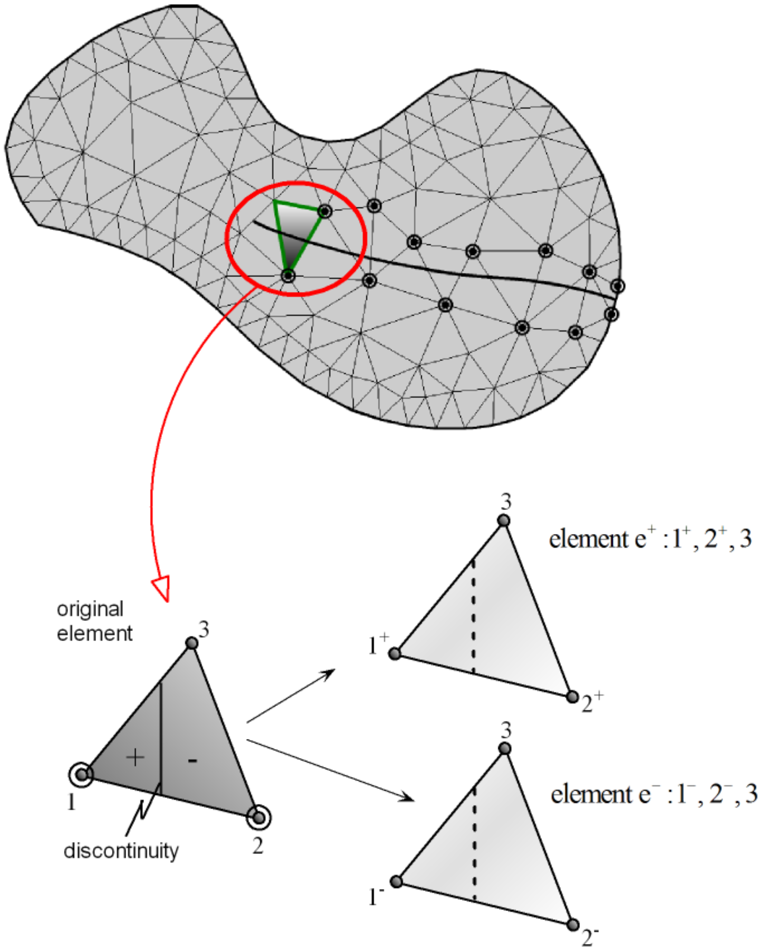

original
element
3
+
-
1
2
discontinuity
3
element e^{+} : 1^{+}, 2^{+}, 3
1^{+}
2^{+}
3
element e^{-} : 1^{-}, 2^{-}, 3
1^{-}
2^{-}

Figure 5-7. Plane element intersected by discontinuity

Node 1 lies to the 'left' of the discontinuity and thus it is renamed 1^{+} and it is duplicated by an additional node 1^{-}. Node 2 lies to the 'right' of the discontinuity, and therefore it is renamed 2^{-} and it is duplicated by an additional node 2^{+}.

The displacement fields to the 'left' and to the 'right' are now described by separate displacement functions:

$$\mathbf{u}=\mathbf{u}^{+}+\mathbf{u}^{-} \tag{5.4}$$

where:

$$\mathbf{u}^{+}=w^{+}\boldsymbol{\alpha}^{+} \tag{5.5}$$

$$\mathbf{u}^{-}=w^{-}\boldsymbol{\alpha}^{-} \tag{5.6}$$

α^+ and α^- denote the cover displacement functions which, in the case that these are constant, represent two alternative sets of nodal displacements, whereas w^+ and w^- represent the standard cover weighting functions modified by the Heaviside function (H):

$$w_i^+ = \mathrm{H}\, w_i \tag{5.7}$$

$$w_i^- = (1 - H)\, w_i \tag{5.8}$$

where:

$$\mathrm{H} = \begin{cases} 1 & \mathbf{x} \in \Omega^+ \\ 0 & \mathbf{x} \in \Omega^- \end{cases} \tag{5.9}$$

Evidently, w_1^+ is equal to the original weighting function w_1 at the left of the discontinuity and equal to zero at the right of the discontinuity (Figure 5-8). Conversely, w_1^- is equal to zero on the left of the discontinuity and equal to w_1 on the right of the discontinuity (Figure 5-9).

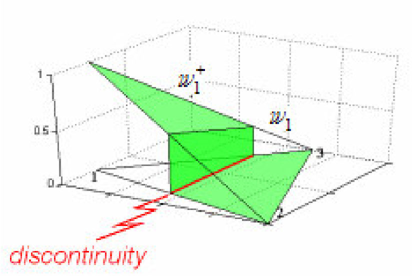


Figure 5-8. Weight function associated with node 1^+. w_1 and w_1^+ represent the original and modified weight functions respectively.

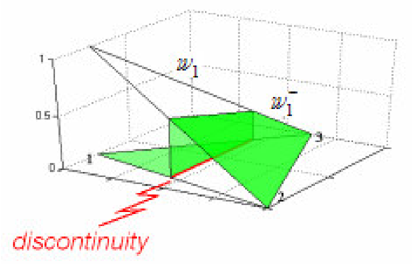


Figure 5-9. Weight function associated with node 1^-. w_1 and w_1^- represent the original and modified weight functions respectively.

The condition for the weighting functions to form a partition of unity is therefore satisfied. In the case that the cover of a node is not intersected, then:

$$w_i^+ = w_i^- = w_i \tag{5.10}$$

It is computationally convenient to consider that the two displacement fields $\mathbf{u}^+$ and $\mathbf{u}^-$ are associated with separate yet identical overlapping elements, denoted as e^+ and e^- (Figure 5-7).

It is worthwhile to note that in Figure 5-7, node 1^+ is identical to node 1 and has only been renamed for notational convenience. Thus, element e^+ consists of nodes 1 and 3 of the original mesh, and the additional node 2^+ which is unconnected to the original mesh. In essence, the displacement jump has been introduced arbitrarily into the element domain via additional degrees of freedom in the form of a strong discontinuity and the displacement field on either side of the discontinuity is decoupled.

The key difference of this approach when compared to remeshing techniques is that the additional degrees of freedom follow directly from the existing mesh topology, hence the approach is simple to implement, efficient and it does not alter the mesh structure.

5.4 Nonlinear constitutive characterization

Discontinuities can be specified *a priori* or they can be introduced when required in arbitrary locations of the domain. In finite elements, the former approach is known to lead to discrepancies associated with bias due to predefined discontinuity trajectories and it requires the use of artificial elastic stiffness in order to maintain continuity in the discontinuous zone before failure.

Simone [98] reports that in the context of partition of unity models, similar to conventional interface models, the use of predefined discontinuities can also lead to numerical discrepancies due to pathological coupling between the degrees of freedom used to resolve the potential displacement jump. It is suggested that this issue can be circumvented when the discontinuous enhancement is introduced only when inelastic strains appear [100]. In this case, it is not required to include an elastic part in the constitutive model as the elastic onset is represented by the continuum prior to the appearance of the jump.

In traditional linear elastic fracture mechanics, the nonlinearity of the process zone is lumped at the tip of a discontinuity, assuming that localization is small compared to problem dimensions. Consequently, the surrounding material can be assumed to remain elastic. For quasi-brittle and certain ductile materials, this concept is not entirely satisfactory [11, 43, 76] and the cohesive forces that exist in the process zone must be taken into account (Figure 5-10). In this case, the degrading mechanism is lumped in a discrete zone in the vicinity of the tip (Figure 5-11).

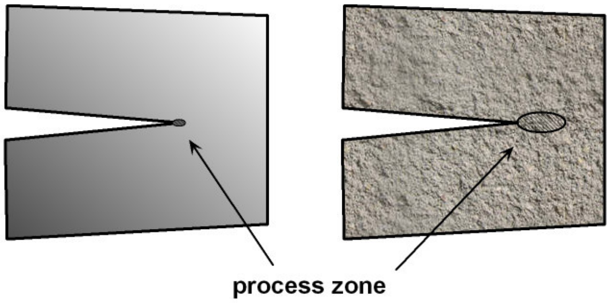


Figure 5-10. Difference between fracture where linear elastic fracture mechanics apply (left) and quasi-brittle problems (right)

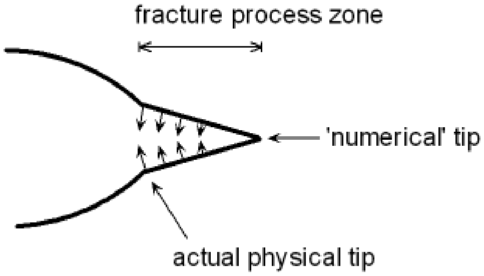


Figure 5-11. Conceptual model of a cohesive zone. Adapted from Moës [76]

Moës and Belytschko [76] note that this distribution of cohesive forces leads to a reduction of the singular stress field observed at the tip in linear elastic fracture mechanics models. Therefore, it is not necessary to use special singular enrichments in the approximation field around the discontinuity tip as it is traditionally conducted in fracture mechanics. For example, in the context of XFEM Moës [76] used non-singular enrichments motivated by asymptotic analysis, while Mariani [69] used quadratic polynomial enrichments.

In this case, since nonlinear behaviour is concentrated in a discrete line or surface, the behaviour of the displacement jump in the vicinity of the tip can be defined directly in terms of traction and displacement. As a result, discrete constitutive models are typically based on explicit traction-separation formulations [117, 125] so that tractions or stresses are computed directly from total relative displacements:

$$\mathbf{t}_d = \mathbf{T}_C \, \mathbf{u}_d \tag{5.11}$$

where $\mathbf{t}_d$ is the traction vector with components in the principal directions of the discontinuity, $\mathbf{T}_C$ is the constitutive tangent matrix and $\mathbf{u}_d$ is the vector of the relative displacement jump in the principal directions of the discontinuity.

Discrete constitutive relationships can also be formulated from classical continuum models by taking into account the Dirac-delta distribution in the strain field [125]. For example, Lourenco [63] developed a plasticity-based composite yield criterion which combines cracking and crushing, with non-associated Coulomb friction for the shear domain, with a view to modelling the behaviour of masonry joints.

Here, a direct damage-based exponential softening model is provided as an example. This model is derived from Wells [125] and implemented in the example given in section 5.13. Indicative units are given in the metric system. Nonlinear response is governed by the tensile strength f_t (N/m²) and the fracture energy release rate G_f (N/m) (Figure 5-12) of the material.

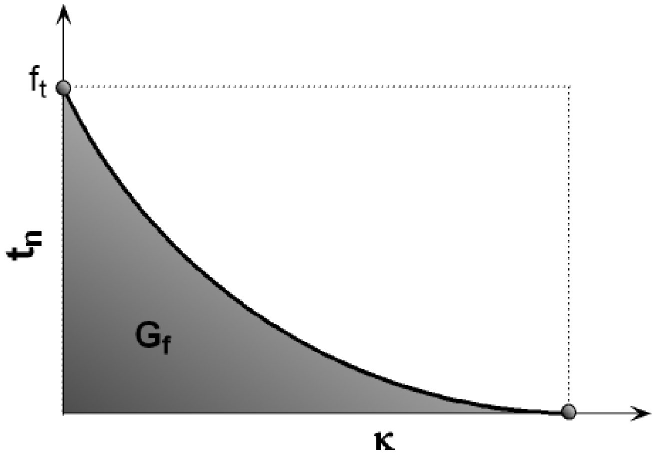


Figure 5-12. Softening behaviour of discrete damage-based model

Assuming that the discrete jump is introduced only when the initiation criterion is met, the elastic regime is resolved by the continuum constitutive representation prior to the jump. Development of the nonlinear onset is described by a loading function *f* (m), which defines the state of loading with respect to the evolution of the displacement jump:

$$f = u^{eq} - \kappa \tag{5.12}$$

where u^{eq} (m) is a scalar measure of the displacement jump at a discontinuity and κ (m) is a history parameter which is equal to the maximum value reached by u^{eq}. For example, while a discontinuity grows κ is equal to u^{eq}, so f equals zero and loading is taking place. If on the other hand f is less than zero, unloading or reloading is taking place.

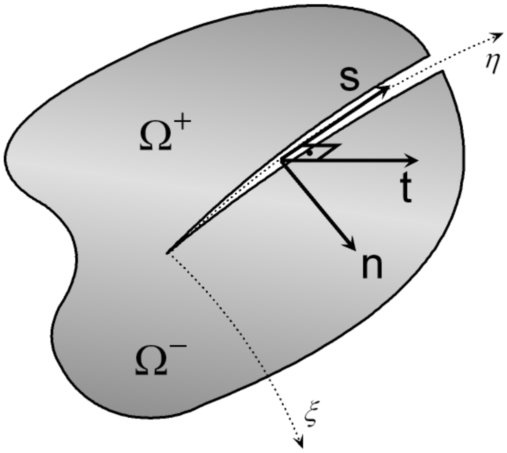


Figure 5-13. Local coordinate system of discontinuity

The normal traction component (Figure 5-13) of a discontinuity is defined as:

$$t_n = f_t \exp\left(-\frac{f_t}{G_f}\kappa\right) \tag{5.13}$$

while the shear components in the directions t and s parallel to the discontinuity plane (Figure 5-13) are given by:

$$\begin{bmatrix} t_s \\ t_t \end{bmatrix} = d_{ini} \exp\left(h_s \kappa\right) \begin{bmatrix} u_s \\ u_t \end{bmatrix} \tag{5.14}$$

where $t_{s,t}$ (N/m^2) and $u_{s,t}$ (m) are the tractions and displacements respectively in the shear directions, d_{ini} (N/m^3) is the shear stiffness at $\kappa = 0$ (initial stiffness) and h_s is a parameter which defines the decay of shear stiffness while the discontinuity evolves in the normal direction. h_s is typically calculated as:

$$h_s = \ln\left(\frac{d_{\kappa=1}}{d_{ini}}\right) \tag{5.15}$$

5.5 Initiation of discontinuities

Initiation of discontinuities in quasi-brittle materials can be instigated using principal stress-based criteria. Principal stresses can be determined from the eigenvalues of the Cauchy stress tensor. If the problem is initially continuous, then the location of the initial discontinuity tip can be determined by sampling all stress points. If a discontinuity already exists, then sampling can be restricted to elements in the neighbourhood of the discontinuity tip. However, if the possibility of multiple discontinuities is considered, then it is necessary to sample all stress points at every step of the analysis.

Simone [98] notes that although mesh refinement studies suggest that the total energy dissipated during crack propagation is a constant material parameter in an elastic medium [126], hence there is merit in stress-based initiation criteria, the approach is not always satisfactory from a physical and mathematical perspective. For example, with refinement it may be possible to recover a singular stress field in the vicinity of the tip. In NMM, this issue can be exacerbated with the use of higher-order displacement functions due to oscillating interpolation errors (Chapter 4).

In such cases, criteria based on elastic principal stress are not meaningful measures of initiation, and energetic criteria based on fracture mechanics principles can be more meaningful [76]. Alternatively, a simple remedy can be the introduction of a length scale associated with the sampling radius to ensure that only stress points at a given distance away from the tip are considered. This approach can avoid the singular stress field concentrated around the tip in cases where mesh refinement or hierarchical enhancement has been carried out.

Although criteria derived using local stress tensors may be more accurate for adequately fine discretisations, they can be associated with bias if the approximation of the local stress field is ill-posed. Incorrect identification of the orientation of discontinuities can subsequently lead to incorrect solution paths.

A more robust approach is the use of a 'non-local' stress tensor to determine the principal stress directions, as suggested by Jirasek [47] in the context of embedded discontinuity models, by Simone [98] in partition of unity based discontinuous elements, and Wells [125] in the context of embedded and partition of unity based discontinuous elements. The approach is not non-local in a constitutive sense, but a more accurate way of averaging stresses in the vicinity of the tip. The stress tensor can be calculated as a weighted average of stresses ahead of the tip using Gaussian weight functions. The weight function associated with a stress point i is:

$$\mathrm{w}_i = \frac{1}{\left(2\pi\right)^{3/2} r^3} \exp\left(-\frac{l_i^2}{2r^2}\right) \tag{5.16}$$

where r is the interaction radius typically taken as three times the average element size in front of the tip, and l_i is the distance of point i from the tip.

It is worthwhile to note that if the material surrounding the discontinuity tip is nonlinear, then the stress field will be bounded and therefore principal stress-based criteria can be used with more confidence. Furthermore, the use of stress-based criteria can be appropriate for cohesive zone models, due to the inherent reduction of the singular stress field at the tip comparing to traditional linear fracture mechanics models, since nonlinearity is distributed rather than lumped at a single point [76]. In this case, the strain field around the tip can be improved by the use of non-singular enrichments or higher-order polynomial functions, as noted in the preceding section.

5.6 *Orientation of discontinuities*

Similar to initiation criteria, the orientation of discontinuities can be obtained using principal stress-based criteria. Since it is known that discrete failure takes place predominantly perpendicular to the direction of the driving principal stress, the simplest approach is to assume that discontinuities extend perpendicular to the direction of major principal stress at the initiation point and intersect elements in the same direction. Alternatively, a more accurate approximation may be achieved if it is assumed that discontinuities extend from the initiation point to the location of major principal stress interpolated using the non-local stress tensor associated with equation (5.16).

5.7 *Discontinuous enhancement and propagation*

When an element is intersected by a discontinuity, appropriate supports on either side of the discontinuity are duplicated and the topology is updated using the procedure described in section 5.2. However, in order to enforce the condition of zero displacement jump at the tip, at least one support on either side of the tip must be unique (Figure 5-14).

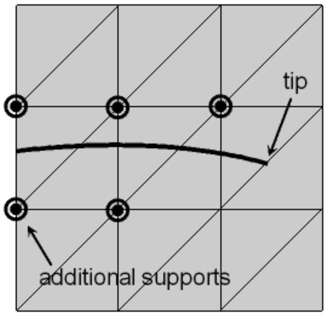


Figure 5-14. Discontinuous enhancement at the tip of a discontinuity

The decision of whether an element is intersected by a discontinuity and whether a support of an intersected element lies on the 'left' or 'right' of the discontinuity, can be determined using geometric predicates such as standard incircle and orientation tests. Incircle tests verify whether elements or supports are intersected by discontinuities whereas orientation tests verify whether supports lie on a given side of a discontinuity.

However, from an algorithmical point of view it can be convenient to only identify newly intersected elements, automatically duplicate all of their supports and subsequently remove the additional supports on either side of the tip in order to enforce the zero displacement jump condition.

Intersection can be typically identified using criteria based on the sign of the area defined by a segment of the discontinuity and the vertices of an element. For example, in two dimensions the signed area formed by two points of the discontinuity and a vertex of a simplex element is:

$$\mathrm{A} = \frac{1}{2!}\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_q & y_q & 1 \end{vmatrix} \tag{5.17}$$

where $(x_1, y_1),(x_2, y_2)$ are the coordinates of the end points of a discontinuity segment and (x_q, y_q) is a 'query' point.

If all three triangular areas formed from a discontinuity and the three vertices (used as query points) of a simplex element have the same sign, then all three vertices lie on the same side of the segment, hence the element is not intersected, provided also that all three areas are non-zero. If on the other hand the areas formed by two vertices have the same sign and the area formed by the third query point has opposite sign, then the element is intersected (Figure 5-15).

An algorithm for detecting intersection in two dimensions is given in Appendix B.

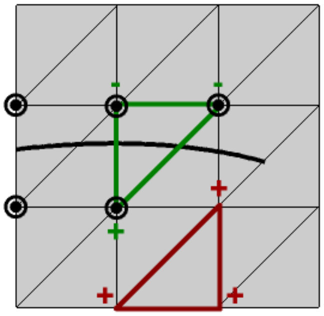


Figure 5-15. Area signs of an element intersected by a discontinuity and an element which is not intersected

In principle, discontinuities can be introduced as single or multiple segments within elements, they can intersect elements completely or they may not, and their trajectory can be straight, curved or branched. However, Wells [125] notes in the context of partition of unity models, the implementation of curved discontinuities or discontinuities that do not intersect elements completely requires the implementation of ramp functions which lead to convergence difficulties when used with incremental solution procedures.

Furthermore, the implementation of branched [31, 76] or curved discontinuities [101] involves augmentation of the standard polynomial basis functions with special enrichments. This can subsequently alter the solution process and dilute the ability to improve the approximation hierarchically for any arbitrary level (Chapter 4) and the ability to undertake integration explicitly (Chapter 5).

Thus, from an implementational point of view, the simplest approach is to extend discontinuities so that they intersect elements completely. Therefore, when the initiation criterion determines that the tip will propagate, the discontinuity is extended fully within the next element.

Evidently, for a very coarse mesh this approach may yield a degree of inaccuracy. However, for fine discretisations the discrepancies would diminish. It is worthwhile to also note that in the context of cohesive zone models, energy dissipation depends on the relative opening or slip movement of discontinuities, therefore results obtained from the complete-intersection approach are not particularly sensitive to the length of extension [125].

Furthermore, if the approximation is enhanced with higher-order displacement functions (Figure 5-16) as it will be discussed later, the deformability of enriched elements is increased and the resulting strain field is inhomogeneous. Consequently, the deformability of elements intersected by discontinuities is enhanced. This can help improve estimates derived by stress-based initiation and orientation criteria.

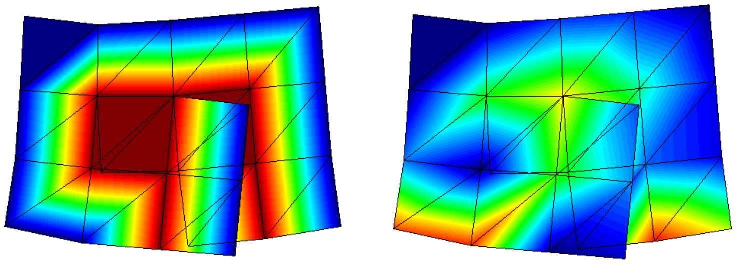


Figure 5-16. Enhanced higher-order approximation of simply supported concrete beam. Contours of higher-order functions (left) and resulting tensile principal stress contours on overlapping mesh (right).

5.8 Integration

Integration in NMM is normally undertaken explicitly using simplex integration. Simplex integration is described in detail in Chapter 6. The sole difference between integration of intact and fractured elements is that in the latter case, the integration volume (or area) is the volume to the 'left' or 'right' of the discontinuity, rather than the volume of the original element. Exact integration over arbitrary domains which result in such cases is already covered by the simplex integration strategy discussed in Chapter 6 and no additional considerations are required.

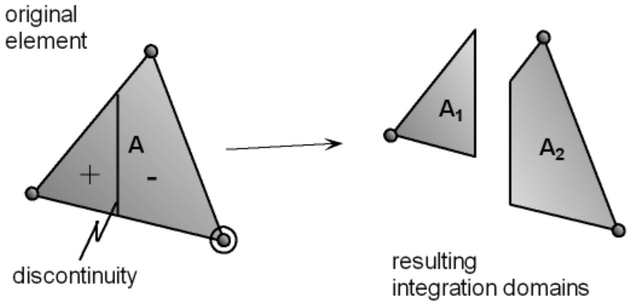


Figure 5-17. Integration domain of element intersected by discontinuity

In contrast, in partition of unity methods based on numerical integration schemes a form of domain sub-partitioning is required [98, 125]. For example, in the particular case of a triangular element intersected by a discontinuity, the two resulting domains would have to be triangulated so that the edges of each sub-domain coincide with the boundary of the discontinuity. Each sub-domain would then be mapped to a parent triangle over which integration would be carried out.

5.9 Contact modelling

Contact of discontinuous parts of the domain due to load reversal and crack closure is not a necessary consideration in many static and quasi-static problems; however it can be essential in dynamic scenarios, such as problems considering impact. General frictionless or frictional contact models developed for finite element methods [134] or DDA [102] can be equally applicable here, depending on the application considered.

Since the displacement jump is represented by degrees of freedom that are located on existing supports on either side of the discontinuity, rather than on the boundary of the discontinuity, contact criteria can be based on the relative movement of enhanced supports.

Alternatively, it may be convenient to track movement of the discontinuous boundary with virtual 'internal' points. This may be necessary in problems enriched with higher-order displacement functions due to the enhanced deformability of enriched elements (at the discontinuous boundary displacement between two points is not linear interpolation of the displacement of these two points).

Node-to-surface spring contact matrices in the context of NMM have been proposed by Lu [65].

5.10 Solution strategy

The solution of linear or nonlinear quasi-static problems requires the use of incremental procedures. Here, a standard Newton-Raphson iterative technique [134] has been implemented with force and displacement control, using a direct solver and convergence criteria based on residual force. The method exhibits quadratic convergence, provided that a good initial solution is achieved. It is worthwhile to note that if the tangent matrix is unsymmetrical, as it can be the case when considering shear cohesive components or frictional contact, unsymmetrical solvers are required. This case is not considered here. An algorithmic account of the solution process is provided in Appendix C.

After a solution has been obtained it may be necessary to crop overlapping parts of the mesh where the partition of unity is not valid, so that the discontinuity is visualised in a topologically convenient manner (Figure 5-18). This can be achieved by implementing an algorithm which considers virtual points on the boundary of the discontinuity, calculates results at these points and produces a cropped mesh for post-processing. No actual additional degrees of freedom are introduced during this process and results at the boundary of discontinuities are computed *a posteriori* using the basic definitions given in Chapter 3.

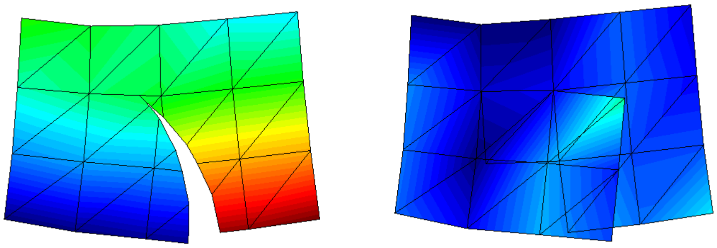


Figure 5-18. Concrete beam of section 5.12. Cropped deformed mesh (left) and tensile principal stress contours on overlapping mesh (right).

For example, displacements at the boundary of a discontinuity or indeed at any point P within a discontinuous sub-domain of an element can be computed directly from equation (3.33), with the shape function matrix evaluated at the location of P.

5.11 *Example: Beam in tension*

In order to illustrate the concept of introducing discontinuities with discontinuous weight functions, the following one-dimensional problem is chosen. The problem consists of a straight beam constrained at one end. The beam is discretised with three bar elements with translational degrees of freedom only. A discontinuity is introduced in the middle of the beam; therefore the problem is enhanced with two additional degrees of freedom which overlap existing supports in order to represent the displacement jump (Figure 5-19). A contact spring is also introduced in order to maintain continuity.

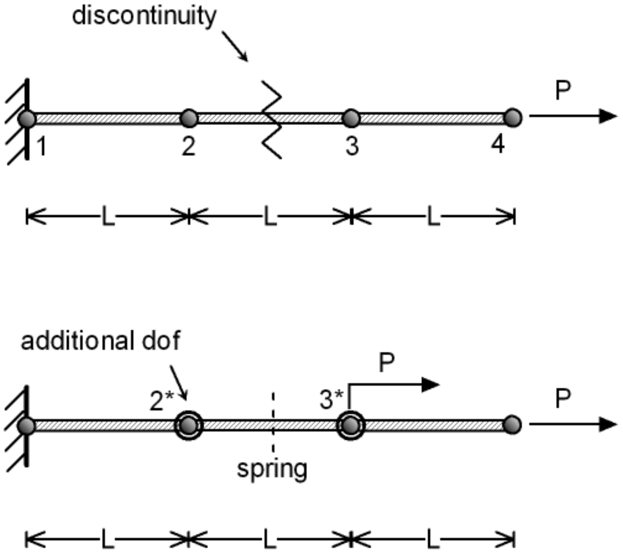


Figure 5-19. Discretisation of tension beam with discontinuity and spring

A horizontal force $P=1$ is applied on the free end of the beam. The elastic modulus is taken as $E=100$, whereas $A=1$ and $L=1$. Using Hooke's law of elasticity, a reference tip displacement can be derived for an equivalent continuous beam:

$$\left|u_4\right|_{ref} = \frac{3PL}{AE} = 0.03 \tag{5.18}$$

If the spring is relatively stiff, so that its stiffness is taken as $s_1 = \frac{EA}{L}$, NMM recovers the elastic solution:

$$\left|u_4\right|_{s_1} = 0.03 \tag{5.19}$$

The complete solution vector is:

$$\begin{aligned} &\begin{bmatrix} u_2 & u_{3*} & u_{2*} & u_3 & u_4 \end{bmatrix} = \\ &= \begin{bmatrix} 0.01 & 0.01 & 0.02 & 0.02 & 0.03 \end{bmatrix} \end{aligned} \tag{5.20}$$

The displacement at the middle of the beam, just to the left and just to the right of the discontinuity can be calculated as:

$$u_{m^-} = w_2 u_2 + w_{3*} u_{3*} \tag{5.21}$$

$$u_{m^+} = w_{2*} u_{2*} + w_3 u_3 \tag{5.22}$$

And the displacement jump can be calculated as the difference:

$$u_j = u_{m^+} - u_{m^-} \tag{5.23}$$

Since the weight functions of the supports to the left and to the right of the discontinuity equal 0.5 at the location of the discontinuity, equation (5.23) can be rewritten as:

$$u_j = 0.5\left(u_{2*} + u_3 - u_2 - u_{3*}\right) \tag{5.24}$$

Therefore,

$$u_{j,s_1} = 0.01 \tag{5.25}$$

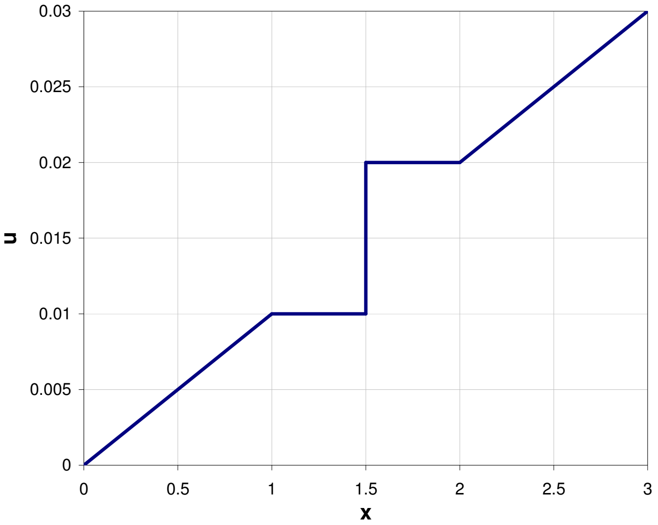


Figure 5-20. Displacement jump of 1D bar problem

The segments of constant displacement immediately to the left and right of the discontinuity in Figure 5-20 are due to the fact that spring connectivity is enforced at $x = 1$ and $x = 2$.

5.12 *Example: Simply supported beam*

An example problem presented by Jirasek [50] has been adopted here as a benchmark. Although the introduction of arbitrary displacement discontinuities in NMM has been suggested in relevant literature, it has rarely been undertaken in practice nor presented in the generic manner described here.

The problem in Figure 5-21 illustrates a concrete beam and a pre-existing crack with a curved trajectory that would be introduced into the model based on some criterion (here arbitrarily). The beam is four metres wide by three meters high. The elastic modulus is taken as 1000, while Poisson's ratio is taken as 0. The applied load consists of a vertical force of 10. Figure 5-21 illustrates not only the discontinuity but also those nodes whose cover or support has been intersected by the discontinuity. This process has been fully automated within the developed modelling framework. The problem is idealised as plane-stress.

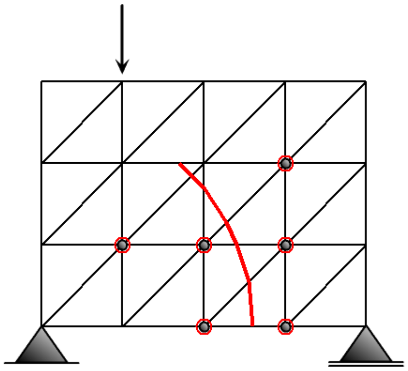


Figure 5-21. Problem set-up, mesh and location of discontinuity. The cover of highlighted nodes is cut by the discontinuity, hence those nodes are duplicated.

Figure 5-22 (a) shows how the structure would behave without a discontinuity whereas Figure 5-22 (b) displays how the structure behaves with the discontinuity included.

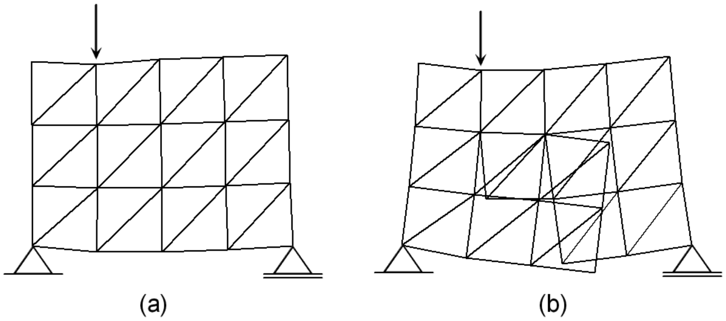


Figure 5-22. (a) Deformed mesh without fracture. (b) Deformed mesh with fracture

In Figure 5-22 (b) the duplicate elements are shown to illustrate how two separate elements are required to capture the displacement jump across the discontinuity. If those parts of each element which correspond to a zero weighting function are removed, a more realistic representation of the discontinuity is revealed in Figure 5-23. This process has also been implemented to be performed automatically within the modelling framework adopted here. The initiation of fractures has not been considered here since the process is not unique to NMM.

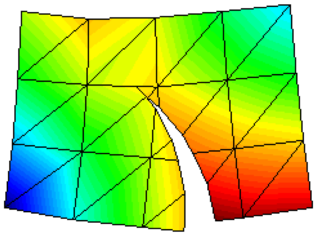


Figure 5-23. Deformed cropped mesh with fracture and displacement contours

It is worthwhile to note that the original model (without the discontinuity) consists of 40 degrees of freedom, while inclusion of the discontinuity increases the solution bandwidth by only 12 degrees of freedom. If we now produce a model with a topologically identical representation of the discrete discontinuity with element boundaries aligned with the crack using an unstructured mesh (Figure 5-24), the resulting mesh consists of 210 degrees of freedom.

The exact same boundary is used in both models and the unstructured mesh is the most-crude mesh possible with the algorithm considered. In both models zero-order elements are used.

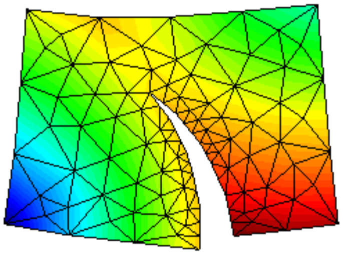


Figure 5-24. Deformed unstructured mesh with fracture aligned with element boundaries

In the partition of unity discretisation considered above the resulting maximum crack width appears much smaller comparing to the solution obtained using the unstructured aligned discontinuity model. This is credited to the coarse discretisation used in the former case and not to the way the discontinuity has been modelled. Indeed, comparison of uncracked results using similar coarse and fine discretisations yield similar discrepancies.

Using what has been discussed so far, the following observations can be made:

1. The partition of unity approach can achieve a kinematically and topologically equivalent representation of discontinuities as an aligned mesh approach, for less effort during both the pre-processing and analysis stages.
2. The partition of unity approach requires an extra step after analysis in order to produce results that can be visualised in the same way as those of an aligned mesh approach (overlapping parts of the mesh have to be cropped and results have to be calculated at the boundary of discontinuities).
3. Accuracy of the partition of unity approach is sensitive to discretisation.

In order to improve the approximation of the partition of unity model, the following general strategies could be adopted:

1. Refine the mesh, while preserve the mesh structure.
2. Refine the mesh and change the mesh structure. For example, an optimal solution that achieves the best results in terms of quality at minimum cost may be a combination of remeshing and use of discontinuous weight functions.
3. Improve the approximation locally or globally, using higher-order displacement functions.
4. Use a combination of 1 and 3, or 2 and 3

5.13 *Example: Simulation of an evolving discontinuity*

A simply supported beam is subjected to a point load as shown in Figure 5-25. The beam dimensions are 10×3×1 (span, depth, width). The point load is applied on the middle top of the beam acting downwards and it is monotonically increased in 14 steps of 0.1. The elastic modulus is E=1000, Poisson's ratio is ν=0.0, tensile strength is f_t=1 and the fracture energy release rate is G_f=0.1.

The purpose of the analysis is to predict how the beam is going to crack. The beam is initially uncracked and modelled using linear elastic material. The nonlinear behaviour of the potential crack interface is modelled using the traction-separation damage model given in section 5.4, considering only normal traction. The problem is discretised using a mesh of plane-stress constant strain triangles and solved using a Newton-Raphson iteration scheme and displacement control. No remeshing is undertaken at any stage.

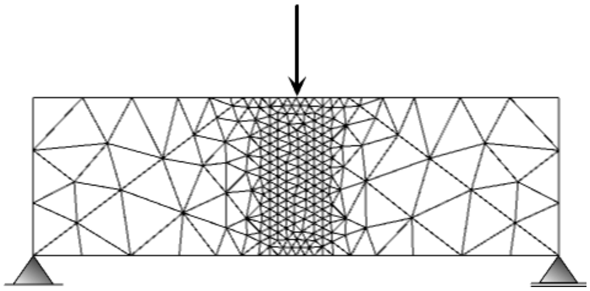


Figure 5-25. Simply supported beam

As the load is increased, a crack is formed that propagates vertically from the bottom of the beam towards the point of load application. This is illustrated in Figure 5-26 to Figure 5-28. It is worthwhile to restate that this result has been achieved without remeshing. A plot of load versus displacement is presented in Figure 5-29.

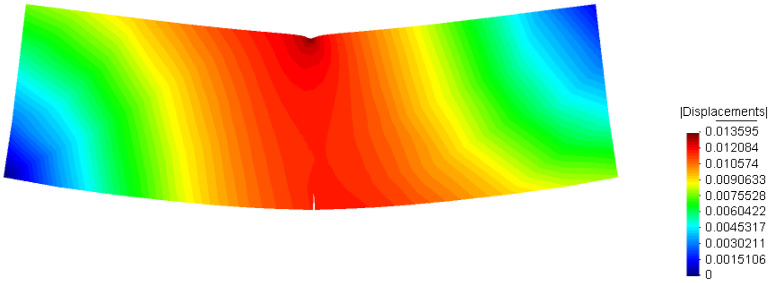


Figure 5-26. Step 4: Deformed shape and contours of resultant displacement

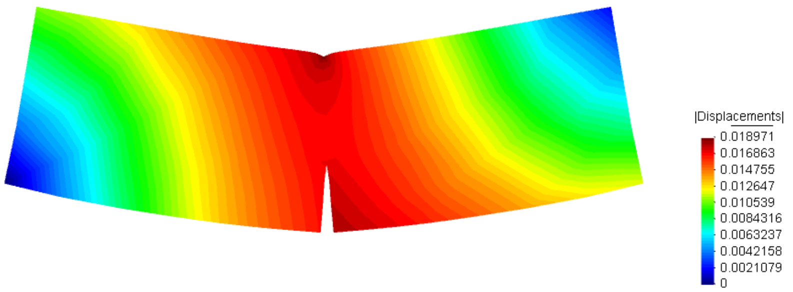


Figure 5-27. Step 9: Deformed shape and contours of resultant displacement

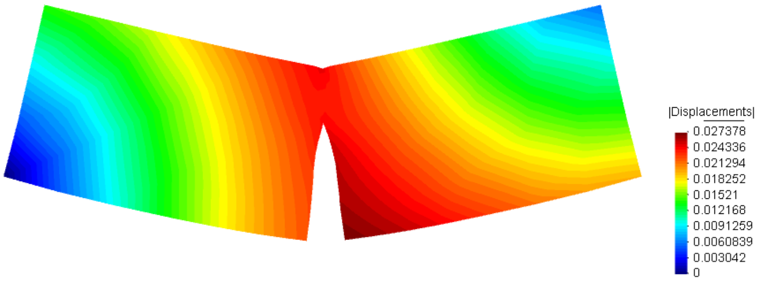


Figure 5-28. Step 14: Deformed shape and contours of resultant displacement

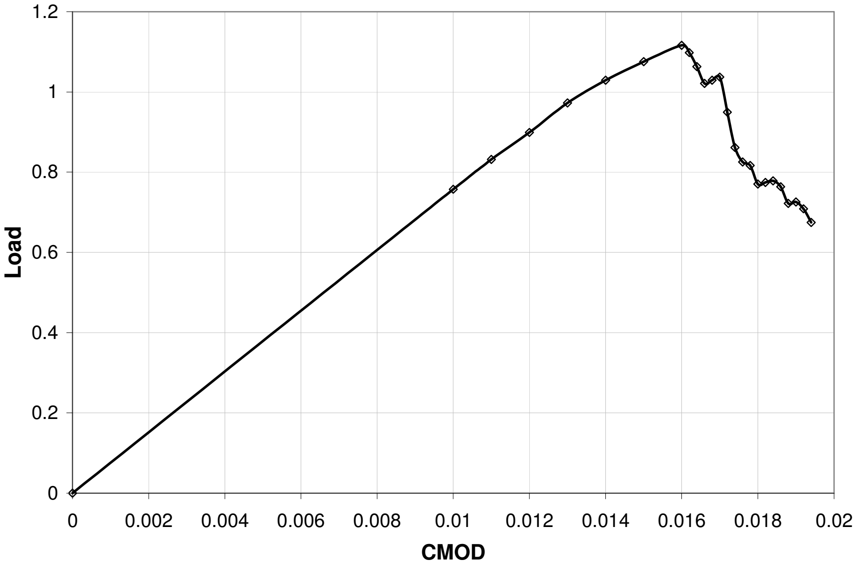


Figure 5-29. Load versus displacement

It is worthwhile to note that smoothness of the softening profile in Figure 5-29 depends on the adopted mesh discretisation along the crack trajectory, as demonstrated [98] in the context of fracturing with partition of unity based finite elements.

5.14 Remarks regarding NMM and XFEM

NMM and XFEM [15, 16] exhibit strong parallels with respect to modelling discontinuities using discontinuous shape functions, as they are both based on the partition of unity framework. In chronological terms, the concept of modelling discontinuities using the partition of unity concept can be traced back to NMM [50], which appears in literature [95, 96] at almost the same time as the identification of the partition of unity framework [6].

The key difference between the two techniques is that NMM captures the jump in the displacement field using discontinuous displacement functions to the 'left' and to the 'right' of a discontinuity in order to restore the partition of unity. XFEM on the other hand, introduces an additional degree of freedom at each node by enriching the trial function. In both methods the resulting stiffness matrix is symmetric and there is an increase of its size due to the introduction of additional degrees of freedom.

Another distinctive difference between the two methods lies in the way the integration process is carried out. In XFEM integration is carried out numerically at Gauss points, while in NMM integration is undertaken analytically using simplex integration. Consequently, XFEM requires a form of partitioning of elements affected by the discontinuity, yet without introducing any additional degrees of freedom, so that numerical integration can account accurately for the fractured surfaces on either side of the discontinuity.

In NMM no additional step is taken for integration due to the multivariate polynomial form of cover displacement functions and the inherent ability to undertake integration explicitly, even for higher-order basis functions, using simplex elements. However, if the effects of partial cracking are to be captured, the displacement function needs to be modified with special enrichments. This would have to be dealt with using an XFEM-like integration approach.

It is worthwhile to note that as Jirásek [50] has shown, XFEM can be cast in a format very similar to that described here. This means that the substantial amount of research that has been carried out in the context of XFEM (such as tracking discontinuities using level sets and resolving crack branching in two and three-dimensions) can be potentially utilised in further developments of NMM.

5.15 *Concluding remarks*

This chapter has described the basic concepts and implementation of NMM for modelling discontinuities in continua, whereby displacement jumps are represented in a strong form by enhancement of the finite element space using discontinuous partition of unity functions. The principal advantages of this approach over traditional modelling techniques, with interface elements or discontinuities aligned with the mesh boundary, can be summarised as follows:

1. Objectivity
2. Computational efficiency
3. Ease of implementation
4. Independency from element type and structure

Resolution of the displacement and strain field in the vicinity of jumps is fully discontinuous and therefore, it can represent discrete discontinuities such as cracks realistically. In addition, the proposed model is not susceptible to ill-effects associated with classical continuum-based weak discontinuity models and zero energy dissipation.

Traditional finite element techniques for modelling strong discontinuities are normally associated with interface models and remeshing. The use of interface models without remeshing implies the adoption of *a priori* assumptions with regard to location and trajectory of potential discontinuities, and therefore may yield results that are not objective due to mesh alignment and issues associated with integration and fictitious elastic stiffness of interface elements. If interface elements are activated only when required, then the use of remeshing techniques is entailed. This implies that continuum elements intersected by discontinuities are remeshed and interface elements are introduced and aligned to boundaries of discontinuities. However, whenever a localization zone is remeshed, the immediately neighbouring region may also be remeshed in order to obtain a smooth transition of the approximation field. This process can result in a significant increase of unknowns.

In NMM, enrichment of the continuum with additional degrees of freedom that represent displacement jumps follows directly from the original topology, so that additional degrees of freedom are introduced only on existing nodes intersected by discontinuities. No surrounding elements are affected and therefore remeshing is not taking place in the traditional sense. Consequently, the procedure is the same for any element type and structure.

Hence, even if the intrinsic computational cost of a remeshing algorithm alone is ignored, the number of additional degrees of freedom resulting with NMM can be significantly smaller than that obtained from remeshing and the use of interface elements.

Another particularly attractive feature of introducing discontinuities using the approach described here is that in essence, the original trial function is not altered. Instead, only the weight functions that interpolate the displacement field are modified in the vicinity of discontinuities.

Consequently, the ability to improve the level of approximation locally or globally, up to any theoretical degree is preserved. Furthermore, integration can be undertaken explicitly (as discussed in Chapter 6) for any level of the approximation, without any particular additional considerations.

Although the introduction of discontinuities with partition of unity functions can achieve objective representation of discrete phenomena, the sharp nature of the displacement field in the vicinity of discontinuities can manifest high stress gradients. Consequently, local improvement strategies of the approximation field in the vicinity of newly-introduced discontinuities may be necessitated.

6 Integration

6.1 *Introduction*

A key aspect of NMM is the ability to undertake integration of element functions analytically via simplex integration. This is possible since the integrals are restricted to simplex domains, even though the functions may comprise complex polynomials. In contrast, integration of these functions in FEM is usually undertaken numerically.

In the case that discontinuities exist and simplex elements are cut into effective and ineffective regions, the domain over which an integration is to be performed may no longer be a simplex (Figure 6-1). This can also be treated naturally using simplex integration.

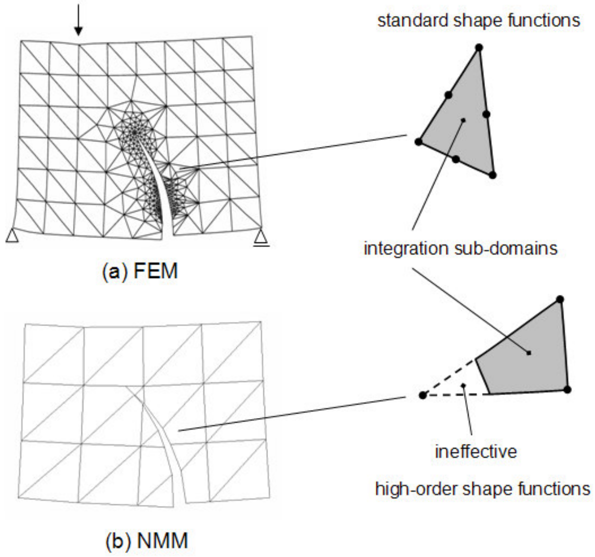


Figure 6-1. (a) Typical integration sub-domain in standard FEM with 6-node triangular elements (b) Typical integration sub-domain in high-order NMM with 3-node triangular elements and ineffective covers

This chapter reviews the principles of simplex integration; furthermore, it demonstrates its computer implementation for irregular domains, higher-order displacement functions and local enrichment. Finally, the potential advantages and disadvantages of the method are discussed.

6.2 *Integration techniques*

Integration of the variational principles and discretised system discussed in Chapter 3 can in general be undertaken using the same numerical approaches employed in FEM [133]. However, most work on NMM to date has utilised triangular elements and simplex (exact) integration, although there are a few examples in which numerical techniques have been employed for integration over rectangular [25, 91] and circular [23] domains.

The principal advantage of numerical integration in FEM over exact integration is versatility, particularly with non-simplex element geometries or shape functions constructed based on the interpolation of a number of nodes greater than that defined by the vertices of the simplex (the simplest possible shape in a given space).

On the other hand, the principal advantage of analytical integration techniques and simplex integration is precision. However, since such exact techniques rely on analytical derivations, they can generally apply to a limited range of integrand functions and sub-domain geometries.

A particularly attractive characteristic associated with simplex geometries (triangles in two dimensions, tetrahedra in three dimensions) is that triangulation can be performed for any shape in a fully automated manner for two- or three-dimensional domains. In addition, assemblages of simplices can reproduce polygonal or polyhedral domains exactly and accurate approximations of complex circular or spherical domains.

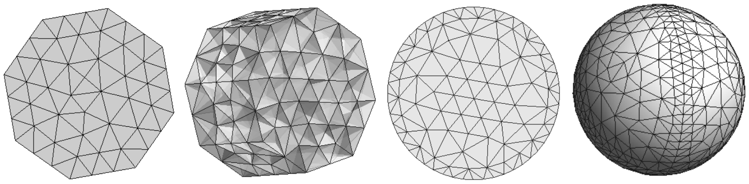


Figure 6-2. Polygonal, polyhedral, circular and spherical domains discretised by simplex elements

However, triangular and tetrahedral elements with standard shape functions based on three-node (in two dimensions) and four-node (in three dimensions) interpolation schemes in FEM are well-known to perform poorly for certain classes of problems.

Unlike FEM with standard shape functions, higher-order NMM formulations offer improved performance and deformability even with three-node triangular or four-node tetrahedral elements. In addition, the integrated functions are relatively simple due to the choice of polynomial displacement functions. In fact, integral terms of any order displacement functions are of the form of Equation (6.1) in three-dimensions, for non-negative integer exponents n_1, n_2 and n_3:

$$\iiint_V x_1^{n1} x_2^{n2} x_3^{n3} dx_1 dx_2 dx_3 \tag{6.1}$$

Due to the nature of the integral kernels and the improved performance and deformability associated with higher-order displacement functions, coupled with the versatility of simplex element geometries and the precision offered by simplex integration, it can be argued that it is not necessary to use numerical integration or non-simplex elements coupled with higher-order NMM.

6.3 Exact integration in simplex domains

A simplex is defined as the simplest possible polytope in any given space [124]. In a single dimension, the simplex is a line segment, in two dimensions it is a triangle whereas in three dimensions it is a tetrahedron (Figure 3-5). The concept can be generalised to a space of any number of dimensions.

Simplex integration can be carried out using analytical expressions of regular shapes transformed into a general coordinate system. In addition, it is possible to integrate in general polygonal areas or polyhedral volumes using domain-subdivision. This is treated in section 6.4.

In higher-order NMM it is generally desired to integrate terms of polynomials of an arbitrary order N (section 3.4). Analytical solutions of integrals of Equation (6.1) exist [40] since the 1950's and have been used in the context of FEM and the Boundary Element Method [19], as well as NMM and DDA [22, 95].

The analytical solution of the integral of a general term of a polynomial function in a regular triangle of area A_r is given by [22]:

$$\iint_{A_r} u_0^{i_0} u_1^{i_1} u_2^{i_2} \, du_1 du_2 = \frac{i_0 ! \, i_1 ! \, i_2 !}{\left(i_0 + i_1 + i_2 + 2\right)!} \tag{6.2}$$

where $i_0, i_1, i_2 \in \mathbb{N}$.

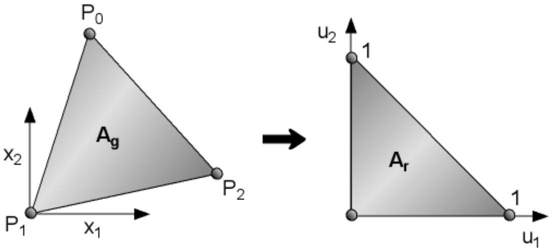


Figure 6-3. Transformation of a two-dimensional simplex from general into regular coordinates

The integral of a function in a general triangle A_g can be converted into integration in a regular area using coordinate transformation (Figure 6-3). The resulting solution for this case is:

$$\iint_{A_g} x_1^{n_1} x_2^{n_2} dx_1 dx_2 = |J| \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} \eta(n_1, n_2, k_1, k_2) x_{1P_1}^{n_1-k_1} x_{1P_2}^{k_1} x_{2P_1}^{n_2-k_2} x_{2P_2}^{k_2} \tag{6.3}$$

where:

$$\begin{aligned} \eta(n_1, n_2, k_1, k_2) = & \frac{n_1!\, n_2!}{(n_1 + n_2 + 2)!} \times \\ & \times \frac{(n_1 + n_2 - k_1 - k_2)!\,(k_1 + k_2)!}{k_1!\, k_2!\,(n_1 - k_1)!\,(n_2 - k_2)!} \end{aligned} \tag{6.4}$$

and J is the Jacobian determinant:

$$J = \begin{vmatrix} x_{1P_1} - x_{1P_0} & x_{1P_2} - x_{1P_0} \\ x_{2P_1} - x_{2P_0} & x_{2P_2} - x_{2P_0} \end{vmatrix} \tag{6.5}$$

The derivation of Equation (6.4) is given in detail in Appendix A.

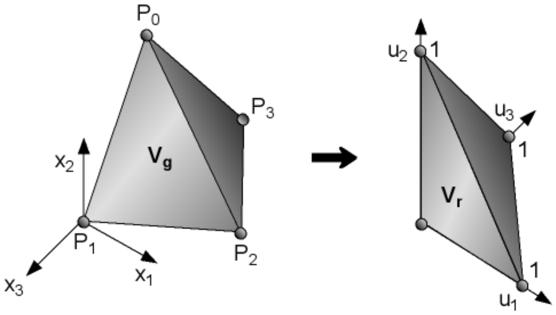


Figure 6-4. Transformation of a three-dimensional simplex from general into regular coordinates

Similarly, in three dimensions the analytical solution of the integral of a general term of a polynomial function in a polyhedron is given by:

$$\begin{aligned} & \iiint_{V_g} x_1^{n_1} x_2^{n_2} x_3^{n_3} dx_1 dx_2 dx_3 = \\ & \sum_{k_1=1}^{ns} \sum_{k_2=1}^{nv} \left[\begin{aligned} & J_{k_1 k_2} \sum_{l_1=0}^{n_1} \sum_{l_2=0}^{n_2} \sum_{l_3=0}^{n_3} \sum_{m_1=0}^{l_1} \sum_{m_2=0}^{l_2} \sum_{m_3=0}^{l_3} \alpha(n_1, n_2, n_3, l_1, l_2, l_3, m_1, m_2, m_3) \\ & \times f\left(P_{k_1 k_2}, P_{k_1 k_2 + 1}, P_{k_1 k_2 + 2}\right) \end{aligned} \right] \end{aligned} \tag{6.6}$$

where:

$$\begin{aligned}\alpha\left(n_1,n_2,n_3,l_1,l_2,l_3,m_1,m_2,m_3\right) &= \\ &= \frac{n_1!\,n_2!\,n_3!\left(n_1+n_2+n_3-l_1-l_2-l_3\right)!}{m_1!\,m_2!\,m_3!\left(n_1+n_2+n_3+3\right)!}\times \\ &\times\frac{\left(l_1+l_2+l_3-m_1-m_2-m_3\right)!\left(m_1+m_2+m_3\right)!}{\left(n_1-l_1\right)!\left(l_1-m_1\right)!\left(n_2-l_2\right)!\left(l_2-m_2\right)!\left(n_3-l_3\right)!\left(l_3-m_3\right)!}\end{aligned} \qquad (6.7)$$

$$\begin{aligned}& f\left(P_{k_1k_2},P_{k_1k_2+1},P_{k_1k_2+2}\right)= \\ &= x_{1Pk_1k_2}^{n_1-l_1}x_{1Pk_1k_2+1}^{l_1-m_1}x_{1Pk_1k_2+2}^{m_1}x_{2Pk_1k_2}^{n_2-l_2}x_{2Pk_1k_2+1}^{l_2-m_2}x_{2Pk_1k_2+2}^{m_2}x_{3Pk_1k_2}^{n_3-l_3}x_{3Pk_1k_2+1}^{l_3-m_3}x_{3Pk_1k_2+2}^{m_3}\end{aligned} \qquad (6.8)$$

and:

$$J_{k_1k_2}=\begin{vmatrix} x_{1Pk_1k_2} & x_{1Pk_1k_2+1} & x_{1Pk_1k_2+2} \\ x_{2Pk_1k_2} & x_{2Pk_1k_2+1} & x_{2Pk_1k_2+2} \\ x_{3Pk_1k_2} & x_{3Pk_1k_2+1} & x_{3Pk_1k_2+2}\end{vmatrix} \qquad (6.9)$$

ns is the total number of surfaces of the volume (for tetrahedra $ns=4$), nv is the number of vertices (for tetrahedra again $nv=4$) of surface k_1 with coordinates $\left(x_{1Pk_1k_2},x_{2Pk_1k_2},x_{3Pk_1k_2}\right)$.

6.4 Integration in arbitrary polygons and polyhedral volumes

Integration in an arbitrary polygon of n vertices with orientated boundary $\partial\Gamma=P_1P_2P_3...P_n$ can be calculated by the closed loop algebraic sum of the integrals of n triangular sub-domains, where each sub-domain is formed by the vertex of a fixed arbitrary point P_0 and two successive polygon vertices [Figure 6-5, Figure 6-6], [22]. The same concept can be applied in integration of polyhedral volumes in three dimensions [22, 95].

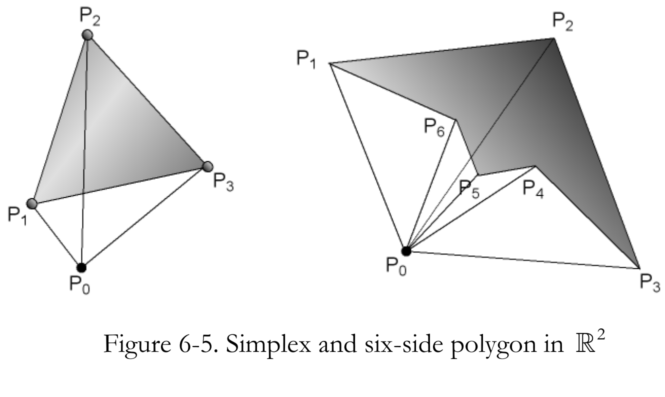


Figure 6-5. Simplex and six-side polygon in $\mathbb{R}^2$

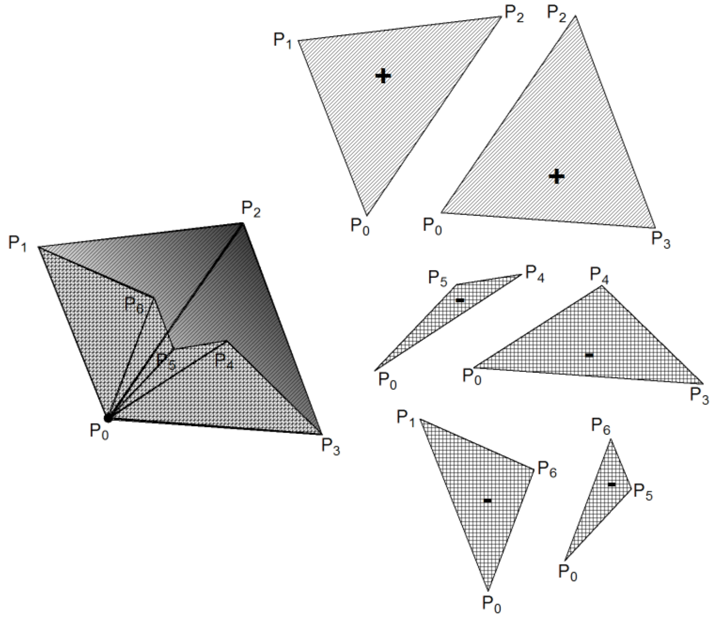


Figure 6-6. Simplex decomposition of a polygon in $\mathbb{R}^2$

In a two-dimensional simplex:

$$\begin{aligned}\iint_{P_1P_2P_3} f\left(x_1,x_2\right)dx_1dx_2 &= \iint_{P_0P_1P_2} f\left(x_1,x_2\right)dx_1dx_2 + \\ +\iint_{P_0P_2P_3} f\left(x_1,x_2\right)dx_1dx_2 &+ \iint_{P_0P_3P_1} f\left(x_1,x_2\right)dx_1dx_2\end{aligned} \tag{6.10}$$

Thus, in a general polygon:

$$\begin{aligned}&\iint_{P_1P_2\ldots P_n} f\left(x_1,x_2\right)dx_1dx_2 = \\ &= \sum_{k=1}^{n}\iint_{P_0P_kP_{k+1}} f\left(x_1,x_2\right)dx_1dx_2 + \iint_{P_0P_nP_1} f\left(x_1,x_2\right)dx_1dx_2\end{aligned} \tag{6.11}$$

Similarly, Equation (6.11) can be generalised for polyhedra in three dimensions:

$$\begin{aligned}&\iiint_{P_1P_2\ldots P} f\left(x_1,x_2,x_3\right)dx_1dx_2dx_3 = \\ &= \sum_{k=1}^{n}\iiint_{P_0P_kP_{k+1}P_{k+2}} f\left(x_1,x_2,x_3\right)dx_1dx_2dx_3 \\ &+\iiint_{P_0P_{n-1}P_nP_1} f\left(x_1,x_2,x_3\right)dx_1dx_2dx_3\end{aligned} \tag{6.12}$$

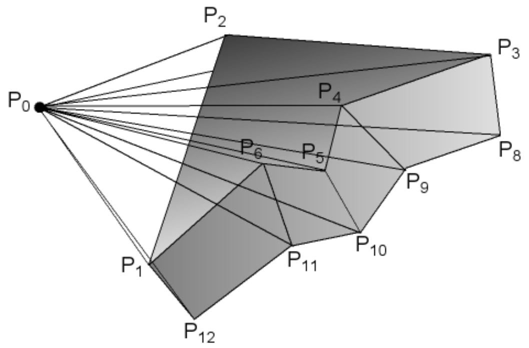


Figure 6-7. Simplex decomposition in $\mathbb{R}^3$

Therefore, in situations where the use of ineffective covers can result in polygonal, or polyhedral domains (Figure 6-1 b), integration can be carried out using the approach discussed in section 6.3 and equations (6.11) or (6.12) without any further considerations. This can offer significant flexibility when compared to other integration techniques as the procedure is readily applicable to a general class of problems with arbitrary polygonal or polyhedral shapes.

6.5 *Integration for arbitrary levels of global and local enrichment*

In cases where global or local enrichment is undertaken by use of higher-order basis functions, the stiffness matrix is populated by integrals of higher-order polynomial terms of the form of Equation (6.1). The procedure outlined previously also applies to higher order kernels of polynomial displacement functions. Therefore, no additional considerations are required in such cases.

6.6 *Numerical implementation*

The simplex concept outlined previously can be implemented numerically for any general high order case using simple algorithms. This section illustrates the implementation of the simplex integration of high-order polynomials in general polygons in two dimensions. The same procedure can be extended easily for polyhedra in three dimensions.

Assuming that $x^{n_1} y^{n_2}$ is integrated in the area of a general polygon with vertices $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ in two dimensions and that the vertex coordinates are stored in vectors $\mathbf{x}$ and $\mathbf{y}$:

```
nnt=size(x); nn=nnt(2); % number of points that define the area of
integration
```

```
s=0; Jac=0;

[coef] = simcoef2d(n1,n2); % function that computes the simplex
coefficient of Equation (6.4)

x(nn+1)=x(1); y(nn+1)=y(1);

for i=0:nn-1

   Jac = x(i+1)*y(i+2)-x(i+2)*y(i+1); % Jacobian

   s1=0;

   for k1=0:n1

         for k2=0:n2

            s1=s1+coef(k1+1,k2+1)*(x(i+1)^(n1-k1))*

            *(x(i+2)^(k1))*(y(i+1)^(n2-k2))*(y(i+2)^(k2));

         end

   end

s=s+s1*Jac;

end
```

Figure 6-8. MATLAB implementation of high-order simplex integration in a general polygon

The coefficient of Equation (6.4) can be computed using the following algorithm:

```
for k1=0:n1

   for k2=0:n2

      coef(k1+1,k2+1)=(factorial(n1)*factorial(n2)*factorial(n1+n2-k1-

      k2)*factorial(k1+k2))/(factorial(n1+n2+2)*factorial(k1)*factorial(k2)

      *factorial(n1-k1)*factorial(n2-k2));

   end

end
```

Figure 6-9. Algorithm for computation of simplex coefficients

The algorithm in Figure 6-8 assumes that the exponents n_1 and n_2 of the polynomial term that is to be integrated are given. In a practical case the monomial exponents have to be determined either analytically or iteratively for a general case with uniform *p*-enrichment or local enhancement respectively. The exponents of any term of a high-order polynomial function can be determined using the approach discussed in Chapter 4.

6.7 *Considerations regarding numerical accuracy and efficiency*

The computation of high-order simplex coefficients of Equations (6.4) and (6.7) in $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively can become a significantly expensive part of the procedure outlined in previous sections of this Chapter, depending on how simplex integration is implemented.

As a typical example, consider the static analysis of a triangular element with displacement functions of order $N=6$. A MATLAB (version 7.1 R14) implementation of this problem yields a solution in which 10% of the processing time is spent in assembly of the stiffness matrix, whereas 80% is expended by the integration algorithm. 82% of the integration algorithm alone (66% of the total processing time) is expended by the algorithm computing simplex coefficients.

In this case, it is found that the indirect determination of factorials as products of array elements using the generic function *prod* (Figure 6-10) rather than the generic direct function *factorial* can result in reduction of the processing time of this computation by over 90% when compared to the direct approach illustrated in Figure 6-9.

```
for k1=0:n1
    for k2=0:n2
        coef(k1+1,k2+1)=(prod(1:n1)*prod(1:n2)*prod(1:(n1+n2-k1-
        k2))*prod(1:(k1+k2)))/(prod(1:(n1+n2+2))*prod(1:k1)*prod(1:k2)*
        prod(1:(n1-k1))*prod(1:(n2-k2)));
    end
end
```

Figure 6-10. Alternative algorithm for computation of simplex coefficients in $\mathbb{R}^2$

Using the alternative approach illustrated in Figure 6-10, analysis of the 6th-order triangular element is undertaken in only 22% of the time of the original analysis, with the exact same order of accuracy. In this case, the alternative algorithm yields a solution in which 34% of the total time is expended in assembly of the stiffness matrix, whereas only 4% of the total time is expended by the integration algorithm (including computation of simplex coefficients).

Another potential issue associated with the determination of simplex coefficients is associated with computer limitations to represent large numbers accurately. For example, double precision numbers occupy 64 bits and have significant precision of 53 bits [39]. This gives double precision numbers accuracy of about 16 decimal digits.

Consequently, a double precision factorial can only be accurate for $n \leq 21$ approximately [70]. For larger order exponents of high order monomials the solution is likely to have the correct magnitude but it will be accurate only for the first 16 digits. Furthermore, for orders $n > 90$ arithmetic overflow problems can occur as the factorial numbers become too large to be represented computationally. In addition, integration of higher-order polynomial displacement functions can increase the total computation time significantly as the number of monomial terms increases according to Equation (3.24) in $\mathbb{R}^p$.

In situations such as those mentioned above, performance can be improved by approximating Equations (6.4) and (6.7) [22], therefore essentially introducing an additional, yet relatively insignificant or 'cancellating' error. In practical smooth or non-smooth problems the displacement function polynomials are not likely to exceed $N = 8$. Therefore float overflow or other performance problems will not be of particular concern unless there are constraints associated with the adopted discretisation scheme (element size).

6.8 Remarks regarding integration in NMM and XFEM

NMM has several conceptual similarities with XFEM as key-aspects of both methods emerge from the framework of partition of unity [6, 7]. However, where discontinuities emerge, NMM traditionally constructs the approximation space as an enriched product of standard basis functions, while XFEM constructs the approximation space as a product of standard basis and special enrichment functions [1, 16].

This key difference influences the integration approach adopted in each case as the shape functions and their derivatives can be conceptually different. As a result, in NMM the stiffness matrix can consist only of integrals of monomial terms, whereas in XFEM it can consist of integrals of monomial terms and other special functions.

In situations where fracture problems are considered within the framework of linear elasticity and zero traction boundary conditions on crack surfaces, the XFEM special functions are typically singular linear elastic near-tip field functions [38, 73], in order to enrich the crack-tip.

In linear elastic cohesive crack models, the situation is slightly different, as the tractions between either side of a discontinuity lead to a reduction of stress in the tip. This reduction is desirable since it reduces the non-physical singular stress field at the tip [76]. Therefore, singular enrichment functions [73] are not valid. In this case, enrichments at the tip can be undertaken using non-singular asymptotic functions [76], or enrichment functions based on higher-order polynomial bases [69].

Consequently, integration in XFEM is not always as straightforward as it is in the case of NMM and it is undertaken numerically. This in turn means that the integration domain must conform to the boundary of the discontinuity. If the discontinuity is not taken into account, then the solution can lead to poor results or a non-invertible set of equations if integration points do not track the discontinuity [56].

Therefore, integration of elements that are intersected by discontinuities requires element partitioning into triangular sub-domains although it is possible to avoid partitioning of the tip elements by introducing additional nodes [69]. In addition, in cohesive models the variational principle involves integration over the domain and integration over segments of the cohesive zone [73, 76].

XFEM partitioning is undertaken without introducing additional unknowns since basis functions are only associated with nodes tied to parent elements [105]. However, computational cost can increase as the number of integration points increases. Furthermore, integration by partitioning around a singularity can yield poor results if the integration rule is not adequate or if the mesh in the proximity of the singularity is coarse.

With simplex integration, non-simplex domains resulting by the intersection of (simplex) elements and discontinuities are also in essence partitioned into simplex sub-domains (section 6.4), similar to the XFEM approach. Also similar to XFEM, no additional unknowns are introduced.

However, unlike XFEM, integration in this case is exact and therefore the additional computational cost of integration of domains intersected by discontinuities depends only on the order of the displacement functions associated with nodes tied to parent elements. Also, this is guaranteed to yield precise results relative to a given discretisation.

Furthermore, although simplex integration constrains the shape of the approximating basis functions, it is likely to avoid the potential issues of zero energy modes associated with inadequate or reduced numerical integration rules [133] of non-smooth as well as smooth problems without the requirement for additional considerations when the order of the basis functions is increased (e.g. in adaptive enrichment).

Depending on the form of the special enrichment functions, simplex integration can also be used in XFEM. Similarly, NMM can benefit from the work undertaken in XFEM in situations where a simplex approach is not desirable, whether this is for the performance reasons discussed earlier in section 6.7, the implementation of non-simplex elements or the enrichment of the approximation field with non-standard basis functions.

6.9 Concluding remarks

One of the distinct advantages of numerical integration over exact integration in standard FE techniques is versatility, since for a certain class of problems it is often necessary to only prescribe different shape functions in order to implement different element types.

Although simplex integration constrains the shapes of approximating basis functions, it is arguable whether implementation of different element shapes (and hence use of other integration techniques) is necessary in higher-order NMM, coupled with the fact that assemblages of simplices can reproduce accurately irregular manifolds in any given space.

Simplex elements in NMM with higher-order basis functions do not generally incorporate the poor accuracy observed using the same elements with zero-order displacement functions (FEM). Coupled with the fact that simplex integration can be fully adapted for domains that are intersected by discontinuities in any space, for any arbitrary order of the cover displacement functions to yield a precise solution, the simplex approach can be attractive in situations where *h*- , *p*-, or a combination of *h*- and *p*-adaptivity is desired.

Furthermore, simplex integration is exact, therefore issues associated with the efficiency and accuracy of numerical schemes (such as the choice of a minimum rule to achieve convergence, the choice of a rule to preserve the convergence of an exact solution and generally, loss of accuracy) can be eliminated without significant computational cost.

It is worthwhile to note that, to date, application of simplex integration has not been investigated within the context of nonlinear constitutive behaviour. Although the explicit integration strategy presented herein applies readily to higher-order NMM with linear elastic behaviour and situations where discontinuities are introduced via partition of unity concepts, its generalised application may be impeded when constitutive behaviour of the continuum is nonlinear, unless the tangent stiffness matrix consists of polynomial terms.

7 Treatment of higher-order boundaries

7.1 Introduction

The use of higher-order displacement functions is clearly an attractive quality of NMM that makes it a potentially powerful approximation technique. However, the enforcement of essential boundary conditions for higher-order NMM is not straightforward. In Chapter 3 the enforcement of these constraints was described in general terms using the point collocation method, whereby the boundary conditions are enforced exactly at specified points on the boundary. However, if the boundary conditions require an entire edge to be constrained and only the nodes are utilised for this purpose, the increased deformability associated with higher-order NMM will result in incomplete enforcement (Figure 7-1). A similar problem manifests itself if distributed loads are not applied consistently.

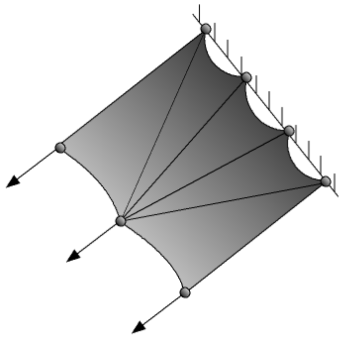


Figure 7-1. Higher-order boundary effects

To resolve this it is possible to restrain additional points along the edge. This approach was utilised in section 4.9. The number and positioning of these additional points will determine whether the boundary conditions are enforced exactly, or approximately. However, this can be cumbersome from a numerical implementation point of view.

Two alternative techniques are the direct enforcement of essential constraints using constant-strain elements at the boundary or the use of zero-order displacement functions for nodes on the boundary. The latter approach relies on the local enrichment strategy discussed in Chapter 4.

Perhaps more significant is that the imposition of essential boundary conditions associated with higher-order displacement functions can lead to rank deficiency of the system matrix. This is due to the NMM displacement functions at restrained nodes resulting in a non-unique solution due to presence of monomials that are reproduced by the weighting functions (Duarte [37] and subsequently Lin [60]). This will be discussed in more detail later.

7.2 *Boundary deformability in higher-order NMM*

In FEM, due to the interpolation of nodal displacements, constraints on element boundaries between nodes are enforced naturally (Figure 7-2 (a)). This is not the case in NMM with higher-order displacement functions (Figure 7-2 (b)), since deformation between nodes associated with higher-order functions is not anymore an interpolation of the nodal displacements. Figure 7-2 (b) also illustrates the effect of inconsistently applied loads.

Both of the above do not normally apply in NMM with higher-order displacement functions (Figure 7-2), since deformation between nodes associated with higher-order functions is not anymore a linear interpolation between the displacements of these nodes. In addition, the unknowns corresponding to nodes associated with higher-order functions are not simply displacements and the unknowns of higher-order functions associated with constrained nodes are not all necessarily zero.

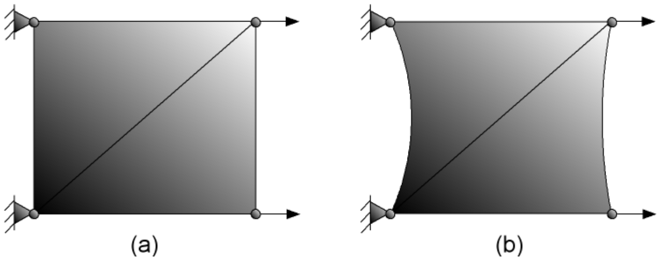


Figure 7-2. Deformed shape of a two-element test. (a) FEM with constant-strain triangles (b) NMM with higher-order displacement functions

This section investigates further the origin of these issues using the benchmark test devised in Chapter 4, in order to develop the potential treatment strategies illustrated in following sections of this chapter.

Consider the single element test of Figure 7-3.

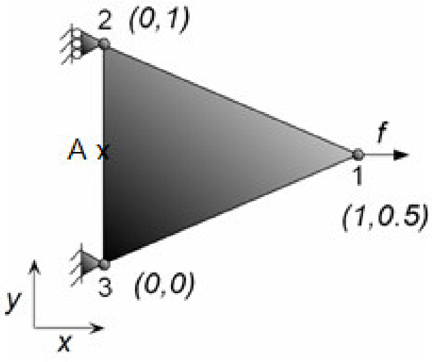


Figure 7-3. Single element test

The number of unknowns is given by:

$$3 \times \sum_{i=1}^{3} (N_i + 1)(N_i + 2) \tag{7.1}$$

where $N_i = 0,1,2,3,4 \ldots n$ is the order of polynomial cover displacement functions associated with nodes $i = 1$ to 3. Here, $N_i = 1$, i.e. first-order displacement functions. A force $f = 0.5$ is applied at node 1 in the positive x direction. Boundary constraints are enforced using the projection matrix approach proposed in Chapter 3. The elastic modulus and Poisson's ratio are taken as 100 and 0.0 respectively. The displacement of point A is monitored.

If nodes 1, 2 and 3 are associated with first-order displacement polynomials, from Figure 4-15 it is evident that point A between the constraints of the left-hand edge displaces; i.e. the constraints of nodes 2 and 3 do not automatically enforce displacement constraints between these two nodes.

Furthermore, as illustrated in Figure 4-16, the stress field is not constant, compared to the zero-order (constant-strain) case of Figure 4-14. This can be expected since high-order displacement functions are employed. Because the weight functions remain linear but are now multiplied by first-order polynomials, the interpolants are of second order. An analogy for this situation in the context of FEM with standard shape functions would be an element with mid-nodes, which is constrained only at corner nodes.

The reason for the displacement of point A may not be immediately clear, since nodes 2 and 3 are restrained. It is known that the weight function of node 1 is zero along the left edge between nodes 2 and 3, whereas the weight functions of nodes 2 and 3 at the same location are both equal to 0.5 (mid-way between the supports). This fact may complicate the issue further since it essentially implies that there is no translation contributed from node one.

The explanation to this issue is that, although the weight function of node 1 is indeed zero along the edge defined by nodes 2 and 3, the unknowns as well as weight functions of nodes 2 and 3 are not all zero. This can be shown analytically.

The displacement at any point has been defined in Chapter 3 as:

$$\mathbf{u} = \sum_{i=1}^{3} \mathbf{T}_i \, \mathbf{a}_i \tag{7.2}$$

where $\mathbf{T}_i$ is the shape function matrix, which is product of the weight function $w_i(x, y) = a + bx + cy$ and the position components associated with the displacement function at node i. If the linear displacement function is given as:

$$u_i = \beta_{i1} + \beta_{i2}x + \beta_{i3}y \tag{7.3}$$

where $\beta_{1,2,3}$ are the displacement function coefficients (unknowns), then $\mathbf{T}_i$ is equal to:

$$\mathbf{T}_i = w_i \begin{bmatrix} 1 & 0 & x & 0 & y & 0 \\ 0 & 1 & 0 & x & 0 & y \end{bmatrix} \tag{7.4}$$

and the vector of unknowns of node i is:

$$\mathbf{a}_i = \begin{bmatrix} \beta_{i1x} & \beta_{i1y} & \beta_{i2x} & \beta_{i2y} & \beta_{i3x} & \beta_{i3y} \end{bmatrix}^T \tag{7.5}$$

For the problem under consideration, it can be shown that the weight function of node 1 is derived as:

$$w_1 = 0 + 1x + 0\,y \tag{7.6}$$

For any point along the left-hand edge:

$$w_1 = 0 + 1 \times 0 + 0 \times y_{2-3} = 0 \tag{7.7}$$

Furthermore, the weight functions of nodes 2 and 3 along the left edge are:

$$w_2 = 0 - 0.5 \times 0 + 1 \times 0.5 = 0.5 \tag{7.8}$$

$$w_3 = 1 - 0.5 \times 0 - 1 \times 0.5 = 0.5 \tag{7.9}$$

Although nodes 2 and 3 do not translate, it can be shown that the corresponding displacement coefficient vectors derived from the solution are equal to:

$$\mathbf{a}_2 = \begin{bmatrix} 0.003 & -0.106 & 0.015 & -0.051 & -0.003 & 0.099 \end{bmatrix}^T \tag{7.10}$$

$$\mathbf{a}_3 = \begin{bmatrix} 0 & 0 & -0.049 & 0.073 & 0.038 & 0.099 \end{bmatrix}^T \tag{7.11}$$

Using the definition given in Equation (7.4), at point five (0, 0.5) the shape function matrices of nodes 2 and 3 along the left edge are:

$$\mathbf{T}_2 = y\begin{bmatrix} 1 & 0 & 0 & 0 & y & 0 \\ 0 & 1 & 0 & 0 & 0 & y \end{bmatrix} \tag{7.12}$$

$$\mathbf{T}_3 = y\begin{bmatrix} 1 & 0 & 0 & 0 & y & 0 \\ 0 & 1 & 0 & 0 & 0 & y \end{bmatrix} \tag{7.13}$$

Therefore,

$$u_x = 0.041y(1-y) \tag{7.14}$$

i.e. the displacement in the x direction is parabolic, satisfies the boundary conditions and yields a displacement of 0.103 at point A.

7.3 Treatment of higher-order boundaries

The previous section demonstrated that deformation of boundaries between constrained nodes is not automatically enforced when higher-order displacement functions are employed. Evidently, this lack of constraint may not always be desired in structural analysis.

Furthermore, loadings have to be applied consistently with respect to the order of the displacement functions. Consider the problem in Figure 7-4, comprising two triangular elements with 1st-order displacement functions.

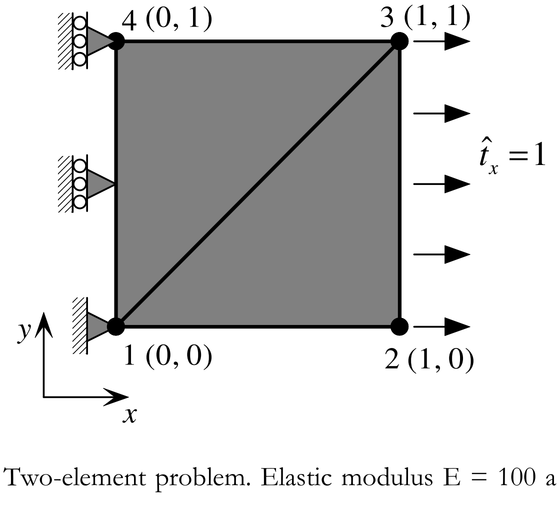


Figure 7-4. Two-element problem. Elastic modulus E = 100 and Poisson's ratio ν = 0.

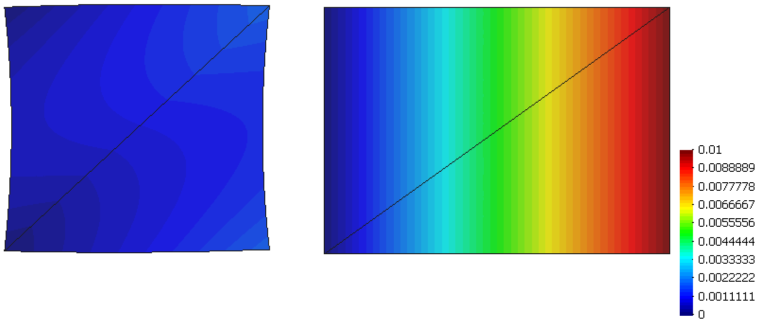


Figure 7-5. Contours of horizontal displacement and deformed mesh (x40); Left: Inconsistently applied boundary conditions and loads. Right: Consistently applied boundary conditions and loads

Figure 7-5 (left) shows the deformed shape and contours of horizontal displacements when the left hand side constraints are only applied at the nodes and the load on the right hand edge is applied as point loads. In comparison, Figure 7-5 (right) shows the results using sufficient constraints (i.e. nodes 1, 4 and an additional point at the mid point between these two nodes) on the left edge and consistently applied loads on the right edge using $\mathbf{f} = \int_{\Gamma_t} \mathbf{T}^T \hat{\mathbf{t}}\, d\Gamma$.

Two potentially simpler yet more general approaches for applying point loads and constraints are discussed in the following two sections.

7.4 *Coupling with Finite Elements*

The concept of treating boundary domains with finite elements may be recognizable from work in meshless methods and BEM. This type of coupled approximation is occasionally employed due to the attractive properties of FEM with regard to application of loads and enforcement of constraints.

Whereas this type of coupled approximation may be relatively complicated in general numerical techniques as it is not natural, in NMM it can be achieved simply by utilising a boundary layer of simplex elements associated with zero-order displacement functions, which are equivalent to constant strain finite elements as discussed previously.

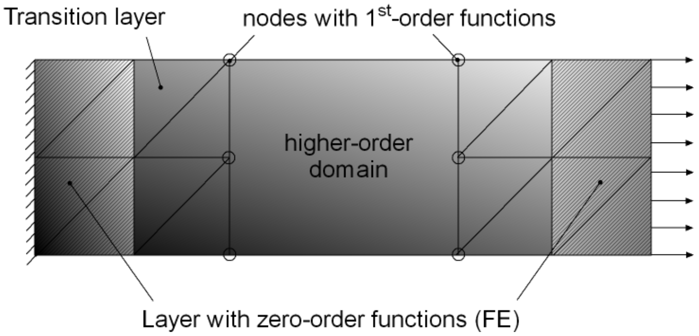

Transition layer
nodes with 1st-order functions
higher-order domain
Layer with zero-order functions (FE)

Figure 7-6. Example of a beam in tension with constant strain elements at the left and right-hand boundaries

This strategy requires a transition layer of elements with mixed displacement functions between the boundary layer and the rest of the domain. However, this can be implemented and fully automated easily using the local enhancement methodology discussed in Chapter 4.

A potential drawback of this approach is that the ability to tailor the approximation at the boundaries using higher-order displacement functions is lost. Depending on the adopted discretisation, the approach may introduce detrimental errors. Generally, a relatively fine mesh discretisation near the boundary is recommended to avoid potential numerical difficulties.

Furthermore, for certain types of discretisations the approach may not be attractive as it can eliminate the ability to achieve a better approximation altogether without undertaking some form of mesh refinement.

A more developed and attractive adaptation of the same concept is discussed in the following section.

7.5 *Treatment of higher-order boundaries with zero-order functions*

An alternative approach is to enforce zero-order displacement functions only on the boundary nodes but continue to permit higher-order functions for all other nodes. This approach implies that the displacement function coefficients associated with the displacement boundary nodes are the nodal displacements. These are then interpolated between boundary nodes, thereby enforcing the correct boundary conditions along the entire boundary and not only at the nodes (Figure 7-7).

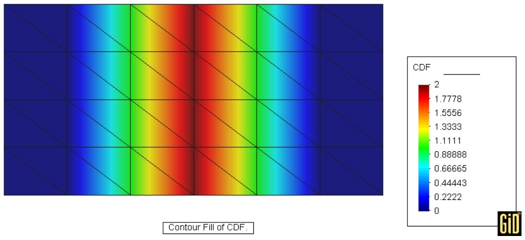


Figure 7-7. Beam in tension – contours of order of displacement functions. Boundary nodes with applied constraints and loads are discretised with zero displacement functions.

This can be demonstrated using the single element test of Figure 7-3. If zero displacement functions are employed at nodes 2 and 3, then the only variables at these nodes are the (already known) prescribed displacements:

$$\mathbf{a}_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{7.15}$$

$$\mathbf{a}_3 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{7.16}$$

At point A between these nodes, the weight functions of both nodes 2 and 3 are 0.5 (as discussed previously), therefore the associated shape matrices are equal to:

$$\mathbf{T}_2 = 0.5\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.17}$$

$$\mathbf{T}_3 = 0.5\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.18}$$

whereas the weight function of node 1 at point A is zero, since this point lies on the boundary:

$$\mathbf{T}_1 = 0\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0 \tag{7.19}$$

Therefore, the displacement of node five equals zero (Figure 7-8), although the resulting stress field is not constant as illustrated by Figure 7-9:

$$\begin{bmatrix} u_{5x} \\ u_{5y} \end{bmatrix} = 0 + \mathbf{T}_2 \mathbf{a}_2 + \mathbf{T}_3 \mathbf{a}_3 = 0 \tag{7.20}$$

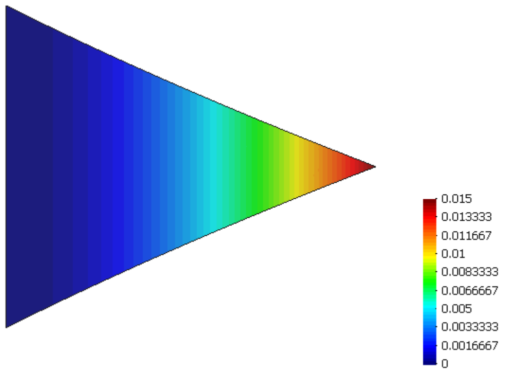


Figure 7-8. Single element test with zero-order displacement functions at nodes 2 and 3 and 1st-order displacement functions at node 1. Contours of displacement along the horizontal axis and deformed shape (×10).

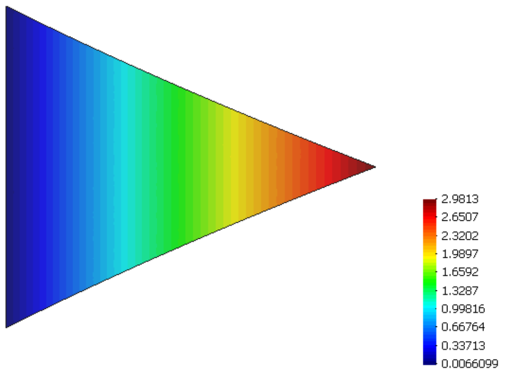


Figure 7-9. Single element test with zero-order displacement functions at nodes 2 and 3 and 1st-order displacement functions at node 1. Contours of stress in the horizontal axis and deformed shape (×10).

Another numerical example to illustrate this concept is that of a concrete compact tension specimen (CTS) in Figure 7-10, Figure 7-11 and Figure 7-12. The specimen features a pre-existing traction-free crack. The left-hand edge of the specimen is fully restrained, whereas a distributed load is applied on the right-hand edge. The specimen is discretised with 1st-order displacement functions, whereas the region around the crack tip is discretised with 2nd-order functions. The constrained boundary is prescribed zero-order functions in order to enforce constraints directly. Furthermore, the load boundary is also discretised with zero-order functions in order to avoid higher-order boundary effects.

It is clear that the adopted approach allows the loading and boundary conditions to be imposed effectively and the expected stress concentration around the crack tip to be captured.

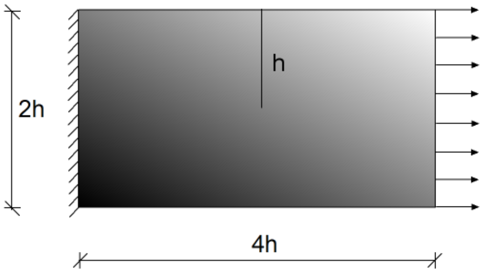


Figure 7-10. Compact tension specimen (CTS) with pre-existing crack

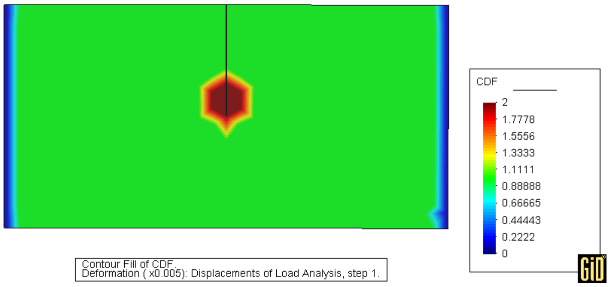


Figure 7-11. CTS contours of the order of displacement functions used to approximate the solution field. Constraint and load boundaries are discretised with zero-order functions.

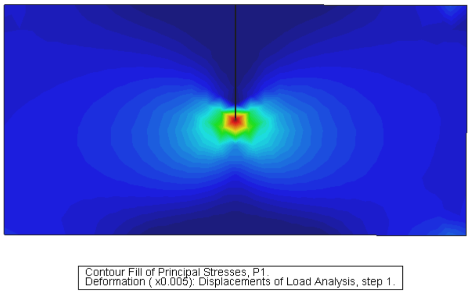


Figure 7-12. CTS contours of major principal stress

7.6 *Shape functions and linear dependence*

It has been observed that either local or global enhancement can result in an ill-conditioned system matrix that can potentially lead to inaccurate results. Duarte [37] and subsequently Lin [60] noted this phenomenon in the context of h-p Clouds and NMM respectively and recognised that this was due to the fact that the polynomials used to construct the displacement function included monomials that are reproduced by the weighting functions. Lin proposed that the linear term should be omitted from the displacement functions. In fact this only partially tackles the problem.

This issue is most easily examined by considering 1D NMM using 2-node simplex elements. The displacement at a node i is given by (note the weighting functions of all other nodes are zero at node i):

$$\begin{aligned} u &= w_i \alpha_i \\ &= \left(a_i + b_i x\right)\left({}^{i}\beta_1 + {}^{i}\beta_2 x + {}^{i}\beta_3 x^2 + \ldots + {}^{i}\beta_{N+1} x^N\right) \\ &= {}^{i}\beta_1 a_i + \left({}^{i}\beta_1 b_i + {}^{i}\beta_2\right)x + \left({}^{i}\beta_2 b_i + {}^{i}\beta_3\right)x^2 + \ldots + {}^{i}\beta_{N+1} b_i x^{N+1} \end{aligned} \tag{7.21}$$

where ${}^{i}\beta_j$ are the unknown displacement function coefficients. Clearly, if u is prescribed to a given value, there is a non-unique solution for the unknown coefficients. To avoid this, it is necessary to remove those terms with odd exponents:

$$\begin{aligned} u &= \left(a_i + b_i x\right)\left({}^{i}\beta_1 + {}^{i}\beta_3 x^2 + {}^{i}\beta_5 x^4 + \ldots + {}^{i}\beta_{N+1} x^N\right) \\ &= {}^{i}\beta_1 a_i + {}^{i}\beta_1 b_i x + {}^{i}\beta_3 a_i x^2 + {}^{i}\beta_3 b_i x^3 + \ldots + {}^{i}\beta_{N+1} b_i x^{N+1} \end{aligned} \tag{7.22}$$

Therefore, at restrained nodes only, it is necessary to use modified displacement functions to avoid repeated monomials. The simplest and most practical approach is to use a zero-order displacement function at these restrained nodes, thereby ensuring linear independence of the system matrix. This also provides a convenient and straightforward technique for imposing essential boundary conditions (as described earlier).

In order to investigate the conditioning of the stiffness matrix consider the problem of Figure 7-13:

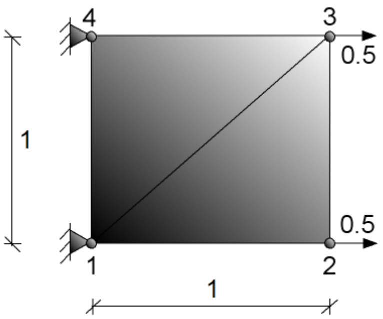


Figure 7-13. Two-element test

The test consists of two three-node elements. Nodes 1 and 4 are fully restrained while a force of 1 is distributed to nodes 2 and 3. Young's modulus is taken as 1000 while Poisson's ratio is taken as 0.0. The displacement function is increased uniformly on all four nodes.

The aim is to investigate how the rank (the number of linearly independent rows or columns) of the stiffness matrix is affected. In order to do this, the rank of the original stiffness matrix $\mathbf{K}$ and the constrained stiffness matrix $\mathbf{K_C}$ can be plotted against the order of displacement functions N. The number of linearly dependent rows (difference between the total degrees of freedom and the rank of $\mathbf{K_C}$) are also plotted.

Two cases are investigated:

A. The displacement function is increased uniformly on all four nodes.

B. The boundary nodes are prescribed with zero-order functions while the order of the displacement functions associated with the other two nodes is increased

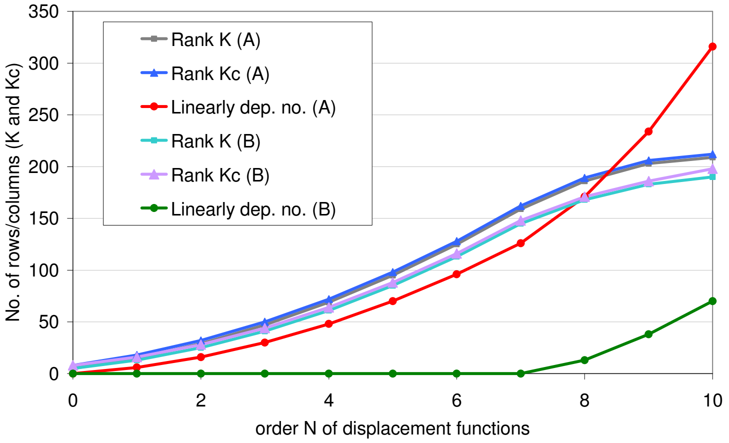


Figure 7-14. Conditioning of two-element test for uniform enrichment (case A) and enrichment of non-prescribed boundaries only (case B)

Figure 7-14 illustrates the rank of the original stiffness matrix $\mathbf{K}$ and the constrained stiffness matrix $\mathbf{K_C}$ against the order of displacement functions N. In addition, the number of linearly dependent rows is also plotted in each case.

In case A it appears that $\mathbf{K_C}$ is of higher rank than $\mathbf{K}$, although it is still rank-deficient. In case B, $\mathbf{K_C}$ is again of higher rank than $\mathbf{K}$ and the rank profiles of both matrices follow the profiles observed in case A for a reduced number of degrees of freedom. In case B, the constrained system has no deficiency up to $N = 8$, unlike case A in which only the constant strain situation $(N = 0)$ is not associated with linear dependence.

Linear dependence and rank deficiency for this problem is resolved with the use of techniques that modify the order of displacement functions of the boundary only (see section 7.4 and 7.5), albeit for $N < 8$ (Figure 7-14)

Consider again the two-element test of Figure 7-13. Two new cases are investigated:

i. the standard displacement functions of the complete domain are increased uniformly

ii. the displacement functions of the complete domain are increased uniformly but the linear term is omitted

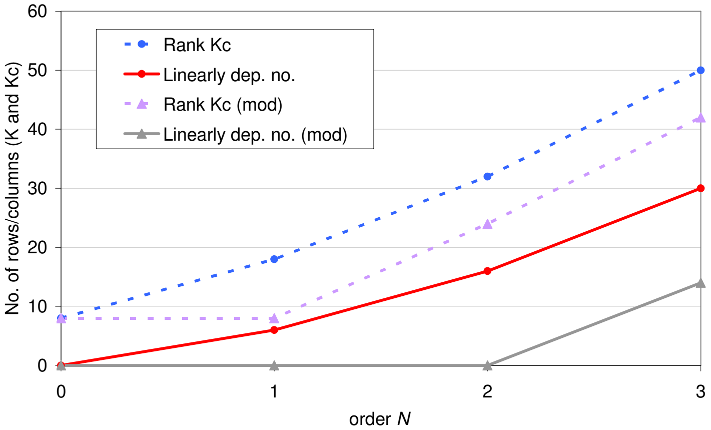


Figure 7-15. Conditioning of two-element test for uniform enrichment with standard and modified displacement functions

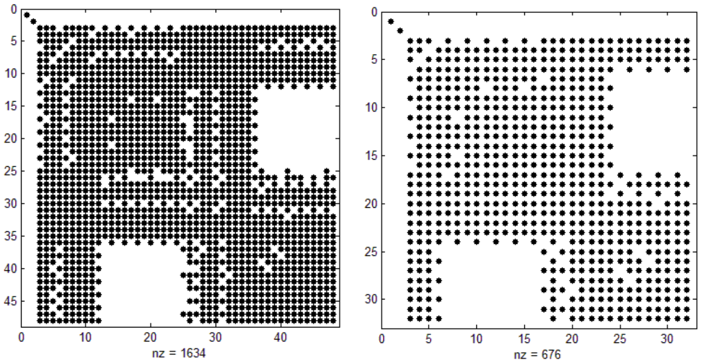


Figure 7-16. Density of non-zero terms of the stiffness matrix for the 2nd-order case of the test problem for the original matrix (left) and with the linear term omitted (right)

For the first-order case, removal of linear monomials from the displacement functions reduces the polynomials to zero-order; therefore the problem essentially reduces to the traditional FEM constant-strain case with standard shape functions, which is not associated with rank deficiency. For the second-order case, removal of the linear monomials ensure the system of equations remain linearly independent. This could also have been achieved if the linear monomials were removed from only displacement functions associated with the prescribed boundary nodes. However, as expected, for the third-order case, exclusion of the linear monomials does not solve the problem of linear dependence and rank deficiency.

7.7 *Concluding remarks*

Although the enrichment of NMM with higher-order shape functions can potentially improve the approximation, it can also potentially lead to insufficiently constrained boundaries and ill-conditioning issues. This chapter has shown that for higher-order NMM care must be given to apply loads and boundary conditions consistently. In the case of boundary conditions, a number of strategies was presented and their performance was investigated. Perhaps the most straightforward approach is the use of zero-order displacement functions for nodes on restrained boundaries.

The issue of rank deficiency of the system matrix has also been investigated and it has been shown that this may be resolved with appropriately modified displacement functions for nodes on restrained boundaries. Once again, the simplest solution to this problem is the enforcement of boundary nodes with zero-order displacement functions.

8 Conclusions and future perspectives

"*The only thing I know is that I know nothing*"

Socrates

8.1 Conclusions

The computational description of quasi-brittle failure, characterised by material degradation, fracturing and potential interaction of fragmented parts, has presented significant challenges to the mechanics community over the past few decades. Efforts to resolve this behaviour numerically have been driven by increasing technological developments, social and economical constraints for safer and more complicated designs and consequently by increasing requirements for more accurate understanding of macro- and micro-structural processes.

Finite element methods have been pushed to their limits in an attempt to resolve strain localisation and ultimately fracturing in a unified and objective manner, while discrete methods have been utilised by artificial connection of discrete bodies which are identified a priori to act as a continuum. Neither of these attempts comprises a *diritta via* for modelling the transition from continuum to discontinuum efficiently and this has led to the investigation of alternative techniques.

Here, the Numerical Manifold Method was investigated as a potentially unifying framework for modelling continua and discontinua, alternative to industry-established techniques, such as FEM and DEM. One of the particularly interesting aspects of NMM is with respect to its potential for modelling efficiently the entire transition between continuum to discontinuum, in a continuum setting, without remeshing.

The work investigated and uniquely extended NMM primarily with respect to the following characteristics:

1. The approximation can be improved globally or locally, for any arbitrary level, without remeshing. This is achieved by hierarchically increasing the order of displacement polynomials which define the trial field. The process has been fully automated algorithmically for any arbitrary level of global and local enhancement.

2. Discontinuities, such as cracks, can be introduced naturally into the numerical domain, in a discrete manner but in a continuum setting, without the need for remeshing, without *a priori* assumptions or the use of interface elements. This is achieved by exploiting the partition of unity property of NMM weight functions.

3. Integration is undertaken explicitly, for any arbitrary level of local improvement of the approximation. This is achieved using simplex elements and simplex integration. The integration process has been implemented, coupled and fully automated algorithmically for any arbitrary level of global or local enrichment of the approximation.

4. Essential boundary conditions are enforced in an elegant manner using projection matrices as an alternative to the Lagrange multiplier method or penalty method. The technique avoids problems such as increasing

the dimensions of the system matrix, loss of the positive definite system or ill-conditioning of the stiffness matrix due to penalty terms.

NMM was reformulated using a constrained variational approach for generalised three-dimensional problems and recast in a form familiar to traditional and extended finite element techniques. Potential issues associated with higher-order approximations, such as conditioning and the enforcement of essential boundary conditions and loads were investigated and potential treatments were proposed. Local higher-order enrichment was implemented within the context of a *p*-adaptive strategy driven by simple error indicators.

Furthermore, throughout the manuscript parallels were identified between NMM and other related techniques (FEM, XFEM, FEM with hierarchical shape functions). Differences between NMM and traditional FEM, hierarchical FEM and extended finite element techniques are subtle, yet distinct and mutually preserving: the use of simplex elements, the use of simplex integration, the use of hierarchical displacement polynomials and the introduction of discontinuities via partition of unity concepts without special enrichments.

For example, the ability to introduce discrete discontinuities without remeshing does not dilute the abilities to enhance the approximation to any arbitrary level without remeshing and undertake integration explicitly, for any arbitrary level of global or local enhancement. Similarly, the use of global or local hierarchical enhancement does not alter the ability to introduce discontinuities using the partition of unity, or to undertake integration analytically.

In contrast, the similarities between NMM and the aforementioned techniques are so striking that it can be postulated that the modelling framework described here is in essence one in which FEM, hierarchical FEM and partition of unity concepts have been intertwined from the early stages of its conceptual birth. As a result, there are merits in investigating NMM as a framework for solving a more general class of problems than it has been originally intended.

8.2 Future perspectives

NMM indicates potential for use as an alternative modelling framework for the analysis of localisation and failure of a large class of materials and problems. However, in comparison to techniques such as FEM, XFEM and DEM, NMM is still at its infancy in terms of development, application and validation.

There are five principal avenues for further research:

1. Unification of the continuous-discontinuous and higher-order aspects of NMM. This has been postulated in this manuscript but it has not been demonstrated.

2. Development of a strategy for explicit integration of element matrices for nonlinear constitutive behaviour. Although the explicit integration strategy presented here applies to the situation where discontinuities are introduced via partition of unity concepts, its generalised application

may be impeded when constitutive behaviour of the continuum is described by nonlinear relationships.

3. Development of the discontinuous modelling approach described here to account for more realistic phenomena, such as crack branching and curved cracks.

4. Implementation and validation of the three-dimensional extensions of higher-order NMM and discontinuous modelling methodologies presented in this manuscript.

5. Further investigations in order to understand and resolve potential conditioning issues observed with the use of hierarchical higher-order shape functions and investigation of the effect of mesh regularity on convergence.

Higher-order approximation, local enhancement strategies and simplex integration have all been developed with the idea of extending them to three-dimensional domains. Conceptually, the introduction of displacement discontinuities in three-dimensions is possible but there are a number of significant algorithmic difficulties that would need to be overcome. For example, tracking of discontinuities in three dimensions represents a significant geometrical challenge, although recent developments in the area of level sets [77] may be of use.

In conjunction with other projects (HYDRO-DDA [45], MAECENAS [67]), in which the modelling of multi-phase flow in discrete cracks is an important consideration, it is recognised that the coupling of NMM with a separate mathematical flow description would represent a significant step forward. Furthermore, the modelling of coupled hygro-mechanical problems using a single mathematical mesh represents an alternative possibility, as long as the mesh in the vicinity of the fracture is sufficiently fine to capture the resulting localised flow to the required accuracy.

Furthermore, the explicit integration strategy developed currently requires a constitutive law that is linear and as yet a methodology for extending this technique for nonlinear material behaviour such as plasticity or damage has still to be undertaken.

It is worthwhile to note that there is strong potential to integrate aspects of the NMM framework described here, such as the modelling of discontinuities using partition of unity concepts, or the higher-order local enhancement using hierarchically increasing displacement polynomials, to FEM, XFEM or DDA. Similarly, the extensive amount of research undertaken with regard to modelling curved or branched cracks and tracking discontinuities in XFEM, or the detection of contact and enforcement of contact constraints in DDA may be potentially utilised for further developments of NMM.

The developments in the Numerical Manifold Method presented here illustrate that the technique could provide a credible alternative modelling framework for analysts in both research and industry. For civil and structural engineers working with quasi-brittle materials this technique could provide the necessary tool for accurately predicting failure mechanisms and collapse loads. For the analysis of jointed material such as rock and masonry, the technique provides a natural modelling framework without the need for a complex mesh that explicitly represents discontinuities, and it also offers the attractive capability to adapt the approximation locally and efficiently without altering the initial discretisation.

9 A. Integration

This section illustrates the transformation and analytical derivation of the integral of a higher order monomial over the area of a simplex in $\mathbb{R}^2$. A similar approach can be followed to derive monomial simplex integrals in $\mathbb{R}^3$, or potentially integrals of other functions. For additional information reference can be made to Chapter 5 and references 22 and 95.

The classical integration problem in NMM is:

$$\iint_{A_g} x^{n_1} y^{n_2} dx dy \qquad (9.1)$$

where $x^{n_1} y^{n_2}$ is a monomial of a general higher-order displacement function, x and y are spatial components and n_1 and n_2 are non-negative integers.

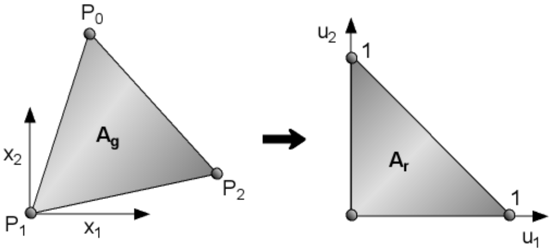


Figure 9-1. Example transformation of a simplex in $\mathbb{R}^2$ from general to regular coordinates

The analytical solution of a polynomial term for the case of regular triangles is [40, 95]:

$$\iint_{A_r} u_0^{i_0} u_1^{i_1} u_2^{i_2} du_1 du_2 = \frac{i_0!\, i_1!\, i_2!}{\left(i_0 + i_1 + i_2 + 2\right)!} \qquad (9.2)$$

where i_0, i_1 and i_2 are non-negative integers.

Adopting $i_0 = 0$ to eliminate the coefficient, the kernel of Equation (9.2) adopts the form of a typical higher-order monomial $x^{n_1} y^{n_2}$:

$$\iint_{A_r} u_1^{i_1} u_2^{i_2} du_1 du_2 = \frac{i_1 ! \, i_2 !}{(i_1 + i_2 + 2)!} \tag{9.3}$$

In order to compute Equation (9.3) for a simplex in a general coordinate system, it is necessary to undertake coordinate transformation. In the general coordinate system, coordinates x and y can be expressed as:

$$x = x_0 u_0 + x_1 u_1 + x_2 u_2 \tag{9.4}$$

$$y = y_0 u_0 + y_1 u_1 + y_2 u_2 \tag{9.5}$$

where $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ are the coordinates of points (nodes) P_1, P_2 and P_3 respectively in the general coordinate system.

Inserting Equations (9.4) and (9.5) in (9.3) yields:

$$\iint_{A_g} x^{n_1} y^{n_2} dx dy = \iint_{A_r} \left(\sum_{i=0}^{2} (x_i u_i) \right)^{n_1} \left(\sum_{i=0}^{2} (y_i u_i) \right)^{n_2} |J| du_1 du_2 \tag{9.6}$$

where J is the Jacobian:

$$J = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix} \tag{9.7}$$

For simplicity, it can be assumed that $(x_0, y_0) = (0,0)$. In this case, using Equations (9.3) and (9.6), the following expansion can be obtained:

$$(x)^{n_1} = (x_1 u_1 + x_2 u_2)^{n_1} = \sum_{k_1=0}^{n_1} \frac{n_1 !}{k_1 ! (n_1 - k_1)!} x_1^{n_1 - k_1} x_2^{k_1} u_1^{n_1 - k_1} u_2^{k_1} \tag{9.8}$$

Similarly:

$$(y)^{n_2} = (y_1 u_1 + y_2 u_2)^{n_2} = \sum_{k_2=0}^{n_2} \frac{n_2 !}{k_2 ! (n_2 - k_2)!} y_1^{n_2 - k_2} y_2^{k_2} u_1^{n_2 - k_2} u_2^{k_2} \tag{9.9}$$

Substituting Equations (9.8) and (9.9) in (9.6) yields the desired integral solution:

$$\iint_{A_g} x^{n_1} y^{n_2} dxdy = |J| \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} \eta(n_1, n_2, k_1, k_2) x^{n_1-k_1} x^{k_1} y^{n_2-k_2} y^{k_2} \qquad (9.10)$$

where:

$$\begin{aligned} \eta(n_1, n_2, k_1, k_2) = & \frac{n_1!\, n_2!}{(n_1+n_2+2)!} \times \\ & \times \frac{(n_1+n_2-k_1-k_2)!\,(k_1+k_2)!}{k_1!\, k_2!\,(n_1-k_1)!\,(n_2-k_2)!} \end{aligned} \qquad (9.11)$$

Chen and Ohnishi [22] have shown that integration using the procedure described here yields accurate results for high orders of n_1 and n_2, and this has been verified by the author.

10 B. Detection of discontinuities

The following code implemented in MATLAB verifies whether two-dimensional simplex elements are intersected by a specified discontinuity using the signed area of the triangle defined by two points of the discontinuity and a query point which can be the vertex of a simplex. If element intersection is detected, then the supports of that element can be duplicated.

```
dx=1; dy=1; nnx=5; nny=4; % trial mesh
dis=[2.6 0; 2.55 0.55; 2.4 1; 2.28 1.28; 2 1.7; 1.85 1.85; 1.7 2]; % coordinates of discontinuity
x=0; y=0; k=0; yc=0;
% Nodal coordinates stored in vectors x & y: standard nodes
for i=1:nny
   xc=0;
   for j=1:nnx
      k=k+1;
      x(k)=xc; y(k)=yc; xc=xc+dx;
   end
   yc=yc+dy;
end
nnodes=k;
% Element topology stored in array top
k=1; top=[];
for i=1:nny-1
   n1=nnx*(i-1)+1; n2=n1+1; n3=nnx*i+2;
   for j=1:nnx-1
      top(k,1:3)=[n1 n2 n3];
      k=k+1;
      top(k,1:3)=[n1 n3 n3-1];
      k=k+1;
      n1=n1+1; n2=n2+1; n3=n3+1;
   end
```

```
end
% Nodal dof stored in nod_dof
nel=k-1; nod_dof=[];
for i=1:nnodes
  nod_dof(i,:)=[i*2-1 i*2];
end
% Element dof stored in edof
edof=[];
for i=1:nel
    edof(i,:)=[i nod_dof(top(i,1),:) nod_dof(top(i,2),:) nod_dof(top(i,3),:)];
end
[d1,d2]=size(dis);
elcrack=0; segcrack=0; SA=0; SA2=0; SA3=0;
k=0; seg=0;
tol=10.^-6; % tolerance
% Loop elements
for i=1:nel
    % Element nodal coordinates
    xxe(1)=x(top(i,1)); xxe(2)=x(top(i,2)); xxe(3)=x(top(i,3)); xxe(4)=x(top(i,1));
    yye(1)=y(top(i,1)); yye(2)=y(top(i,2)); yye(3)=y(top(i,3)); yye(4)=y(top(i,1));
    % Loop discontinuity points
    for di=1:d1-1
        seg=di;
        x1=dis(di,1); y1=dis(di,2);
        x2=dis(di+1,1); y2=dis(di+1,2);
        % Loop element nodes and check their orientation with respect to
        % the discontinuity geometry
        % SignA values (for each node) are stored in 1x3 array SA
        % SignA is twice the signed area of the triangle defined by (xx,yy);
        % (x1,y1); (x2,y2); where (xx,yy) are the query point coordinates and
        % (x1,y1); (x2,y2) the coordinates of the discontinuity
        for j=1:3
    xx=x(top(i,j)); yy=y(top(i,j));
    SignA=(x1-xx)*(y2-yy)-(x2-xx)*(y1-yy);
    SA(j)=SignA;
    if SA(j)>=-tol & SA(j)<=tol
        SA(j)=0;
    end
end
% Check if the discontinuity projection passes through the element
% numbers of intersected elements are stored in vector 'elcrack'
% ‘Crack’ segments passing through the corresponding elements of
% 'elcrack' are stored in 'segcrack'
if elcrack(1,:)~=i
    % the above line simply ensures that the element has not
    % already been classified as cracked
    if sign(SA(1))==sign(SA(2))
        if sign(SA(2))==sign(SA(3))
        elseif sign(SA(2))~=sign(SA(3))
            % Check if the discontinuity intesects the element
            for jj=1:3
                SignA=(xxe(jj)-x1)*(yye(jj+1)-y1)-(xxe(jj+1)-x1)*(yye(jj)-y1);
                SA2(jj)=SignA;
            end
            for jj=1:3
                SignA=(xxe(jj)-x2)*(yye(jj+1)-y2)-(xxe(jj+1)-x2)*(yye(jj)-y2);
                SA3(jj)=SignA;
            end
            % if sign(SA2)=sign(SA3) then there is no intersection
            % but if sign(SA2)~=sign(SA3) we also have to have
            % two terms of each of sign(SA2) & sign(SA3) equal
            % and there has to be at least one non-zero term in
            % each of sign(SA2), sign(SA3); this means that the
            % crack segment goes through two element edges. If
            % sign(SA2) is equal to sign(SA3) then the crack
    % segment is parallel to one of the element edges
    % but without intersecting the element
    for jj=1:3
        if SA2(jj)>=-tol & SA2(jj)<=tol
            SA2(jj)=0;
        end
        if SA3(jj)>=-tol & SA3(jj)<=tol
            SA3(jj)=0;
        end
    end
    if isequal(sign(SA2), sign(SA3))==0 && nnz(SA2)~=3 &&
nnz(SA3)~=3
        aa=sign(SA2); bb=sign(SA3);
        posa=find(aa>0); posb=find(bb>0);
        nega=find(aa<0); negb=find(bb<0);
        zera=find(aa==0); zerb=find(bb==0);

        if size(posa,2)==2 && size(posb,2)==2
            k=k+1;
            elcrack(k)=i;
            segcrack(k)=seg;
        elseif size(posa,2)==2 && size(negb,2)==2
            k=k+1;
            elcrack(k)=i;
            segcrack(k)=seg;
        elseif size(posa,2)==2 && size(zerb,2)==2
            k=k+1;
            elcrack(k)=i;
            segcrack(k)=seg;
        elseif size(nega,2)==2 && size(posb,2)==2
            k=k+1;
            elcrack(k)=i;
            segcrack(k)=seg;
            elseif size(nega,2)==2 && size(negb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(nega,2)==2 && size(zerb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(zera,2)==2 && size(posb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(zera,2)==2 && size(negb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(zera,2)==2 && size(zerb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            end
        end
    end
elseif sign(SA(1))~=sign(SA(2))
    for jj=1:3
        SignA=(xxe(jj)-x1)*(yye(jj+1)-y1)-(xxe(jj+1)-x1)*(yye(jj)-y1);
        SA2(jj)=SignA;
    end
    for jj=1:3
        SignA=(xxe(jj)-x2)*(yye(jj+1)-y2)-(xxe(jj+1)-x2)*(yye(jj)-y2);
        SA3(jj)=SignA;
    end
    for jj=1:3
```

```
            if SA2(jj)>=-tol & SA2(jj)<=tol
                SA2(jj)=0;
            end
            if SA3(jj)>=-tol & SA3(jj)<=tol
                SA3(jj)=0;
            end
        end
        % if sign(SA2)=sign(SA3) then there is no
        % intersection
        if isequal(sign(SA2), sign(SA3))==0 && nnz(SA2)~=3 &&
nnz(SA3)~=3
            aa=sign(SA2); bb=sign(SA3);
            posa=find(aa>0); posb=find(bb>0);
            nega=find(aa<0); negb=find(bb<0);
            zera=find(aa==0); zerb=find(bb==0);

            if size(posa,2)==2 && size(posb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(posa,2)==2 && size(negb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(posa,2)==2 && size(zerb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(nega,2)==2 && size(posb,2)==2
                k=k+1;
                elcrack(k)=i;
                segcrack(k)=seg;
            elseif size(nega,2)==2 && size(negb,2)==2
```

```
                        k=k+1;
                        elcrack(k)=i;
                        segcrack(k)=seg;
                    elseif size(nega,2)==2 && size(zerb,2)==2
                        k=k+1;
                        elcrack(k)=i;
                        segcrack(k)=seg;
                    elseif size(zera,2)==2 && size(posb,2)==2
                        k=k+1;
                        elcrack(k)=i;
                        segcrack(k)=seg;
                    elseif size(zera,2)==2 && size(negb,2)==2
                        k=k+1;
                        elcrack(k)=i;
                        segcrack(k)=seg;
                    elseif size(zera,2)==2 && size(zerb,2)==2
                        k=k+1;
                        elcrack(k)=i;
                        segcrack(k)=seg;
                    end
                end
            end
        end
    end
end
```

11 C. Nonlinear solution procedure for discontinuous fracturing

This section provides an algorithmic account of the nonlinear solution procedure for cohesive crack problems using displacement control. The following is an example of a pseudo-code that can be implemented in MATLAB.

```
- Resolve which elements are intersected by the discontinuity
kappa=zeros(1,nel); kappa_t=kappa; % 'nel' is the total number of elements
- Apply displacement control
    % loop over all elements
    for i=1:nel
        - if the element is intersected by the discontinuity:
            - calculate stress and strain vectors
            - determine coordinates of displacement discontinuity
            - define matrix 'C' for transforming local to global
            - define displacement jump 'u_jump' at centre of discontinuity in the global coordinate system
            - Transform displacement jump 'u_jump' at centre of discontinuity in the local coord system
            if abs(u_jump(1))>kappa_t(i); kappa(i)=u_jump(1); end
            if the element contains the crack tip:
                t=zeros(2,1); d=0;
            else
                t(1,1)=ft*exp(-ft/Gf*kappa(i));  t(2,1)=0; t=C*t;
                d=- ft^2/Gf*exp(-ft/Gf*kappa(i));
                % Gf is the fracture energy release rate, whereas ft is the tensile
                % strength
            end
            - Calculate traction matrix in local and global coord system
            - Calculate traction stiffness matrix
            - Calculate tangent stiffness matrix
            - Assemble element contributions to the system
          elseif: if the element is not intersected by the discontinuity
            - Assemble element contributions to the system
        end
      end
    Solve
    - Determine elements in vicinity of cracktip
    - Calculate stress ahead of crack tip
    - Resolve which elements are intersected by the discontinuity
    - Modify new nodes for crack tip
    - Modify elements for fracture
    - Update displacement degrees of freedom
- Repeat
```

12 D. Sub-matrix method

This section describes the sub-matrix method for computing the stiffness matrix of general higher-order systems. The approach is based on the work of Lu [65] but it is extended for three-dimensions and problems in which nodal displacement functions are not necessarily of the same order.

Consider the strain vector for a general solid element with n number of nodes:

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_n \end{pmatrix} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{pmatrix} \tag{12.1}$$

where $\mathbf{B}_i$ is a sub-matrix of the strain interpolation matrix $\mathbf{B}$ with dimensions $r \times p\left(N_i + 1\right)$, for r strain components. $\mathbf{B}$ is defined as (section 3.6):

$$\mathbf{B} = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{pmatrix} \mathbf{T} \tag{12.2}$$

Therefore, for the three-dimensional case the strain interpolation sub-matrix can be expressed as:

$$
\mathbf{B}_i = \left[\begin{array}{ccccc}
\dfrac{\partial w_i}{\partial x_1} & 0 & 0 & \dfrac{\partial\left(w_i x_1\right)}{\partial x_1} & \cdots \\
0 & \dfrac{\partial w_i}{\partial x_2} & 0 & 0 & \cdots \\
0 & 0 & \dfrac{\partial w_i}{\partial x_3} & 0 & \cdots \\
\dfrac{\partial w_i}{\partial x_2} & \dfrac{\partial w_i}{\partial x_1} & 0 & \dfrac{\partial\left(w_i x_1\right)}{\partial x_2} & \cdots \\
0 & \dfrac{\partial w_i}{\partial x_3} & \dfrac{\partial w_i}{\partial x_2} & 0 & \cdots \\
\dfrac{\partial w_i}{\partial x_3} & 0 & \dfrac{\partial w_i}{\partial x_1} & \dfrac{\partial\left(w_i x_1\right)}{\partial x_3} & \cdots
\end{array}\right.
$$

$$
\left.\begin{array}{cccc}
\cdots & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_1} & 0 & 0 \\
\cdots & 0 & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_2} & 0 \\
\cdots & 0 & 0 & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_3} \\
\cdots & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_2} & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_1} & 0 \\
\cdots & 0 & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_3} & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_2} \\
\cdots & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_3} & 0 & \dfrac{\partial\left(w_i x_3^{N_i}\right)}{\partial x_1}
\end{array}\right] \tag{12.3}
$$

From (12.3) it can be observed that:

$$
\mathbf{B}_i = \left[\mathbf{B}_{i1} \quad \mathbf{B}_{i2} \quad \cdots \quad \mathbf{B}_{im}\right] \tag{12.4}
$$

where m is the number of polynomial terms of the displacement functions (section 3.4), and any term $\mathbf{B}_{ik}$, $k = 1...m$, is of the general form:

$$\mathbf{B}_{ik}=\begin{bmatrix} \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_1} & 0 & 0 \\ 0 & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_2} & 0 \\ 0 & 0 & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_3} \\ \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_2} & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_1} & 0 \\ 0 & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_3} & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_2} \\ \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_3} & 0 & \dfrac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_1} \end{bmatrix} \tag{12.5}$$

where $n_1, n_2, n_3 = 0 \ldots N_i$.

Using the definition of the weight function (section 3.3), the general term $w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}$ can be expanded as following:

$$\begin{aligned} w_i x_1^{n_1}x_2^{n_2}x_3^{n_3} &= \left(a_i + b_i x_1 + c_i x_2 + d_i x_3\right) x_1^{n_1}x_2^{n_2}x_3^{n_3} \\ &= a_i x_1^{n_1}x_2^{n_2}x_3^{n_3} + b_i x_1^{n_1+1}x_2^{n_2}x_3^{n_3} + c_i x_1^{n_1}x_2^{n_2+1}x_3^{n_3} + d_i x_1^{n_1}x_2^{n_2}x_3^{n_3+1} \end{aligned} \tag{12.6}$$

Therefore:

$$\begin{aligned} \frac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_1} &= a_i n_1 x_1^{n_1-1}x_2^{n_2}x_3^{n_3} + b_i\left(n_1+1\right)x_1^{n_1}x_2^{n_2}x_3^{n_3} + \\ &\quad + c_i n_1 x_1^{n_1-1}x_2^{n_2+1}x_3^{n_3} + d_i n_1 x_1^{n_1-1}x_2^{n_2}x_3^{n_3+1} \end{aligned} \tag{12.7}$$

And similarly:

$$\begin{aligned} \frac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_2} &= a_i n_2 x_1^{n_1}x_2^{n_2-1}x_3^{n_3} + b_i n_2 x_1^{n_1+1}x_2^{n_2-1}x_3^{n_3} + \\ &\quad + c_i\left(n_2+1\right)x_1^{n_1}x_2^{n_2}x_3^{n_3} + d_i n_2 x_1^{n_1}x_2^{n_2-1}x_3^{n_3+1} \end{aligned} \tag{12.8}$$

$$\begin{aligned} \frac{\partial\left(w_i x_1^{n_1}x_2^{n_2}x_3^{n_3}\right)}{\partial x_3} &= a_i n_3 x_1^{n_1}x_2^{n_2}x_3^{n_3-1} + b_i n_3 x_1^{n_1+1}x_2^{n_2}x_3^{n_3-1} + \\ &\quad + c_i n_3 x_1^{n_1}x_2^{n_2+1}x_3^{n_3-1} + d_i\left(n_3+1\right)x_1^{n_1}x_2^{n_2}x_3^{n_3} \end{aligned} \tag{12.9}$$

The discretised system can be expressed in a general form as:

$$\left[\begin{array}{cccc:cc}
\mathbf{K}_{11,11} & \mathbf{K}_{11,12} & \cdots & \mathbf{K}_{11,1N_1} & \mathbf{K}_{11,21} & \cdots \\
\mathbf{K}_{12,11} & \mathbf{K}_{12,12} & \cdots & \mathbf{K}_{12,1N_1} & \mathbf{K}_{12,21} & \cdots \\
\vdots & \vdots & \ldots & \vdots & \vdots & \ldots \\
\mathbf{K}_{1N_1,11} & \mathbf{K}_{1N_1,12} & \cdots & \mathbf{K}_{1N_1,1N_1} & \mathbf{K}_{1N_1,21} & \cdots \\
\hdashline
\mathbf{K}_{21,11} & \mathbf{K}_{21,12} & \cdots & \mathbf{K}_{21,1N_2} & \mathbf{K}_{21,21} & \cdots \\
\vdots & \vdots & \ldots & \vdots & \vdots & \ldots
\end{array}\right]
\left[\begin{array}{c}
\mathbf{u}_{11} \\ \mathbf{u}_{12} \\ \vdots \\ \mathbf{u}_{1N_1} \\ \hdashline \mathbf{u}_{21} \\ \vdots
\end{array}\right]
=
\left[\begin{array}{c}
\mathbf{f}_{11} \\ \mathbf{f}_{12} \\ \vdots \\ \mathbf{f}_{1N_1} \\ \hdashline \mathbf{f}_{21} \\ \vdots
\end{array}\right]
\qquad (12.10)$$

with:

$$\mathbf{K}_{ij,kl} = \int_e \mathbf{B}_{ij}^T \mathbf{E} \mathbf{B}_{kl}^T \, dx_1 dx_2 dx_3 \qquad (12.11)$$

where j corresponds to the j^{th} degree of freedom of node i, whereas l corresponds to the l^{th} degree of freedom of node k. From equations (12.7), (12.8) and (12.9), it can be observed that it is possible to derive explicit expressions of the stiffness sub-matrices $\mathbf{K}_{ij,kl}$. Therefore, $\mathbf{K}_{ij,kl}$ can be alternatively expressed as:

$$\mathbf{K}_{ij,kl} = \begin{bmatrix}
\int_e k_{ij,kl}^{1,1} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{1,2} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{1,3} \, dx_1 dx_2 dx_3 \\
\int_e k_{ij,kl}^{2,1} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{2,2} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{2,3} \, dx_1 dx_2 dx_3 \\
\int_e k_{ij,kl}^{3,1} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{3,2} \, dx_1 dx_2 dx_3 & \int_e k_{ij,kl}^{3,3} \, dx_1 dx_2 dx_3
\end{bmatrix} \qquad (12.12)$$

All terms $\int_e k_{ij,kl} dx_1 dx_2 dx_3$ of Equation (12.12) are typically derived explicitly and hard-coded. Integration can be undertaken analytically using the simplex integration strategy discussed in section 6.3. The derivation of the terms of Equation (12.12) is laborious to present here. Indicatively, for the two-dimensional case each term $\int_e k_{ij,kl} dx_1 dx_2 dx_3$ is the sum of sixteen integrated terms.

The global stiffness matrix can be assembled using an algorithm similar to the following:

```
for element=1:(to total number of elements)
    for i=1:(number of element covers)      % cover i loop
        for mi=1:(number of polynomial terms of cover i)
            for j=1:(number of element covers)      % cover j loop
                for nj=1:(number of polynomial terms of cover j)
                    evaluate $\mathbf{K}_{ij,kl}$
                    add $\mathbf{K}_{ij,kl}$ in $\mathbf{K}$
                end
            end
        end
    end
end
```

Figure 12-1. Element-by-element assembly of the global stiffness matrix from sub-matrices $\mathbf{K}_{ij,kl}$.

It can be shown that the location of sub-matrix $\mathbf{K}_{ij,kl}$ in the global stiffness matrix is:

$$Row = 2\left(\left(i-1\right)m_i + j\right) - 1 \tag{12.13}$$

$$Column = 2\left(\left(k-1\right)m_k + l\right) - 1 \tag{12.14}$$

where m_i and m_k are the numbers of polynomial terms of the displacement functions associated with nodes i and k respectively (see section 3.4).

Equations (12.13) and (12.14) are valid provided that all displacement functions are of the same order, i.e. $N_1 = N_2 = ...N_n$. If this is not the case, then an iterative approach is required to determine the location of each sub-matrix in the global system. This has been illustrated in section 4.7.

13 E. Multi-dimensional matrix method

Chapter 4 advocated that the terms of sub-matrices $\mathbf{B}_i$ can be stored in multi-dimensional arrays $\overline{\mathbf{B}}_i$, which can then be used to compute the stiffness matrix without explicitly deriving any of its parts. Each matrix $\overline{\mathbf{B}}_i$ has dimensions $r \times p\left(N_i+1\right)\times\left(N_i+1\right)\times\left(N_i+1\right)\times\left(N_i+1\right)$, where r is the number of strain components considered, and the position of each term of $\overline{\mathbf{B}}_i$ determines the exponents of $\mathbf{x}$.

The first two dimensions of $\overline{\mathbf{B}}_i$ correspond to the location of terms of $\mathbf{B}_i$ whereas the third dimension of $\overline{\mathbf{B}}_i$ corresponds to the exponents of x_1 terms, the fourth dimension corresponds to the exponents of x_2 terms and so on. For example, for a two-dimensional problem ($p=2$) with three strain components ($r=3$), $\overline{\mathbf{B}}_i$ will be $3\times 2\left(N_i+1\right)\times\left(N_i+1\right)\times\left(N_i+1\right)$.

As a further example, from Equations (12.3) and (12.7) given in Appendix D, term $\left(1,1\right)$ of $\mathbf{B}_i$ in a three-dimensional case is:

$$\begin{aligned}\frac{\partial\left(w_i x_1^{n_1} x_2^{n_2} x_3^{n_3}\right)}{\partial x_1} &= a_i n_1 x_1^{n_1-1} x_2^{n_2} x_3^{n_3} + b_i\left(n_1+1\right) x_1^{n_1} x_2^{n_2} x_3^{n_3} + \\ &\quad + c_i n_1 x_1^{n_1-1} x_2^{n_2+1} x_3^{n_3} + d_i n_1 x_1^{n_1-1} x_2^{n_2} x_3^{n_3+1}\end{aligned} \tag{13.1}$$

Therefore, term $a_i n_1$ can be stored in location $\left(1,1,n_1-1,n_2,n_3\right)$ of $\overline{\mathbf{B}}_i$, whereas term $b_i\left(n_1+1\right)$ can be stored in location $\left(1,1,n_1,n_2,n_3\right)$ and so on.

Thus, the element stiffness sub-matrix $\mathbf{K}_{ij}$ associated with nodes i and j can be generally expressed as:

$$\begin{aligned}\mathbf{K}_{ij} &= \int_V \mathbf{B}_i^T \mathbf{E} \mathbf{B}_j \, dV \\ &= \sum_{q_1=1}^{(N+1)} \sum_{r_1=1}^{(N+1)} \sum_{s_1=1}^{(N+1)} \sum_{q_2=1}^{(N+1)} \sum_{r_2=1}^{(N+1)} \sum_{s_2=1}^{(N+1)} \\ &\quad \overline{\mathbf{B}}_i^T\left(:,:,q_1,r_1,s_1\right) \\ &\quad \mathbf{E}\overline{\mathbf{B}}_j\left(:,:,q_2,r_2,s_2\right) S\left(x_1^{q_1+q_2\text{-}2}, x_2^{r_1+r_2\text{-}2}, x_3^{s_1+s_2\text{-}2}\right)\end{aligned} \tag{13.2}$$

The term $S\left(x_1^{q_1+q_2-2}, x_2^{r_1+r_2-2}, x_3^{s_1+s_2-2}\right)$ in equation (13.2) is the explicit integral of a general polyhedral (simplex) volume as given in Chapter 6.

In order to further illustrate the concept consider a one-dimensional ($p=1$), two-node ($n=2$) axial bar element defined by nodes i and j, with first-order displacement functions ($N_i = N_j = 1$). Due to the one-dimensional nature of the problem, the weight function simplifies to:

$$w_i = a_i + b_i x \tag{13.3}$$

Therefore, the strain interpolation sub-matrix $\mathbf{B}_i$ is:

$$\mathbf{B}_i = \begin{pmatrix} \dfrac{\partial w_i}{\partial x} & \dfrac{\partial (x w_i)}{\partial x} \end{pmatrix} = \begin{pmatrix} b_i & a_i + 2b_i x \end{pmatrix} \tag{13.4}$$

The multi-dimensional sub-matrix $\overline{\mathbf{B}}_i$ has dimensions $1 \times 2 \times 2$ and is defined as:

$$\begin{aligned} \overline{\mathbf{B}}_i(1,:,1) &= \begin{pmatrix} b_i & a_i \end{pmatrix} \\ \overline{\mathbf{B}}_i(1,:,2) &= \begin{pmatrix} 0 & 2b_i \end{pmatrix} \end{aligned} \tag{13.5}$$

$\overline{\mathbf{B}}_i(1,:,1)$ represents those coefficients of $\mathbf{B}_i$ that are multiplied with x^0, whereas $\overline{\mathbf{B}}_i(1,:,2)$ represents those coefficients of $\mathbf{B}_i$ that are multiplied by x^1.

The product of coefficient sub-matrices is an operation unavailable in programming libraries. An algorithmical approach for deriving such multi-dimensional products is discussed in the following section.

14 F. Convolution of multi-dimensional coefficient arrays

In order to compute the stiffness sub-matrices $\mathbf{K}_{ij}^{e}$ in the multi-dimensional matrix approach (Appendix E), it is necessary to multiply multi-dimensional coefficient arrays of strain interpolation sub-matrices. This section reviews the logic behind the implementation of this operation.

In order to multiply the coefficient arrays of Equation (13.2), consider the following problem:

$$\mathbf{A}\cdot\mathbf{B}=\mathbf{C} \tag{14.1}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are n-order multi-dimensional arrays. In order to simplify the problem, let us assume that we have fourth-order arrays as it would be the case for a two-dimensional problem.

As discussed above, $\mathbf{C}(i,j,m_1,m_2)$ will be assembled from coefficients which are multiplied by $x_1^{m_1}x_2^{m_2}$ monomials in arrays $\mathbf{A}$ and $\mathbf{B}$, which are strain interpolation coefficient arrays.

Therefore, $\mathbf{C}(i,j,m_1,m_2)$ can be computed by summing $\mathbf{A}(i,j,n_1,n_2)$ and $\mathbf{B}(i,j,n_3,n_4)$ for all n_1,n_2,n_3,n_4, where n_1,n_2,n_3,n_4 are such that:

$$x_1^{n_1}x_2^{n_2}\cdot x_1^{n_3}x_2^{n_4}=x_1^{m_1}x_2^{m_2} \tag{14.2}$$

This means that the individual elements of the product of arrays $\mathbf{A}$ and $\mathbf{B}$ can be computed from:

$$\mathbf{C}(:,:,m_1,m_2)=\sum\mathbf{A}(:,:,n_1,n_2)\cdot\mathbf{B}(:,:,n_3,n_4) \tag{14.3}$$

for all n_1,n_2,n_3,n_4 for which:

$$n_1+n_2+n_3+n_4=m_1+m_2 \tag{14.4}$$

Similarly, for the three-dimensional case with fourth-order coefficient arrays it is required that:

$$\mathbf{C}(:,:,m_1,m_2,m_3) = \sum \mathbf{A}(:,:,n_1,n_2,n_3)\cdot\mathbf{B}(:,:,n_4,n_5,n_6) \tag{14.5}$$

for all $n_1, n_2, n_3, n_4, n_5, n_6$ for which:

$$n_1 + n_2 + n_3 + n_4 + n_5 + n_6 = m_1 + m_2 + m_3 \tag{14.6}$$

It is worthwhile to note that all m and n are integers. Furthermore, commutativity and associativity apply.

To illustrate the derivation of the product of two-dimensional coefficient arrays, the first three terms $(:,:,0,0)$, $(:,:,1,0)$ and $(:,:,2,0)$ of a fourth-order coefficient array $\mathbf{C}$ can be computed from:

$$\mathbf{C}(:,:,0,0) = \mathbf{A}(:,:,0,0)\cdot\mathbf{B}(:,:,0,0) \tag{14.7}$$

$$\begin{aligned}\mathbf{C}(:,:,1,0) &= \mathbf{A}(:,:,0,0)\cdot\mathbf{B}(:,:,1,0)\\ &+ \mathbf{A}(:,:,1,0)\cdot\mathbf{B}(:,:,0,0)\end{aligned} \tag{14.8}$$

$$\begin{aligned}\mathbf{C}(:,:,2,0) &= \mathbf{A}(:,:,0,0)\cdot\mathbf{B}(:,:,2,0)\\ &+ \mathbf{A}(:,:,2,0)\cdot\mathbf{B}(:,:,0,0)\\ &+ \mathbf{A}(:,:,1,0)\cdot\mathbf{B}(:,:,1,0)\end{aligned} \tag{14.9}$$

Element stiffness sub-assemblies $\mathbf{K}_{ij}$ can be used to construct and operate on the global stiffness matrix, or alternatively on element assemblies $\mathbf{K}^e$. The global and element stiffness matrices can be assembled using procedures similar to the sub-matrix approach. Alternatively, the product of $\mathbf{B}^T\mathbf{E}\mathbf{B}$ can be stored in multi-dimensional arrays using the approach of Figure 14-1. $\mathbf{K}^e$ can be assembled using the approach illustrated by the algorithm of Figure 14-2.

```
% Compute $(\mathbf{B}^T\mathbf{E})\mathbf{B}$ and store result in multi-dimensional array KS
% $(\mathbf{B}^T\mathbf{E})$ is stored in matrix BTE

for an1=0:N
    for an2=0:N
            for bn1=0:N
            for bn2=0:N

                TKT(1:6*m,1:3)=BTE(1:6*m,1:3,an1,an2);

                TBT=B(1:3,1:6*m,bn1,bn2);

                KS(1:6*m,1:6*m,an1+bn1,an2+bn2)

                =TKT*TBT;

            end

        end

    end

end
```

Figure 14-1. Two-dimensional convolution of multi-dimensional arrays $\mathbf{B}^T\mathbf{E}$ and $\mathbf{B}$

```
% Assemble the element stiffness matrix by adding all terms (:,:,n1,n2) of
% KS

for i=1:6*m

    for j=1:6*m

        for n1=0:2*N

            for n2=0:2*N

                KE(i,j)=KE(i,j)+KS(i,j,n1,n2);

            end

        end

    end

end
```

Figure 14-2. Assembly of two-dimensional element stiffness matrix using a multi-dimensional matrix approach

15 Published papers

The following papers were published and presented as part of this research (given in chronological order):

1. Kourepinis, D., Bićanić, N, Pearce, C. J., 2002. Aspects of the Numerical Manifold Method. In *Proceedings of the 10th National Conference of the British Association for Computational Mechanics in Engineering (ACME)*, Swansea University, Wales
2. Kourepinis, D., Bićanić, N, Pearce, C. J., 2003, A Higher-Order Numerical Manifold Method Simplex Integration Strategy. In *Proceedings of the 11th National Conference of the British Association for Computational Mechanics in Engineering (ACME)*, University of Strathclyde, Scotland
3. Kourepinis, D., Bićanić, N., and Pearce, C. J., 2003. A Higher-Order Variational Numerical Manifold Method Formulation and Simplex Integration strategy. 145-152. *SINTEF. The 6th International Conference on Analysis of Discontinuous Deformation*. Lu, M.
4. Kourepinis, D., Bićanić, N., Pearce, C. J., 2004. The Numerical Manifold Method for Modelling Fracturing in Quasi-Brittle Materials. In *Proceedings of the 12th National Conference of the Association of Computational Methods in Engineering (ACME)*, Cardiff University
5. Bićanić, N., Pearce, C. J., Davie, C., Kourepinis, D., 2005. Some Aspects of Computational Modelling of Safety Critical Concrete Structures. Plenary Lecture. In *Proceedings of the 5th International Congress on Computational Mechanics*, Greek Association of Computational Mechanics, Limassol, Cyprus

16 References

1. Ainsworth, M. 2001. Essential boundary conditions and multi-point constraints in finite element analysis. In *Computer Methods in Applied Mechanics and Engineering*. 190, 6323-6339

2. Anthoine, A., 1995. Derivation of the in-plane elastic characteristics of masonry through homogenization theory. In *International Journal of Solids and Structures*, 32 (2), 137-163

3. Areias, P. M. A., Belytschko, T., 2005. Analysis of three-dimensional crack initiation and propagation using the extended finite element method. In *International Journal for Numerical Methods in Engineering*, in press, published online in Wiley InterScience, DOI: 10.1002/nme.1305

4. Argyris, J., Kelsey, S., 1960. Energy Theorems and Structural Analysis. Butterworth, London

5. Askes, H., Advanced Spatial Discretisation Strategies for Localised Failure, Mesh Adaptivity and Meshless Methods, PhD Thesis, Delft University of Technology, ISBN 90 9013714 9

6. Babuška, I. & Melenk, J.M., 1996. The partition of unity finite element method: Basic theory and applications. In *Computer methods in applied mechanics and engineering*, 139: 289-314

7. Babuška, I. & Melenk, J.M., 1997. The partition of unity method. In *International Journal of Numerical Methods in Engineering*, 40: 727-758

8. Banerjee, P. K., Butterfield, R., 1977. Boundary Element Method in Geomechanics. In Finite Elements in Geomechanics, Chapter 16, G. Gudehus (Editor), Wiley

9. Bathe, K.J. & Wilson, E.L. 1976. Numerical methods in finite element analysis. Englewood Cliffs, N.J: Prentice-Hall

10. Bathe, K. J., 1996, Finite Element Procedures, Prentice-Hall

11. Bažant, Z. P., 1982. Crack Band Model for Fracture of Geomaterials. In *Proceedings of the Fourth International Conference on Numerical Methods in Geomechanics*, 3, 1137-1152

12. Bažant, Z. P., 2002. Concrete fracture models: testing and practice. In *Engineering Fracture Mechanics*, 69, 165-205

13. Belytschko, T., Lu, Y. Y., Gu, L., 1994, Element-free Galerkin methods, in *International Journal for Numerical Methods in Engineering*, 37: 229-256

14. Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P, 1996. Meshless methods: An overview and recent developments. In *Computer methods in applied mechanics and engineering*, 139, 3-47

15. Belytschko, T. & Black, T., 1999. Elastic Crack Growth in Finite Elements with Minimal Remeshing. In *International Journal of Numerical Methods in Engineering*, Vol. 45, 5: 601-620

16. Belytschko T, Moës N, Usui S, Parimi C. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* 2001; 50:993-1013.

17. Bićanić, N., Stirling, C., Pearce, C. J., 2002. Discontinuous Modelling of Structural Masonry. In *Proceedings of Fifth World Congress on Computational Mechanics (WCCM V)*, July 7-12, 2002, Vienna, Austria

18. Bischoff, P. H., Perry, S. H., 1991. Compressive Behaviour of Concrete at High Strain Rates. In *Materials and Structures*, 24, 425-450

19. Brebbia, C., Telles, J. and Wrobel, L. 1984. Boundary Element Techniques. Springer-Verlag. pp. 134

20. British Nuclear Group, Ltd., 1997. Berkeley reactor building (photograph). From British Nuclear Group Image Library: http://magnox.info. Copyright belongs exclusively to British Nuclear Group Ltd.

21. Chen, G., Ohnishi, Y., Ito, T. 1998. Development of high-order manifold method. In *International Journal of Numerical Methods in Engineering*, 43: 685-712, 1998

22. Chen, G. & Ohnishi, Y. 1999. Practical Computing Formulas of Simplex Integration. In *Proceedings of 3rd International Conference on Analysis of Discontinuous Deformation*: 75-84, June 1999

23. Chen, G., Zen, K., Ohnishi, Y., Kosama, K.., Extension of Manifold Method and its Applications. In *Proceedings of the Fourth International Conference on Analysis of Discontinuous Deformation (ICADD-4)*, edited by Nenad Bićanić, pp. 439-450, June 2001, University of Glasgow

24. Chen, Y. Y., 2007. Nonlinear finite element analysis of certain shear failure mechanisms in reinforced concrete beams. Doctorate thesis, Heriot-Watt University

25. Cheng, Y.M., Zhang, Y.H, Chen, W.S. 2002. Wilson non-conforming element in numerical manifold method. In *Communications in Numerical Methods in Engineering*, 18: 877-884

26. Chien, W. Z., 1983, Method of high-order Lagrange multiplier and generalized variational principles of elasticity with more general forms of functionals, *Applied Mathematics and Mechanics*, 4, 137-150

27. Clough, R. W., 1965. Stress Analysis: The Finite Element Method in Structural Mechanics (Chapter 7). O. C. Zienkiewicz and G. S. Holister (Editors), Wiley

28. Clough, R. W, Penzien, J., 1993. Dynamics of Structures, Second Edition, International Editions, McGraw-Hill, ISBN 0 07 113241 4

29. Comité Euro-International du Béton, 1993. CEB-FIP Model Code 1990. Thomas Telford Services Ltd., ISBN 0-7277-1696-4

30. Cundall, P. A., 1971. A computer model for simulating progressive large scale movements in blocky rock systems. In *Proceedings of the Symposium of the International Society of Rock Mechanics*, Volume 1, Nancy, France

31. Daux, C., Moës, N., Dolbow, J., Sukumar, N., Belytschko, T., 2000. Arbitrary branched and intersecting cracks with the extended finite

element method. In *International Journal for Numerical Methods in Engineering*, 48, 1741-1760

32. Davie, C. T., 2002. Particulate Mechanics Framework for Modelling Multi-Physics Processes in Fracturing Geomaterials. Doctorate thesis. University of Glasgow

33. De Borst, R., Mühlhaus, H. B., 1992. Gradient-dependent plasticity – formulation and algorithmic aspects. In *International Journal for Numerical Methods in Engineering*, 35 (3), 521-539

34. De Borst, R., 2001. Some recent issues in computational failure mechanics. In *International Journal for Numerical Methods in Engineering*, 52, 63-95

35. De Borst, R., Gutiérrez, M. A., Wells, G. N., Remmers, J. J. C. and Askes, H., 2004. Cohesive-zone models, higher-order continuum theories and reliability methods for computational failure analysis. In *International Journal for Numerical Methods in Engineering*, 60. 289-315

36. Duarte, C. A., 1995. A review of some meshless methods to solve partial differential equations. *Technical Report 95-06*, TICAM, University of Texas at Austin

37. Duarte, C. A. and Oden, J. T., 1996. H-p Clouds – an h-p Meshless Method. In *Numerical Methods for Partial Differential Equations*, 12, 673-705, John Wiley and Sons, Inc.

38. Fleming, M., Chu, Y. A., Moran, B., Belytschko, T. 1997. Enriched element-free Galerkin methods for crack tip fields. In *International Journal for Numerical Methods in Engineering*, 40, pp. 1483-1504

39. Goldberg, David. What Every Computer Scientist Should Know About Floating-Point Arithmetic. In *Computing Surveys*, Association of Computing Machinery Inc., published in March 1991, reprinted in http://docs.sun.com/source/806-3568/ncg_goldberg.html by permission

40. Hammer, P. C., Stroud, A. H., Numerical integration over simplices, Math. Tables Aids Comput., 10, 137–139, 1956.

41. He, J. H., 2001. Derivation of Generalized Variational Principles without using Lagrange Multipliers, Part II, Applications to Solid Mechanics, *Facta Universitatis, Mechanics, Automatic Control and Robotics*, Vol. 3, 11, 13-20

42. Hildebrand, F. B., 1968. Finite-Difference Equations and Simulations. Prentice-Hall, New Jersey

43. Hillerborg, A., Modeer, M., Petersson, P.E., 1976. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. In *Cement and Concrete Research*, 6, 773–82

44. Hrennikoff, A., 1941. Solutions of problems of elasticity by the framework method. In *Journal of Applied Mechanics*, 12, 169-175

45. HYDRO-DDA, A Hydro-mechanical Modelling Framework Applied to the Sealing Capacity of Fractured Mudrocks, 1999. EPSRC Research Project GR/M62532/01, Principal investigator: Dr. G. D. Couples,

Heriot-Watt University, Project partners: British Gas plc, Schlumberger Cambridge Research Ltd.

46. International Centre for Numerical Methods in Engineering (CIMNE), 2007, GiD pre and post processor. At http://gid.cimne.upc.es/

47. Jirásek, M., 1998. Embedded crack models for concrete fracture. In *Proceedings of Euro-C: Computer Modelling of Concrete Structures*, R. de Borst, N. Bićanić, H. Mang and G. Meschke (Editors), Balkema, Rotterdam, 291-300

48. Jirásek, M., Zimmerman, T., 1998. Analysis of rotating crack model. In *Journal of Engineering Mechanics*, ASCE, 124, 842-851

49. Jirásek, M., Patzák, B., 2001. Models for Quasi-Brittle Failure: Theoretical and Computational Aspects. In *Proceedings of the European Conference on Computational Mechanics*, Cracow, Poland

50. Jirásek, M. and Belytschko, T., 2002. Computational resolution of strong discontinuities. WCCM V - *Fifth World Congress on Computational Mechanics*. Mang, H., Rammerstorfer, F. G., and Eberhardsteiner, J.

51. Jirásek, M., 2004. RILEM TC QFS Quasibrittle fracture scaling and size effect – Final report. In *Materials and Structures*, 37, 547-568

52. Kaczmarczyk, L., Pearce, C. J., Bićanić, N., 2007. Scale transition and enforcement of RVE boundary conditions in second-order computational homogenization. In *International Journal of Numerical Methods in Engineering*, DOI: 10.1002/nme.2188

53. Kotrechko, S. A., Meshkov, Y. Y., 1999. Mechanics and physics of quasi-brittle fracture of polycrystalline metals under conditions of stress concentration. Part 2. Theoretical basis. In *Strength of Materials*, 31(3), 1573-9325

54. Kourepinis, D., Bićanić, N., and Pearce, C. J., 2003. A Higher-Order Variational Numerical Manifold Method Formulation and Simplex Integration strategy. 145-152. *SINTEF. The 6th International Conference on Analysis of Discontinuous Deformation*. Lu, M.

55. Kourepinis, D., Bićanić, N., Pearce, C. J., 2004. The Numerical Manifold Method for Modelling Fracturing in Quasi-Brittle Materials. In *Proceedings of the Association of Computational Methods in Engineering (ACME)*, 5-6 April 2004, Cardiff University

56. Laborde, P., Pommier, J., Renard, Y., Salaun, M. 2005. High-order extended finite element method for cracked domains. In *International Journal for Numerical Methods in Engineering*, 64, pp. 354-381

57. Lancaster, P., Salkauskas, K., 1981. Surfaces generated by moving least-squares methods. In *Mathematics of Computation*, 37, 141-158

58. Li, S., Liu, W. K., 2002. Meshfree and particle methods and their applications. In *Applied Mechanics Review*, 55, 1-34

59. Li, Q., Duan, Y., Wang, G., 2002. Behaviour of large concrete specimens in uniaxial tension. In *Magazine of Concrete Research*, 54 (5), 385-391

60. Lin, J. S., 2003. A mesh-based partition of unity method for discontinuity modelling. In *Computer Methods in Applied Mechanics and Engineering*, 192, 1515-1532

61. Liu, W. K., Jun, S., Zhang, Y. F., 1995. Reproducing kernel particle methods. In *International Journal for Numerical Methods in Engineering*, 20, 1081-1106

62. Liu, W. K., Hao, S., Belytschko, T., Lim S., Chang, C. T., 1999. Multiple scale meshfree methods for damage fracture and localization. In *Computational Material Science*, 16, 197-205

63. Lourenço, P. J. B. B., 1996. Computational Strategies for Masonry Structures. Doctorate thesis, Delft University of Technology, Delft University Press

64. Lu, M., Bostrøm, B., Svanø, G. 2001. Hydraulic Fracturing Simulation with Numerical Manifold Method. In *Proceedings of 4th International Conference on Analysis of Discontinuous Deformation*: 391-400, Glasgow, 6-8 June 2001

65. Lu, M. 2002. Numerical Manifold Method with complete N-order cover function. STF22-F02121 to STF22-F02124, SINTEF

66. Lucy, L. B., 1977. A numerical approach to the testing of the fission hypothesis. In *The Astronomical Journal*, 8(12), 1013-1024

67. MAECENAS, Modelling of Ageing in Concrete Nuclear Power Plant Structures, 2001. EU Research Project FIKS-CT-2001-00186, Coordinator: Department of Civil and Structural Engineering, University of Sheffield, Participants: University of Rome "La Sapienza", Czech Technical University of Prague, British Energy Generation Ltd., International Centre for Mechanical Sciences (Italy), Ecole Centrale de Nantes, Health and Safety Executive (UK), University of Glasgow

68. Maekawa, K., Pimanmas, A., Okahura, H., 2003. Nonlinear Mechanics of Reinforced Concrete. Spon Press, ISBN 0-415-27126-6

69. Mariani, S., Perego, U. 2003. Extended finite element method for quasi-brittle fracture. In *International Journal for Numerical Methods in Engineering*, 58, pp. 103-126

70. Mathworks, The, 2005. MATLAB and Simulink Release Notes for Release 14 with Service Pack 3, The Mathworks, Inc.

71. Mathworks, The, 2007. MATLAB. At http://www.mathworks.com/

72. McCrum, D. 2006. Higher-Order Modelling Aspects of the Numerical Manifold Method. MSc dissertation, Department of Civil Engineering, University of Glasgow

73. Moës, N., Dolbow, J., Belytschko, T. 1999. A Finite Element Method for Crack Growth without Remeshing. In *International Journal of Numerical Methods in Engineering*, Vol. 43, 1: 131-150

74. Moës, N., Sukumar, N., Moran, B., Belytschko, T., 2000, An Extended Finite Element Method (XFEM) for Two- and Three-Dimensional Crack Modelling, European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Barcelona, 11-14 September 2000

75. Moës, N., Sukumar, N., Moran, B., Belytschko, T., 2000, Extended finite element method for three-dimensional crack modelling, In *International Journal of Numerical Methods in Engineering*, 48: 1549-1570

76. Moës, N., Belytschko, T. 2002. Extended finite element method for cohesive crack growth. In *Engineering Fracture Mechanics*, 69, pp. 813-833

77. Moës, N., 2003. A computational approach to handle complex microstructure geometries. In *Computer Methods in Applied Mechanics and Engineering*, 192: 3163-3177

78. National Archives and Records Administration, 2007. Boulder (Hoover) Dam, 1941 (photograph). ARC identifier 519837. From: http://www.archives.gov/

79. Nayroles, B., Touzot, G., Villon, P., 1992. Generalizing the finite element method: diffuse approximation and diffuse elements. In *Computer Methods in Applied Mechanics and Engineering*, 10, 307-318

80. Oden, J. T., Duarte, C. A. M., Zienkiewicz, O. C., 1998. A new cloud-based *hp* finite element method. In *Computer Methods in Applied Mechanics and Engineering*, 153, 117-126

81. Ohnishi, Y., Ohtsu, H., Nishiyama, S., Koyama, T., Wu, J. H., 2001. Deformation analysis considering water effect by manifold method. In *Proceedings of Fourth International Conference on Analysis of Discontinuous Deformation*, N. Bićanić (Editor), University of Glasgow, 47-62

82. Pedersen, R. R., Simone, A., Stroeven, M. and Sluys, L. J., 2007. Mesoscopic Modelling of Concrete under Impact. In *Proceedings of Fracture Mechanics of Concrete and Concrete Structures*, A. Carpinteri et al. (Editors), Catania, Italy

83. Pijaudier-Cabot, G., Bažant, Z., 1987. Nonlocal damage theory. In *Journal of Engineering Mechanics*, ASCE, 113(10), 1512-1533

84. Pope, D. P., Vitek, V., 1995. Mechanisms of Quasi-Brittle Failure in Metals and Intermetallic Compounds. Pentagon report no. A064023

85. Roca, P., 2001, Studies on the structure of Gothic Cathedrals, Historical Constructions 2001, Edited by P. B. Lourenco and P. Roca

86. Rougier, E., Munjiza, A., John, N. W. M., 2004. Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics. In *International Journal for Numerical Methods in Engineering*, 61, 856-879

87. Runge, C. D. T., 1901. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, Zeitschrift für Mathematik und Physik 46: 224–243

88. Sandia National Laboratories, 2000. Pre-test Round Robin Analysis of a Pre-stressed Concrete Containment Vessel Model, NUREG/CR-6678, SAND 00-1535

89. Sandia National Laboratories, 2003. Overpressurization Test of a 1:4-Scale Prestressed Concrete Containment Vessel Model, NUREG/CR-6810, SAND 2003-0840P

90. Sandia National Laboratories, 2003. Post-test Analysis of the NUPEC/NRC 1:4 Scale Prestressed Concrete Containment Vessel Model, NUREG/CR-6809, SAND 2003-0839P

91. Sasaki, T. & Ohnishi, Y. 2001. Analysis of the Discontinuous Rock Mass by four-node iso-parametric Manifold Method. In *Proceedings of 4th International Conference on Analysis of Discontinuous Deformation*: 369-378, Glasgow, 6-8 June 2001

92. Schlangen, E., Van Mier, J. G. M., 1995. Crack Propagation in Sandstone: Combined Experimental and Numerical Approach. In *Rock Mechanics and Rock Engineering*, 28(2), 93-110

93. Selby, R. G., Vecchio, F. J., 1993. Three-dimensional constitutive relations for reinforced concrete. Publication no. 93-02, Department of Civil Engineering, University of Toronto, Canada

94. Shi, G.H., 1988. Discontinuous Deformation Analysis – A new numerical model for statics and dynamics of block systems. PhD Thesis. University of California, Berkeley

95. Shi, G. H., 1995. Simplex integration for Manifold Method and Discontinuous Deformation Analysis. In *Proceedings of Working Forum on the Manifold Method of Material Analysis*. Pp. 129-164

96. Shi, G.H. 1996. Manifold Method. In *Proceedings of the First International Forum on Discontinuous Deformation Analysis and Simulations of Discontinuous Medium*: 52-204, New Mexico, U.S.A.

97. Shi, G.H., 1997. Numerical manifold method. In Y. Ohnishi (ed.), *Proceedings of 2nd International Conference on Analysis of Discontinuous Deformation*: 1-35

98. Simone, A., 2003. Continuous-Discontinuous Modelling of Failure. PhD thesis, Delft University Press, ISBN 90-407-2434-2

99. Simone, A., Wells, G. N., Sluys, L. J., 2003. Discontinuities in Regularised Media. In *Proceedings of VII International Conference on Computational Plasticity (COMPLAS)*, E. Oñate and D. R. J. Owen (Editors), CIMNE, Barcelona

100. Simone, A., 2004. Partition of unity-based discontinuous elements for interface phenomena: computational issues. In *Communications in Numerical Methods in Engineering*, 20, 465-478

101. Stazi, F. L., Budyn, E., Chessa, J., Belytschko, T., 2003. An extended finite element method with higher-order elements for curved cracks. In *Computational Mechanics*, 31, 38-48

102. Stirling, C. G., 2004. Algorithmic and modelling aspects of the discontinuous behaviour of structural masonry. Doctorate thesis. University of Glasgow

103. Sukumar, N., 1998. The Natural Element Method in Solid Mechanics. Doctorate thesis. Northwestern University, Evanston, Illinois

104. Sukumar, N., Moës, N., Moran, B., Belytschko, T., 2000, Extended Finite Element Method for Three-Dimensional Crack Modelling. In *International Journal for Numerical Methods in Engineering*, Vol. 48, 11: 1549-1570

105.Sukumar, N., Prevost, J. H. 2003. Modelling quasi-static crack growth with the extended finite element method, Part I: Computer implementation. In *International Journal of Solids and Structures*, 40, pp. 7513-7537

106.Sukumar, N., 2004, Construction of polygonal interpolants: a maximum entropy approach, in *International Journal for Numerical Methods in Engineering*, 61: 2159-2181

107.Sukumar, N., Dolbow, J., Devan, A., Yvonnet, J., Chinesta, F., Ryckelynck, D., Lorong, P., Alfaro, I., Martínez, M. A., Cueto, E., Dolbaré, M., 2005. Meshless Methods and Partition of Unity Finite Elements, In *International Journal of Forming Processes*, 8, 409-427

108.Sun, Y., Miki, S., 2001. A Study on Coupling Calculation of Manifold Method and Saturated-Unsaturated Flow Analysis. In *Proceedings of 4th International Conference on Analysis of Discontinuous Deformation*, N. Bićanić (Editor), University of Glasgow, 391-400

109.Tabarraei, A., Sukumar, N., 2006, Application of Polygonal Finite Elements in Linear Elasticity, *International Journal of Computational Methods*, World Scientific Publishing Company, January 12, 2006

110. Taylor, R. L., Zienkiewicz, O. C. , Oñate, E., 1998. A hierarchical finite element method based on the partition-of unity. In *Computer Methods in Applied Mechanics and Engineering*, 152, 73-84.

111.Tcl Developer Xchange, 2007. Tcl/Tk. At http://www.tcl.tk/

112.Tijssens, M. G. A., Sluys, B. L. J., van der Giessen, E., 2000. Numerical simulation of quasi-brittle fracture using damaging cohesive surfaces. In *European Journal of Mechanics and Solids*, Elsevier, 19, 761-779

113.Timoshenko, S. P., Goodier, J. N., 1951. Theory of Elasticity. McGraw-Hill

114.Timoshenko, S. P., Goodier, J. N., 1970. Theory of Elasticity. McGraw-Hill

115.TNO DIANA BV, DIANA User's Manual release 9.2, Analysis Procedures, Edited by: Frits C. de Witte and Max A. N. Hendriks, Published by TNO DIANA BV, First edition, February 12, 2007

116.TNO DIANA BV, DIANA User's Manual release 9.2, Application Modules, Edited by: Frits C. de Witte and Max A. N. Hendriks, Published by TNO DIANA BV, First edition, February 12, 2007

117.TNO DIANA BV, DIANA User's Manual release 9.2, Material library, Edited by: Frits C. de Witte and Max A. N. Hendriks, Published by TNO DIANA BV, First edition, February 12, 2007

118.United States Geological Survey, 2007. The October 17, 1989, Loma Prieta, California, Earthquake - Selected Photographs. From http://pubs.usgs.gov/dds/dds-29

119.University of Cambridge, 2007. Detail of vault of King's College Chapel, University of Cambridge, photograph. From http://www.kings.cam.ac.uk/

120.Varga, R. S., 1962. Matrix Iterative Analysis. Prentice-Hall

121. Vardoulakis, I., Sulem, J., 1995. Bifurcation analysis in geomechanics. Blackie Academic & Professional, ISBN 0751402141

122. Vecchio, F. J., 1992. Finite element modelling of concrete expansion and confinement. In *Journal of Structural Engineering*, ASCE, 118(9), 2390-2406

123. Wang, S., Zhu, W., Li, S., Qui, X., 2001. The numerical manifold method considering lateral effect and its application to engineering. In *Chinese Journal of Rock Mechanics and Engineering*, 20 (3), 297-300

124. Weisstein, E. W. Simplex. In *Mathworld*, a Wolfram Web Resource, http://mathworld.wolfram.com/simplex.html

125. Wells, G. N., 2001. Discontinuous modelling of strain localisation and failure. PhD thesis. Delft University Press, ISBN 90-9014731-4

126. Wells, G. N., Sluys, L. J., 2001. A new method for modelling cohesive cracks using finite elements. In *International Journal for Numerical Methods in Engineering*, 50 (12), 2667-2682

127. Wells, G.N., 2002. A p-adaptive scheme for overcoming volumetric locking during plastic flow. In *Computer Methods in Applied Mechanics and Engineering*, 191, 3153-3164

128. Wikipedia, 2007. Zhaozhou Bridge photograph. From http://en.wikipedia.org/wiki/Zhaozhou_Bridge

129. Zhang, G., Wang, G., Pei, J., 2001. Structure failure analysis based on manifold method. In *Chinese Journal of Rock Mechanics and Engineering*, 20 (3), 281-287

130. Zhang, G. X., Wu, X. F., Sugiura, Y., Hasegawa, H., 2001. Coupled Hydro-Mechanical Analysis of Jointed Rock Masses by Manifold Method. In *Proceedings of 4th International Conference on Analysis of Discontinuous Deformation*, N. Bićanić (Editor), University of Glasgow, 425-426

131. Zhang, N., Zhou, X., Sha, D., Yuan, X., Tamma, K., Chen, B., 2006. Integrating Mesh and Meshfree Methods for Physics-Based Fracture and Debris Cloud Simulation. In *Proceedings of Eurographics Symposium on Point-Based Graphics*, The Eurographics Association, 2006

132. Zienkiewicz, O. C., Cheung, Y. K., 1964. The finite element method for analysis of isotropic and orthotropic slabs. In *Proceedings of Institute of Civil Engineering*, 28, 471-488

133. Zienkiewicz, O. C. & Taylor, R.L., 2000. The Finite Element Method. Volume 1. The Basis. Butterworth-Heinemann

134. Zienkiewicz, O. C. & Taylor, R. L., 2005, The Finite Element Method For Solids and Structural Mechanics, Sixth edition, published by Elsevier Butterworth-Heinemann, ISBN 0 7506 6321 9