May 24, 2025

To be published by Springer-Nature, in press.

Finite Element Procedures – *En Plus*

Klaus-Jürgen Bathe

Professor of Mechanical Engineering

Massachusetts Institute of Technology

– **Finite element analysis is an art to predict the future**
–

and assess the past –

A statement made in 1967 *"Progress in design of new structures seems to be unlimited."*

Last sentence of article: "The Use of the Electronic Computer in Structural Analysis," by K.J. Bathe (undergraduate student), published in *Impact, Journal of the University of Cape Town Engineering Society*, pp.57-61, 1967.

Contents

Preface

The first printing of the book *Finite Element Procedures* was published by Prentice Hall in 1996, see KJ Bathe [A], and the book was printed 16 times. Seeing the success of this book, I updated the book about 10 years ago, but not extensively. At that time, *I deliberately aimed to not change the page numbers in the main body of the book* to be able to refer to specific pages for all printed copies, which is very important in correspondence regarding research and teaching.

However, I had to call the updated book a second edition, because there were too many changes to continue printing the book as a first edition. This second edition has been printed 7 times.

Due to major developments in finite element procedures during the last two decades, it is now necessary to substantially add to the material presented earlier. I am aiming to do so in this new book, the "*En Plus*" meaning to add to the book *Finite Element Procedures, 2nd edition.*

The basic philosophy of this *En Plus* book is the same as of the *Finite Element Procedures* book. The *En Plus* shall not present a survey of recent developments but only focus on, quoting from the earlier Preface, "basic finite element procedures, very useful in engineering and the sciences and that will probably be employed for many years to come". An important aspect of a finite element procedure is its reliability, see KJ Bathe [A, I, L], and hence we focus on finite element procedures that are general, reliable and efficient.

While the focus is only on developments of linear analysis procedures for solids and structures; the extension of these methods to nonlinear analysis would be largely achieved as discussed in the book *Finite Element Procedures.*

When reading and using this book, it will as well become apparent that there are still many exciting avenues for further developments of computational procedures with, in particular, the possibilities of Machine Learning to strengthen analysis capabilities – surely, this avenue will likely receive much attention and valuable developments.

With these thoughts, an important reason for me to write this book is also to show the wealth of possible further developments in the field of computational engineering and to raise interest in some exciting research to still be pursued.

Since I have been working for some decades in the field of computational engineering, I have seen interesting advances during the last decades, accomplished by many researchers. While following these activities, I was not involved and hence only refer briefly to some of these developments in this book. I describe my journey and focus in research and developments at the University of California, Berkeley and at M.I.T. over the years in the book *To Enrich Life, 2nd edn.*

In closing, I would like to thoroughly thank all my students and colleagues who collaborated with me over many years and have greatly contributed to my knowledge. It was a great privilege to work with them. Indeed, this *En Plus* book is largely based on the research we considered important and therefore pursued, and the papers we have written during the recent years.

As for the book *Finite Element Procedures*, I humbly hope that this book *En Plus* will also be of value to many students and professionals, to enrich their knowledge of engineering analysis. Towards this aim, throughout my life, following Johann Wolfgang von Goethe, I pursued my work with the following adage in mind "*Willst du dich deines Wertes freuen, so mußt der Welt du Wert verleihen"(FN) ,* and I tried to work

"with the spirit of God sparking my faith, hope and love into new action each day, hoping that my small contributions will be beneficial to many people".

Klaus-Jürgen Bathe

M.I.T., Cambridge, MA

FN:

J.W. von Goethe's adage says in English (approximately)*:*

"If you want to be happy about your value, then you need to give value to others to the world"

Here "value" for "Wert" means value in a very broad sense, to "enrich the life of others".

1. Chapter 1 – Introduction

Finite element procedures are already widely used, and for some time, for the analysis of structures and multiphysics systems. While advances in finite element methods are continuously sought, the available methods can already be used for the analyses of very complex physical systems. Indeed, commercial programs, like ABAQUS, ADINA, ANSYS and NASTRAN, now offer powerful analyses capabilities with good pre- and post-processing features that are used world-wide with great success.

A driving factor to now have the wide use of finite element methods was the initial publication of freely available finite element computer programs, foremost NASTRAN, see NASA TM X-2637 [A], SAP IV of K.J. Bathe, E.L. Wilson, and F. E. Peterson [A] , NONSAP of K.J. Bathe and E.L. Wilson [A], see also KJ Bathe [G, M]. Soon, the development of commercial finite element programs was pursued, and the benefit of their use was realized. Another driving factor was the publication of books on finite element techniques. In addition to the book *Finite Element Procedures* by K.J. Bathe [A], many widely read books are available, and the reader can easily find titles by the famous authors O.C. Zienkiewicz, J.N. Reddy, T.J.R. Hughes, E. Oñate, J.T. Oden, I. Babuska, F. Brezzi, T.B. Belytschko and others. Some first books published on finite element methods were those by OC Zienkiewicz and YK Cheung [A], O.C. Zienkiewicz and P.B. Morice [A] and K.J. Bathe and E.L. Wilson [C]. In addition, also the wide availability of video lectures on the theory and use of finite element methods was valuable, see e.g. K.J. Bathe [P].

While the use of finite element methods was already quite mature for certain analyses, during the last two decades further advances in finite element methods have been accomplished and many more are still

forthcoming to model and analyze ever more complex physical systems. Some of these advances are the focus of this book.

This book builds on the material presented in the book "Finite Element Procedures", by an "*En Plus"*. The book focuses on some new finite element formulations for the analysis of shells and solids, the development and use of overlapping finite elements and meshes, with the aim to use AMORE (an acronym for "*A*utomatic *M*eshing with *O*verlapping and *R*egular *E*lements"), advances in solution procedures for predictions of dynamic response, with direct time integrations and frequency solutions, and finally on using Machine Learning with finite element procedures.

As mentioned in the Preface, we focus on some new finite element developments that are general, reliable and efficient for engineering analyses and scientific studies and that we believe to be fundamental and useful for many years to come. However, with this focus, we aim to avoid repeating what is already presented in the book *Finite Element Procedures,* see K.J. Bathe [A]. Also, because the focus is only on *some* new procedures for general finite element analyses, other perhaps important finite element methods are not discussed. For these, the reader is referred to the already published books and papers, and the references given therein.

Chapter 2 Advances in Shell Elements

We first briefly discuss the general difficulties in developing shell elements and then focus on the advances achieved in the family of MITC shell elements.

2.1 On shell element formulations and their testing

The effective analysis of shells has been a particular challenge in the development of finite element procedures and many different formulations have been proposed. The difficulties in reaching effective shell elements are that – in engineering practice – the elements should ideally be "optimal" in solving

any shell structure subjected to any (physically reasonable) boundary conditions and loadings. This means that the shell may be thin or thick, has different curvatures in different directions, and be subjected to any admissible boundary conditions in displacements, rotations and loads. Furthermore, the shell structure may be modeled with uniform (regular) or distorted meshes and the use of such meshes should lead to good solution results regardless of the asymptotic category of the problem, with membrane-dominated, bending-dominated, or mixed shell behaviors, see D. Chapelle and K.J. Bathe [A, B] and K.J Bathe, D. Chapelle and P.S. Lee [A].

An "ideal" shell finite element would pass the three basic tests:

1/ The *isotropy test* which ensures that any renumbering of the element nodes, like using different element nodes to start, and using clockwise or anticlockwise node numbering, will not change the basic properties of the element and solution results; and

2/ the *spurious zero energy mode test* which ensures that the element does not contain a spurious zero eigenvalue; and

3/ *the patch test* which ensures that the element can represent the basic constant membrane, shear and bending strain states.

In addition, the “ideal” shell element does not lock, that is, the element satisfies the relevant inf-sup conditions. Then the element should show the optimal possible convergence rate in solutions, see K.J. Bathe [A], D. Chapelle and KJ Bathe [B, D], and Y. Ko and K.J. Bathe [B].

Furthermore, such shell element would not contain any adjustable numerical factor, since artificial factors can render the element unreliable, although methods are used to stabilize element formulations, see for example T. Belytschko and C.S. Tsai[A] and W.A. Wall, M. Bischoff and E. Ramm [A].

Even though the above conditions are deemed satisfied, the element should be further tested, see the discussion in Section 2.1.2. In particular, the “optimality” of the shell element discretization would be

seen by measuring the solution error in an *appropriate* norm in the solution of appropriate membrane- and bending-dominated shell problems. Discretizations using an optimal element would always show the solution error curve close to the theoretically optimal curve. We exemplify this behavior in example solutions below.

These requirements for an effective shell element and the modeling problems encountered are focused on in the books by D. Chapelle and K.J. Bathe [B] and M.L Bucalem and K.J. Bathe [A]. In the following, we summarize some important observations.

2.1.1 Firstly – regarding shell element formulations

An important point is that finite element formulations based on the superposition of bending and membrane actions may be limited in their applications because the coupling of bending and membrane actions is not properly enforced within the element domains. Hence convergence to the correct solution of a shell problem may not be reached when using such "bending-membrane coupled" elements. The correct solution is either the analytical solution or a solution obtained with a very fine mesh of correctly formulated elements; these could also be three-dimensional elements.

The approach used in the formulation of bending-membrane coupled elements gives the element stiffness matrix **K** used in the following equation

$$\mathbf{KU} = \begin{bmatrix} \mathbf{K}_b & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & k_d \end{bmatrix} \begin{bmatrix} \mathbf{U}_b \\ \mathbf{U}_m \\ \theta \end{bmatrix} \tag{2.1}$$

where $\mathbf{K}_b$ and $\mathbf{K}_m$ are the plate bending and plate membrane stiffness matrices, formulated in the plane of the element and corresponding to the usual degrees of freedom, and k_d is the stiffness for the rotation

θ about the element normal axis. Assuming that the usual nodal bending transverse displacement and in-plane rotations, and the usual nodal membrane in-plane displacements are listed in $\mathbf{U}_b$ and $\mathbf{U}_m$, respectively, and are measured in the x, y coordinate system, θ is a rotation about the z-axis. This degree of freedom is commonly referred to as a "drilling rotation". The complete stiffness matrix shows that there is no coupling between the plate bending, membrane and drilling stiffnesses.

Equation (2.1) seems to imply that if we have an effective plate bending element, and many such elements are available, see for example J.L. Batoz, K.J. Bathe and L.W. Ho [A] for 3-node triangular plate bending elements, and also an effective plane stress element, and here too many such elements are available, then these two element stiffness matrices can simply be combined to obtain an acceptable shell element. This is, however, not the case and we refer to D. Chapelle and K.J. Bathe [B] for a deep discussion and an example which shows that with this approach convergence to the correct (analytical) solution may not be reached.

The first shell elements were published based on this approach and then the drilling stiffness was selected based on the other diagonal entries in the matrix and based on a "reasonable " physical judgement. Since the primary reason for introducing this stiffness is to eliminate the physical rigid body mode about the z-axis, a small fraction of the smallest of the diagonal stiffness matrix elements has been used, see for example K.J. Bathe and L.W. Ho [A]. However, also a significant research effort has been directed to naturally, and based on strong theory, include the drilling degrees of freedom in the element formulations, see D. J. Allman [A, B, C], R.D. Cook [A], A. Ibrahimbegovic and E.L. Wilson [A] and K. Wisniewski and E. Turska [A].

Given these considerations, we have focused our research efforts for shell analyses on the development of shell elements based on degenerating three-dimensional solid element formulations and using curvilinear coordinate systems. For these elements, the membrane and bending actions are automatically fully coupled throughout the element domains which, as mentioned in D. Chapelle and K.J. Bathe [B], is important for convergence to the continuum mechanics exact solution when shells of smooth large

surfaces are analyzed. The drilling degree of freedom is not present (but can be introduced, see below), that is, only five degrees of freedom aligned with the shell surface are used.

Considering then intersecting shell surfaces, a node at the intersection of the shells may use an "average normal vector" with still five degrees of freedom if the angle between the actual normal vectors of the intersecting shells is small, and use six degrees of freedom when the shell surfaces intersect at a significant angle. In engineering practice, typically, when different shells physically intersect, the angle is large. Then the six degrees of freedom at a node would naturally arise because each shell surface would have a different normal vector and the associated five degrees of freedom for each shell surface and hence element node would be transformed to the six global degrees of freedom at the node. However, with such models a response prediction may not be accurate, even if there is no singularity in the stiffness matrix, because the element displacements associated with the nodal degrees of freedom do not necessarily preserve compatibility along the actually physically compatible edges of the shells. Also, in geometrically nonlinear analyses, convergence difficulties can be encountered. The use of transition elements may improve the results, see K.J. Bathe [A].

A practical case is also when a general beam element with bending and torsional stiffnesses is to be connected to a shell surface. To properly transmit these beam stiffnesses, constraint equations need to be used and the shell elements at that node need to carry six degrees of freedom into which the beam adds stiffness. Once again, the accuracy in such solutions may not be acceptable.

In such cases we frequently best use a shell element based on the approach we followed (that is, degenerating the three-dimensional solid element formulations and using curvilinear coordinate systems) but in addition to the usual five degrees of freedom, the element formulation includes the nodal drilling rotational degrees of freedom in the formulation. This is accomplished quite naturally for the lower and the higher-order MITC shell elements in the following manner.

Consider, for example, that the MITC4 shell element is to be enriched by drilling degrees of freedom and the approach of D.J. Allman [A, B]. We introduce for the shell element "fictious" nodes 5, 6, 7, 8 at the

mid-points of the element sides. Then assuming along the element edges, a linear tangential displacement and a parabolic normal in-plane displacement, the displacements at the nodes 5, 6, 7, 8 are expressed in terms of the nodal in-plane displacements plus the drilling rotations at the four element corner nodes. In this procedure, it can be expeditious to use for the fictious nodes, interpolation functions for the membrane normal displacements that express the relative displacements to the usual 4-node bi-linear displacement interpolations.

The membrane displacements at the fictitious nodes located at the mid-points of the element edges are then eliminated using the drilling degrees of freedom and thus the drilling rotations have been naturally introduced in the element formulation. This step can also enrich the element membrane strain field. The complete shell element formulation then couples within the element domain the assumed internal membrane and bending displacements including the effects of the drilling displacements, see Y. Ko, K.J. Bathe and X. Zhang [A]. To include nodal drilling degrees of freedom and the associated effects within the element for the 3-node triangular shell element and other higher-order elements is achieved analogously.

There are important details to give attention to because it is important that the resulting shell element with drilling degrees of freedom passes the basic tests, that is, as discussed above, the spatial isotropy, the no-spurious zero energy mode, and the membrane, bending and shearing patch tests. The membrane, bending and shearing patch test should be passed with the minimum physical support conditions to physically only eliminate the physical rigid body modes. In addition, as also discussed above, the relevant inf-sup conditions should be passed and no artificial factor to stabilize the element formulation should be used. Also, in the numerical testing discussed in the next section, optimal convergence should be seen. For the details of such a formulation, using the MITC approach, see Y. Ko, K.J. Bathe and X. Zhang [A].

It is interesting to also see that the above approach could in addition be used to establish MITC shell elements that carry only at some corner nodes the drilling degrees of freedom, the other nodes still carry the usual five degrees of freedom. These elements could then be used to properly connect at shell intersections or with beams, and could be used as coupling elements to the usual five degrees of freedom shell elements. This approach could save considerable computational resources. The same idea is, in essence, followed in the AMORE scheme in which mostly traditional finite elements are used but some overlapping elements are employed, and these two types of elements are connected using coupling elements, see Section 3.6.

2.1.2 Secondly – regarding testing of shell elements

An important point, already implicitly referred to above, is that since a deep theoretical analysis of general shell element formulations is usually out of reach, appropriate numerical testing of any proposed shell element formulation should be performed, see K. J. Bathe, A. Iosilevich and D. Chapelle [A, B], D. Chapelle and K.J. Bathe [A, B, E], J.F. Hiller and K.J. Bathe [A], P.S. Lee and K.J. Bathe [A, B] and Y. Ko and K.J. Bathe [B].

The testing must be based on choosing appropriate shell structures to solve, with appropriate boundary conditions and applied loads, and must use proper norms to measure the convergence. The testing should also include to identify whether the relevant inf-sup conditions are satisfied.

In particular, membrane-dominated and bending-dominated shell conditions need to be considered, with single and double curvatures in the geometries, that is, elliptic, parabolic and hyperbolic geometries. We use in Sections 2.2 to 2.4 appropriate shell conditions for testing shell elements, that is, the structures, boundary conditions and loads to use in the tests.

In the numerical testing, it is important to measure the convergence of the finite element solutions in the chosen norm as the thickness of the shell structure decreases from $t/L = 0.01$ to $t/L = 0.0001$, which is the typical range of shell thicknesses, where t = thickness of the shell and L is a characteristic length. Ideally the convergence curves show the theoretical optimal slope in the norm and closely fall on the same line.

We demonstrate proper shell element testing in Sections 2.2 to 2.4. Such "all encompassing element testing" is important to understand the behavior of the shell elements in the various shell conditions, as mentioned above.

2.1.3 Some important element developments

In our developments for shell analyses, we have focused on MITC shell element formulations, where MITC stands for Mixed Interpolation of Tensorial Components, see KJ Bathe and EN Dvorkin [A], because we have found this approach to lead to effective shell elements, see Sections 2.2 to 2.4.

The MITC approach is referred to by some authors as an Assumed Natural Strain (ANS) technique, see e.g. U. Andelfinger and E. Ramm [A] and G.M. Kulikov and S.V. Plotnikova [A].

Another approach is to use the Enhanced Assumed *S*train (EAS) method proposed by JC Simo and MS Rifai [A] to develop new shell elements. The EAS technique uses static condensation of the enhanced strain parameters, which can be costly and introduce difficulties. The approach is closely related to the method of incompatible modes, also using static condensation of internal variables. The incompatible modes method is very useful but needs to be employed with care, see T Sussman and KJ Bathe [A] for some practical difficulties encountered with elements using incompatible modes. Other authors have also used the EAS method for the development of shell elements, see for example M. Bischoff, E. Ramm and

D. Braess [A]. and J. M. A. César de Sá, R.M. Natal Jorge, R. A. Fontes Valente and P.M. Almeida Areias [A].

There have been several other attempts to alleviate the shear and membrane locking, in particular of 4-node continuum mechanics-based shell finite elements. The method of reduced integration, see e.g. T. Belytschko and C.S. Tsai [A], can alleviate membrane locking as well as shear locking, but the elements largely suffer from rank deficiency unless artificial stabilization factors are used.

Another valuable development is the approach given by K.U. Bletzinger, M. Bischoff and E. Ramm [A]. The authors introduced the Discrete Shear Gap method for the formulation of shell elements.

Also important are the developments of H.Y. Roh and M. Cho [A] and G.M. Kulikov and S.V. Plotnikova [A] who worked on the developments of shell elements using the exact shell geometry.

Even though the above-referred to elements – like many other elements – may not have been thoroughly tested by the developers in the way we describe above and demonstrate below, the element developments may be very valuable and may also be used in widely employed computer programs, but then the program developers have to perform a thorough testing and perhaps introduce some “changes and tuning” to make the elements “fit for industrial use”.

Since we have not been involved in such developments, we cannot describe the details of the formulations and evaluate the elements for practical use. Instead, we next focus on the development of significant improvements of the original MITC4 and MITC3 elements and the proper testing of the improved elements.

2.2 The MITC4+ shell element

We first give some thoughts on how the 4-node MITC4 shell element was "born" and then focus on the formulation of the significantly improved MITC4+ shell element.

2.2.1 Thoughts on the development of MITC shell elements

All general MITC shell elements are based on the fundamental mathematical model discussed in D. Chapelle and KJ Bathe [B, C], but the *shear and membrane locking* of an element needs to be prevented. To alleviate the shear locking behavior, the MITC method was established to obtain various quadrilateral and triangular shell finite elements. Of particular interest in engineering practice are 4-node and 3-node continuum mechanics-based shell elements.

The MITC4 element was the first shell element developed in the series of elements based on Mixed Interpolation of Tensorial Components, here the "4" signifies 4 nodes, see E.N. Dvorkin and K.J. Bathe [A] and K.J. Bathe and E.N. Dvorkin [A]. The element is observed to usually give good solution accuracy when uniform meshes are employed to model the problem. This behavior is seen because of the effective treatment of shear locking, see also K. J. Bathe and E.N. Dvorkin [B]. The MITC4 shell element is widely used in engineering practice, but since membrane locking is not treated in the element formulation, the solution accuracy can deteriorate when curved geometries are solved with distorted meshes.

The key points are that the MITC4 shell element is formulated by interpolating the displacements as for the displacement-based 4-node shell element degenerated from the three-dimensional behavior of the continuum. However, instead of simply using the displacements to evaluate the transverse shear strains, in the MITC4 formulation a separate interpolation of the covariant transverse shear strain components is introduced which is tied to the interpolations of the displacements.

The approach of the transverse shear formulation for the shell element is conceptually based on knowing how shear locking is prevented in a 2-node displacement-based beam element based on Timoshenko beam theory. In this case a constant transverse shear strain is assumed and tied to the shear strain calculated from the displacement interpolations at the element mid-point to relieve the element of shear locking, see K.J. Bathe [A]. In the case of the 4-node MITC shell element we think conceptually of four Timoshenko beam elements with one beam representing each of the sides of the shell element and a constant transverse shear along each beam. We then use linear interpolations of the transvers shear strains (seen in the four conceptual beams) across the shell element.

The MITC4 element is formulated in convected coordinates, which is important to fully simulate the physical behavior of geometrically complex shells and relieve the element of shear locking with the above-described *Ansatz* (frequently referred to as the "Dvorkin-Bathe Ansatz") for the transverse shear strains. The same approach is used to formulate the 3-node MITC3 shell element, see P.S. Lee and K.J. Bathe [D].

Thereafter, the MITC approach was employed to propose higher-order MITC shell elements, see KJ Bathe and E.N. Dvorkin [A], M.L. Bucalem and K.J. Bathe [B, C], P.S. Lee PS and K.J. Bathe [C], and D.N. Kim and K.J. Bathe [A], and to also propose novel plate elements, see F. Brezzi, K. J. Bathe and M. Fortin [A]. Whereas the general MITC shell elements could not be analyzed mathematically in depth, and hence thorough numerical testing (as for all shell elements) was necessary, see K.J. Bathe, A. Iosilevich and D. Chapelle [A, B], mathematical convergence analyses of the MITC plate elements were possible, see K.J. Bathe and F. Brezzi [A], K.J. Bathe, F. Brezzi and L.D. Marini LD [A] and K. J. Bathe, M. L. Bucalem and F. Brezzi [A].

All these MITC plate and shell element and related developments, are largely discussed in K.J. Bathe [A], and hence we do not focus on these developments in this book. Instead, we focus below on the

shortcomings of the MITC4 shell element, and the significant improvements reached in the MITC4+ shell element. Thereafter, we focus on the development of a triangular general MITC3 shell element and the significant improvements reached in the MITC3+ element, see Section 2.3. In addition, we discuss in Section 2.4 an element to solve for the fully three-dimensional shell behavior, also based on MITC interpolations.

2.2.2 The formulation of the MITC4+ element for general shell analyses

In engineering practice of shell analyses, the low-order 3-node and 4-node elements are by far mostly used. Indeed, the 4-node quadrilateral element is usually employed as much as possible and typically the 3-node triangular elements are only used to fill in areas where 4-noded elements become too distorted, typically when there is need to fill in empty spaces in a mesh generation. The strategy of only using 4-node and 3-node shell elements, and not higher-order shell elements using more than 4 nodes, is not necessarily the best for analyses of shell structures since the higher-order MITC shell elements can be effective, see M.L. Bucalem and K.J. Bathe [A, B, C]

With the development of the MITC4 shell element, many more analyses of complex shells could be performed reliably when, previously, difficulties were encountered due to shear locking. The element is widely employed using ADINA, see Bentley Systems, Inc. [A]. And the Dvorkin-Bathe Ansatz is also used to model the transverse shear behavior in other closely related formulations of quadrilateral shell elements, notably those used in commercial computer programs.

However, shortcomings of the MITC4 shell element and related 4-node elements were also identified. While the elements perform well regarding the possible detrimental effects of shear locking, fine meshes may be needed when *membrane locking* is possible, due to spurious membrane strains at the shell mid-surface. In addition, there is the possibility of *pinching locking*, also referred to as *thickness locking*, due to spurious strains through the shell thickness, see Section 2.4.1, which is however not as serious a locking phenomenon.

Shear locking is a result of the displacement interpolations not being able to exactly represent simple transverse shear strain situations; an example is the 2-node displacement-based beam element based on using the Timoshenko beam theory which cannot represent a constant moment with zero transverse shear strain. Similarly, the actions of membrane and pinching locking are also a result of the membrane strains and pinching strains not being correctly represented in the element by the displacement interpolations.

To reach an improved general MITC4 shell element, the difficulty lies in that we need a shell element that is not formulated based on an exact representation of the shell mid-surface and will not lock in shear and membrane actions. In the formulation of a continuum mechanics-based shell finite element, like the MITC4 element, curvatures do not explicitly appear in the formulation, which introduces difficulties to identify how to alleviate membrane locking to reach a "generally applicable element for any shell geometry and boundary conditions".

Let us briefly recall some basic equations used in the formulation of the standard 4-node displacement-based shell element. The geometry of the element is interpolated using, see K.J. Bathe [A]

$$\boldsymbol{x}(r,s,t) = \sum_{i=1}^{4} h_i(r,s)\boldsymbol{x}_i + \frac{t}{2}\sum_{i=1}^{4} a_i h_i(r,s)\boldsymbol{V}_n^i, \qquad (2.2)$$

where the $h_i(r,s)$ are the interpolation functions of the standard two-dimensional isoparametric element corresponding to node i, $\boldsymbol{x}_i$ is the position vector of node i in the global Cartesian coordinate system, and a_i and $\boldsymbol{V}_n^i$ denote the shell thickness and the director vector at the node, respectively, see Fig. 2.1.

The interpolation functions $h_i(r,s)$ are given by

$$h_i(r,s) = \frac{1}{4}(1+\xi_i r)(1+\eta_i s) \quad \text{for} \quad i = 1,2,3,4, \qquad (2.3)$$

with

$$[\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4] = [1 \quad -1 \quad -1 \quad 1],\ [\eta_1 \quad \eta_2 \quad \eta_3 \quad \eta_4] = [1 \quad 1 \quad -1 \quad -1] \qquad (2.4)$$

The corresponding element displacement interpolation obtained from Eq. (2.2) is

$$\boldsymbol{u}(r,s,t) = \sum_{i=1}^{4} h_i(r,s)\boldsymbol{u}_i + \frac{t}{2}\sum_{i=1}^{4} a_i h_i(r,s)(-\boldsymbol{V}_2^i\alpha_i + \boldsymbol{V}_1^i\beta_i), \qquad (2.5)$$

in which $\boldsymbol{u}_i$ is the nodal displacement vector in the global Cartesian coordinate system, $\mathbf{V}_1^i$ and $\boldsymbol{V}_2^i$ are unit vectors orthogonal to $\boldsymbol{V}_n^i$ and to each other, and α_i, β_i are the rotations of the director vector $\boldsymbol{V}_n^i$ about $\mathbf{V}_1^i$ and $\boldsymbol{V}_2^i$, respectively, at node i. Note that we do not introduce a rotation about $\boldsymbol{V}_n^i$ (a drilling degree of freedom) which for a non-smooth shell surface requires that the nodal vector $\boldsymbol{V}_n^i$ is the mean of the vectors normal to the shell surfaces meeting at the node. If that mean is measured to be significantly different from the normal vectors of the individual shell surfaces, like for intersecting shells, the procedures described in Section 2.1.1 should be used.

The linear terms of the displacement-based covariant strain components are given by

$$e_{ij} = \frac{1}{2}(\boldsymbol{g}_i \cdot \boldsymbol{u}_{,j} + \boldsymbol{g}_j \cdot \boldsymbol{u}_{,i}), \qquad (2.6)$$

in which

$$\boldsymbol{g}_i = \frac{\partial \boldsymbol{x}}{\partial r_i},\ \boldsymbol{u}_{,i} = \frac{\partial \boldsymbol{u}}{\partial r_i} \text{ with } r_1 = r, r_2 = s, r_3 = t. \qquad (2.7)$$

In the MITC4 shell element formulation, the covariant in-plane strain components are calculated using Eqs. (2.2) to (2.7) without any modification. However, the transverse shear strain field is based on assuming constant covariant transverse shear strain components along the element edges, see EN Dvorkin and KJ Bathe [A]

$$\tilde{e}_{rt} = \frac{1}{2}(1+s)e_{rt}^{(A)} + \frac{1}{2}(1-s)e_{rt}^{(B)}$$

(2.8)

$$\tilde{e}_{st} = \frac{1}{2}(1+r)e_{st}^{(C)} + \frac{1}{2}(1-r)e_{st}^{(D)},$$

where the $e_{rt}^{(A)}$, $e_{rt}^{(B,)}$, $e_{st}^{(C)}$, $e_{st}^{(D)}$ are the transverse shear strains obtained from Eq. (2.6) evaluated at the points *A, B, C* and *D* of the element, respectively, see Fig. 2.2.

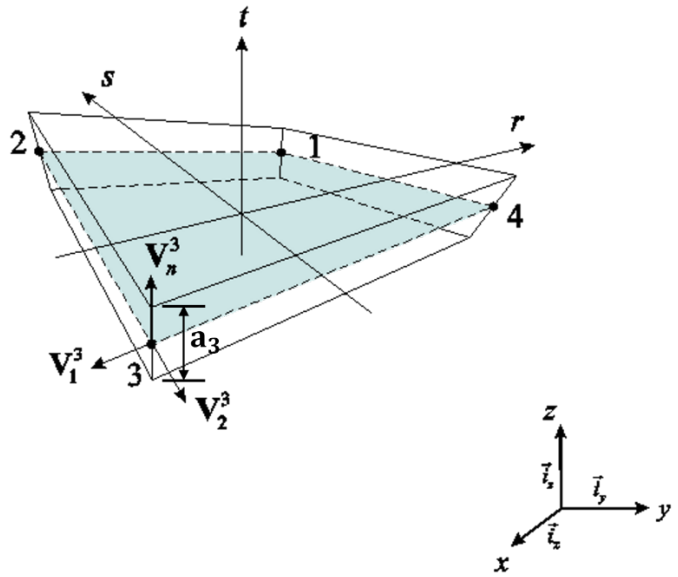


Figure 2.1 The standard continuum mechanics displacement-based 4-node shell element, K.J. Bathe [A].

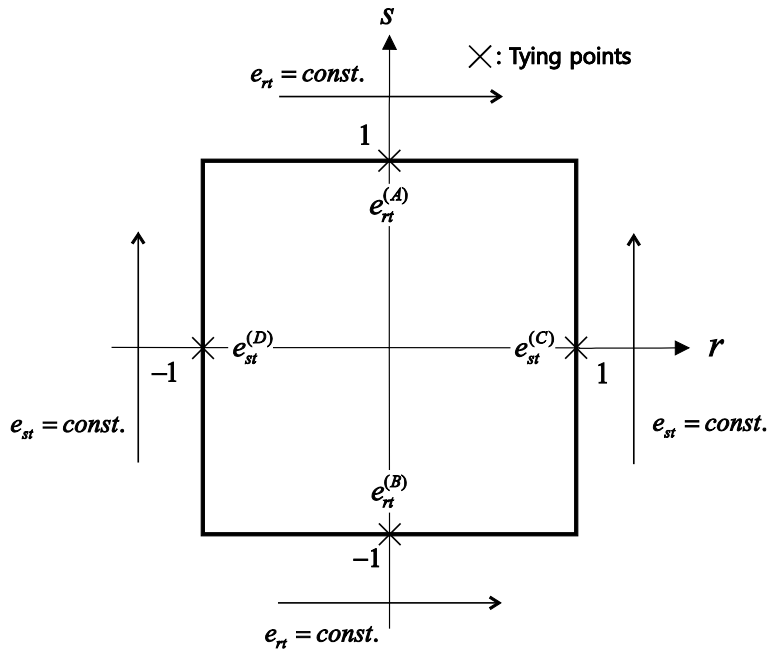


Fig. 2.2 The tying positions (A), (B), (C) and (D) for the assumed transverse shear strain field of the MITC4 and MITC4+ shell elements. The constant transverse shear strain conditions are imposed along the element edges, K.J. Bathe [A].

The assumption used in Eqs. (2.8) is also employed for the transverse shear strains of the MITC4+ shell element. However, to also prevent membrane locking, we need an assumption for the interpolation of the membrane strains at the mid-surface of the shell element, and we use the following formulation for the MITC4+ element, referred to as the “New MITC4+” in the publication by Y. Ko, P.S. Lee and K.J. Bathe [A].

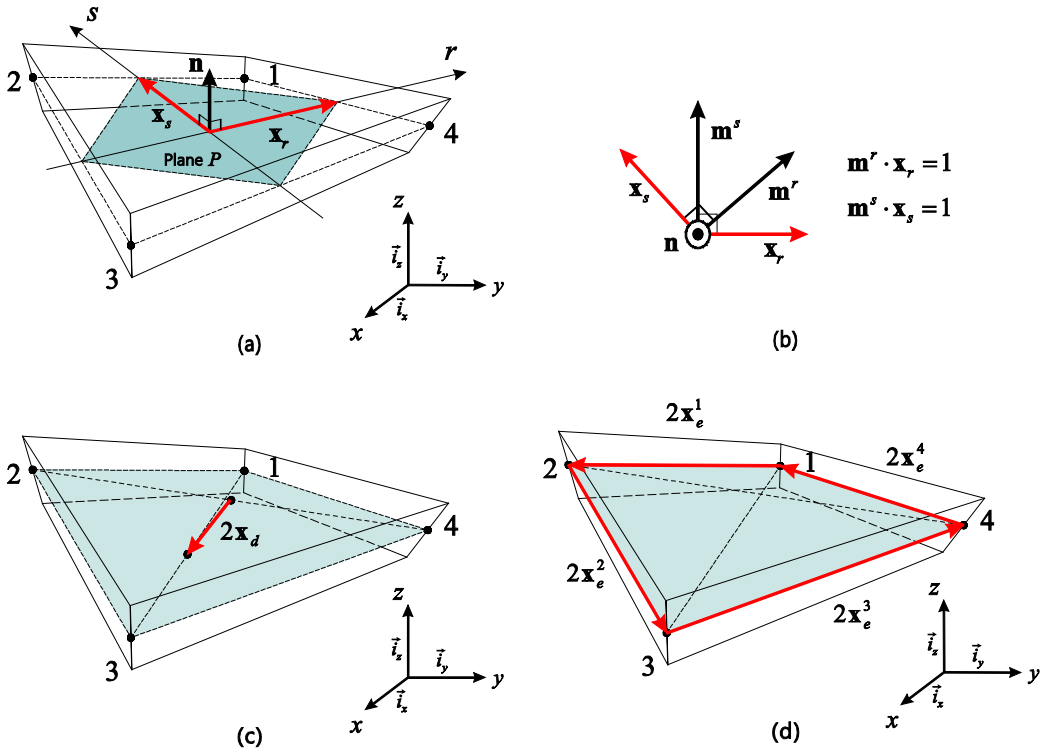


Fig.2.3 The vectors used for the element geometry. (a) and (b) The in-plane vectors. (c) The distortion vector $\boldsymbol{x}_d$ and (d) Four edge vectors $\boldsymbol{x}_e^1$, $\boldsymbol{x}_e^2$, $\boldsymbol{x}_e^3$ and $\boldsymbol{x}_e^4$. Regarding Figs. 2.3 to 2.14 and more information, see Y. Ko, P.S. Lee and K.J. Bathe [A].

The displacement-based membrane strain at the shell element mid-surface is given by

$$e_{ij}^m = \frac{1}{2}\left(\frac{\partial \boldsymbol{x}_m}{\partial r_i}\cdot\frac{\partial \boldsymbol{u}_m}{\partial r_j} + \frac{\partial \boldsymbol{x}_m}{\partial r_j}\cdot\frac{\partial \boldsymbol{u}_m}{\partial r_i}\right) \tag{2.9}$$

where

$$\boldsymbol{x}_m = \sum_{i=1}^4 h_i(r,s)\boldsymbol{x}_i \;\; ; \quad \boldsymbol{u}_m = \sum_{i=1}^4 h_i(r,s)\boldsymbol{u}_i \tag{2.10}$$

Consider the following useful vectors describing the geometry and displacements of the shell element at the mid-surface, see Fig. 2.3,

$$\boldsymbol{x}_r = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \xi_i \boldsymbol{x}_i, \quad \boldsymbol{x}_s = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \eta_i \boldsymbol{x}_i, \quad \boldsymbol{x}_d = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \xi_i \eta_i \boldsymbol{x}_i, \tag{2.11}$$

$$\boldsymbol{u}_r = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \xi_i \boldsymbol{u}_i, \quad \boldsymbol{u}_s = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \eta_i \boldsymbol{u}_i, \quad \boldsymbol{u}_d = \tfrac{1}{4}\textstyle\sum_{i=1}^{4} \xi_i \eta_i \boldsymbol{u}_i, \tag{2.12}$$

where we use Eq. (2.3). The characteristic geometry vectors $\boldsymbol{x}_r$, $\boldsymbol{x}_s$ and $\boldsymbol{x}_d$ arise naturally from the nodal point positions, and the vectors $\boldsymbol{u}_r$, $\boldsymbol{u}_s$ and $\boldsymbol{u}_d$ give the corresponding displacements. The geometry vector $\boldsymbol{x}_d$ connects the centers of the two diagonal lines in the element mid-surface for measuring a distortion of the element and can be written as

$$\boldsymbol{x}_d = (\boldsymbol{x}_d \cdot \boldsymbol{m}^r)\boldsymbol{x}_r + (\boldsymbol{x}_d \cdot \boldsymbol{m}^s)\boldsymbol{x}_s + (\boldsymbol{x}_d \cdot \boldsymbol{n})\boldsymbol{n}$$

with the first two terms measuring the in-plane distortion and the last term giving the out-of-plane distortion.

The displacement-based mid-surface strain terms in Eq. (2.9) can now be evaluated using

$$\frac{\partial \boldsymbol{x}_m}{\partial r} = \boldsymbol{x}_r + s\boldsymbol{x}_d, \quad \frac{\partial \boldsymbol{x}_m}{\partial s} = \boldsymbol{x}_s + r\boldsymbol{x}_d, \quad \frac{\partial \boldsymbol{u}_m}{\partial r} = \boldsymbol{u}_r + s\boldsymbol{u}_d, \quad \frac{\partial \boldsymbol{u}_m}{\partial s} = \boldsymbol{u}_s + r\boldsymbol{u}_d, \tag{2.13}$$

Following the results of HY Roh and M Cho [A] and G.M. Kulikov and S.V. Plotnikova [A] we use the tying positions shown in Fig. 2.4 and assume the mid-surface membrane strain field to be given by

$$\tilde{e}_{rr}^{m} = \frac{1}{2}(1 - 2a_A + s + 2a_A \cdot s^2)e_{rr}^{m(A)} + \frac{1}{2}(1 - 2a_B - s + 2a_B \cdot s^2)e_{rr}^{m(B)}$$
$$+ \ a_C(-1 + s^2)e_{ss}^{m(C)} + \ a_D(-1 + s^2)e_{ss}^{m(D)} + a_E(-1 + s^2)e_{rs}^{m(E)} \qquad (2.14a)$$

$$\tilde{e}_{ss}^{m} = \frac{1}{2}(1 - 2a_C + r + 2a_C \cdot r^2)e_{ss}^{m(C)} + \frac{1}{2}(1 - 2a_D - r + 2a_D \cdot r^2)e_{ss}^{m(D)}$$
$$+a_A(-1 + r^2)e_{rr}^{m(A)} + a_B(-1 + r^2)e_{rr}^{m(B)} + a_E(-1 + r^2)e_{rs}^{m(E)} \qquad (2.14b)$$

$$\tilde{e}_{rs}^{m} = \frac{1}{4}(r + 4a_A \cdot rs)e_{rr}^{m(A)} + \frac{1}{4}(-r + 4a_B \cdot rs)e_{rr}^{m(B)} + \frac{1}{4}(s + 4a_C \cdot rs)e_{ss}^{m(C)} +$$
$$\frac{1}{4}(-s + 4a_D \cdot rs)e_{ss}^{m(D)} + (1 + a_E \cdot rs)e_{rs}^{m(E)} \qquad (2.14c)$$

where we have used the notation of superscripts defined for Eq. (2.8) and

$$a_A = \frac{c_r(c_r-1)}{2a}, \quad a_B = \frac{c_r(c_r+1)}{2a}, \quad a_C = \frac{c_s(c_s-1)}{2a}, \quad a_D = \frac{c_s(c_s+1)}{2a}, \; a_E = \frac{2c_r c_s}{a} \qquad (2.15)$$

with, using the vectors in Fig. 2.3, including the edge vectors,

$$c_r = \boldsymbol{x}_d \cdot \boldsymbol{m}^r, \qquad c_s = \boldsymbol{x}_d \cdot \boldsymbol{m}^s \qquad (2.16)$$

$$a = {c_r}^2 + {c_s}^2 - 1 = (\boldsymbol{x}_e^2 \cdot \boldsymbol{m}^s)(\boldsymbol{x}_e^4 \cdot \boldsymbol{m}^s) + (\boldsymbol{x}_e^3 \cdot \boldsymbol{m}^r)(\boldsymbol{x}_e^1 \cdot \boldsymbol{m}^r) + 1 \qquad (2.17)$$

We note that the constant "*a"* measures the distortion of the element and is only very close to zero if the element is extremely distorted. We assume that a mesh generation scheme would identify such elements and not use them in the mesh.

We also observe that the constants in Eq. (2.15) are all zero when the distortion vector $\boldsymbol{x}_d$ is zero, and in that case the assumed normal strain in the *r*-direction varies linearly in the *s*-direction, the assumed normal strain in the *s*-direction varies linearly in the *r*-direction, and the assumed in-plane shear strain is equal to the tied value at the mid-point plus linear terms in *r* and *s* of the normal strains. The constants have been determined to fulfill the following conditions:

- The MITC4+ element performs as the MITC4 shell element when the element is flat (that is, the four element nodes are located in a flat plane). Hence in that case the results obtained using the MITC4+ element and the MITC4 element are identical, see Section 2.2.3. We note that when the element geometry is flat, the membrane parts of the MITC4 and the MITC4+ shell elements are identical to that of the displacement-based element.

- When the MITC4+ element is used in a curved surface and is distorted, the membrane actions do not introduce locking.

Since in the analysis of two-dimensional plane stress problems, the MITC4 and MITC4+ shell elements always yield identical solutions, namely those of the 4-node displacement-based plane stress element, it can be of benefit to introduce in both element formulations the use of incompatible modes – although element incompatible modes need to be employed with care, because the modes can introduce instabilities. see T. Sussman and K.J. Bathe [A].

Finally, we note that it is effective to use 2x2x2 Gauss numerical integration in the evaluation of the element stiffness matrices, for both elements. Hence the computational cost in using the MITC4+ shell element is only slightly larger than when using the MITC4 shell element.

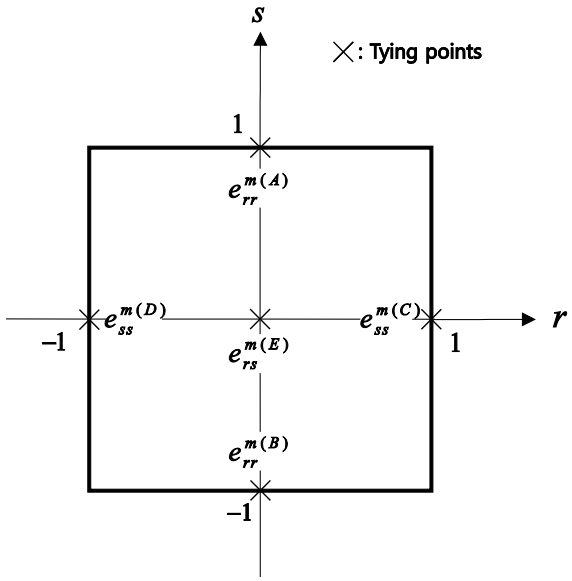


Fig. 2.4. Tying positions (A), (B), (C), (D) and (E) for the assumed mid-surface membrane strains. Note that in comparison to Fig. 2.2, the position (E) is also used.

The details of the MITC4+ element formulation are given in Y. Ko, P.S. Lee and K.J. Bathe [A], and the element is available for use in the ADINA program, see Bentley Systems, Inc. [A].

2.2.3 The testing of the MITC4+ element

As mentioned above, the thorough testing of shell elements is of much importance. For the reliability of an analysis, the element should pass the *basic tests:* the isotropy test, the test to not contain any spurious zero energy mode, and the patch test for the membrane, transverse shear and bending strains. The patch test can be performed using the patch of elements shown in Fig. 2.5. For more details about the tests, see K.J. Bathe [A].

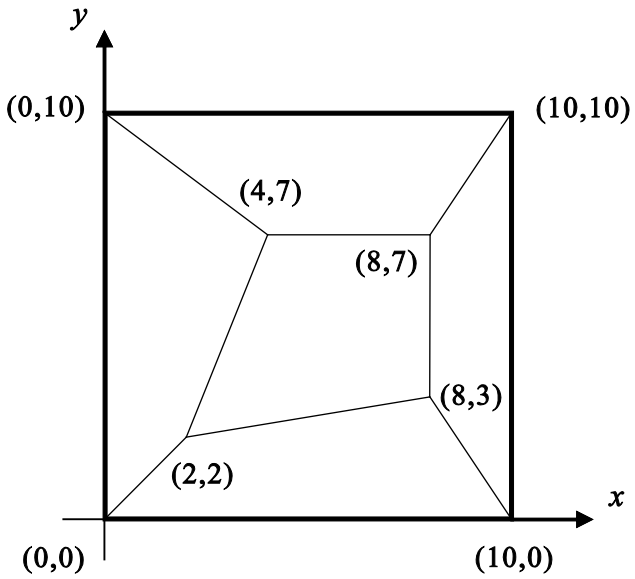


Fig. 2.5 Mesh used for patch tests.

The MITC4+ element passes the above basic tests and hence we can turn to solve a sequence of problems for a thorough testing of the element. The sequence of shell problems we propose to solve can be used in general to identify the robustness and effectiveness of a shell element. The chosen problems are *"*behavior all encompassing*"*, in that membrane-dominated and bending-dominated shell behaviors are considered – as encountered in practical shell solutions. Mixed shell behaviors with combined bending and membrane behaviors can then also be solved, see J.F. Hiller and K.J. Bathe [A] and K.J. Bathe, D. Chapelle and P.S. Lee [A]. Of course, the solution of additional test problems can also be valuable, and we might consider the problems we solve here as the minimum number of problems to solve for a thorough testing of a shell element.

We measure the solution errors in the s-norm proposed by Hiller and Bathe [A], which is appropriate to measure convergence for all shell problems considering membrane and bending-dominated conditions resulting from various curvatures, thicknesses, and boundary conditions. The s-norm is defined as

$$\|\boldsymbol{u}-\boldsymbol{u}_h\|_s^2=\int_\Omega \Delta\boldsymbol{\varepsilon}^T\Delta\boldsymbol{\tau}d\Omega \tag{2.18}$$

with $\Delta\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}_h, \Delta\boldsymbol{\tau}=\boldsymbol{\tau}-\boldsymbol{\tau}_h$

where $\boldsymbol{u}$ is the exact solution, $\boldsymbol{u}_h$ is the solution of the finite element discretization, and $\boldsymbol{\varepsilon}$ and $\boldsymbol{\tau}$ are the strain and stress vectors. This is a proper norm for investigating whether a finite element formulation is effective, for details on the use of the norm, see JF Hiller and KJ Bathe [A].

If a benchmark shell problem designed to detect locking has no analytical solution, an accurate finite element solution using a very fine mesh $\boldsymbol{u}_{ref}$ is used to replace the exact solution $\boldsymbol{u}$. Hence the s-norm in Eq. (2.18) is given by

$$\left\|\boldsymbol{u}_{ref}-\boldsymbol{u}_h\right\|_s^2=\int_{\Omega_{ref}}\Delta\boldsymbol{\varepsilon}^T\Delta\boldsymbol{\tau}\,d\Omega_{ref} \tag{2.19}$$

with

$$\Delta\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}_{ref}-\boldsymbol{\varepsilon}_h,\qquad \Delta\boldsymbol{\tau}=\boldsymbol{\tau}_{ref}-\boldsymbol{\tau}_h \tag{2.20}$$

It is important to study the solution convergence of shell finite elements as the shell thickness decreases and we use the normalized relative error E_h

$$E_h=\frac{\left\|\boldsymbol{u}_{ref}-\boldsymbol{u}_h\right\|_s^2}{\left\|\boldsymbol{u}_{ref}\right\|_s^2}. \tag{2.21}$$

The optimal convergence behavior is given by

$$E_h \cong Ch^k, \qquad (2.22)$$

in which C is a constant independent of the shell thickness and h is the element size. For 4-node shell elements, $k = 2$.

To obtain well-converged accurate reference solutions (using very fine meshes), any shell element known to be robust and effective may be employed and we have used the MITC9 shell element. The MITC9 element is known to satisfy the ellipticity and consistency conditions and to show good convergence behaviors, see K.J. Bathe, P.S. Lee and J.F. Hiller [A] and P.S. Lee PS and K.J. Bathe [C]. However, as mentioned earlier, in engineering practice, 4-node and 3-node shell elements are much preferred.

In each of the problem analyses below we compare the results obtained using the classical MITC4 shell element and the MITC4+ shell element, and we show how the results change as the thickness of the shell decreases. We focus on the use of regular and distorted meshes, because distorted meshes need frequently be used in engineering practice. In all solution graphs, we compare the results with the optimal rate of solution, shown by a bold line in the figures. The slope of this line, using Eq. (2.22) is $k = 2$.

Fully clamped square plate problem

The problem considered here is a very simple *shell* problem, because the shell is flat. Many shell elements can be used to solve this problem effectively. However, it is still a problem to consider in the sequence of test problems.

The plate is a clamped square plate of width $2L$, constant thickness t, subjected to a uniform pressure, see Fig. 2.6. Using symmetry, we can solve a one-quarter shell model, with the following shell boundary conditions: $u_x = \theta_y = 0$ along BC, $u_y = \theta_x = 0$ along DC and $u_x = u_y = u_z = \theta_x = \theta_y = 0$ along AB

and AD. In addition to the regular mesh shown in Fig.2.6, we also use the distorted meshes shown in Fig. 2.7. with L_1:L_2:L_3: …… L_N = 1 : 2 : 3 : …N.

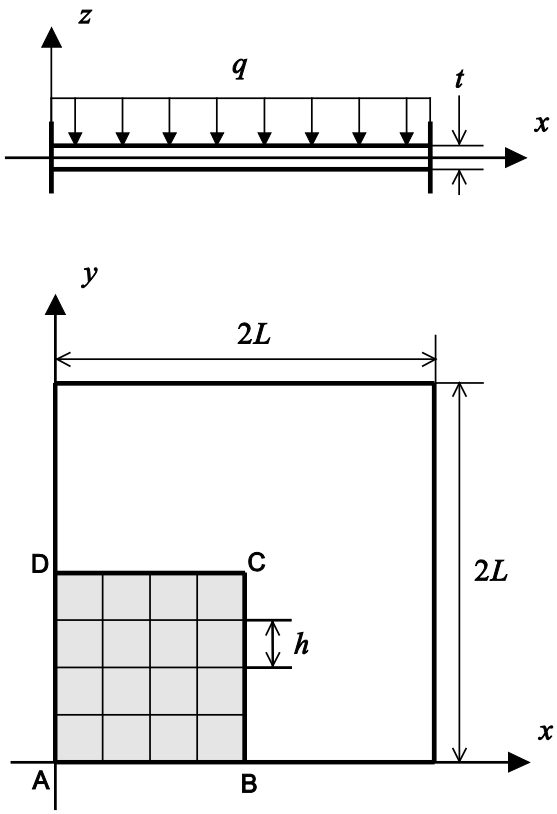


Fig. 2.6 Fully clamped square plate under uniform pressure

($L = 1.0$, $E = 1.7472 \times 10^7$, $q = 1.0$ and $\nu = 0.3$).

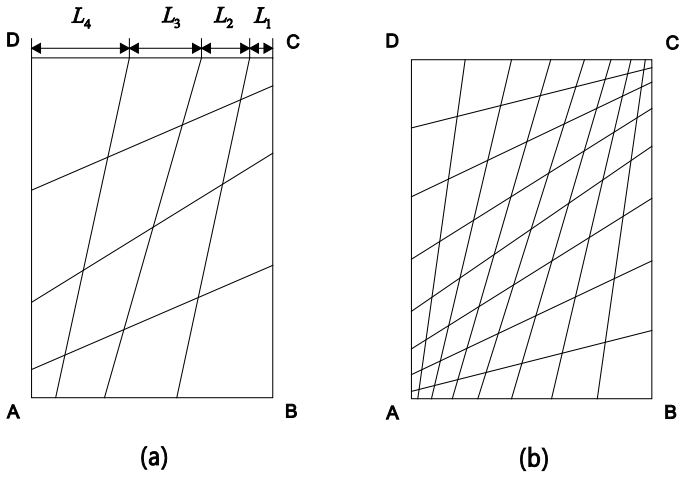


Fig. 2.7 Distorted mesh patterns for (a) $N = 4$ and (b) $N = 8$.

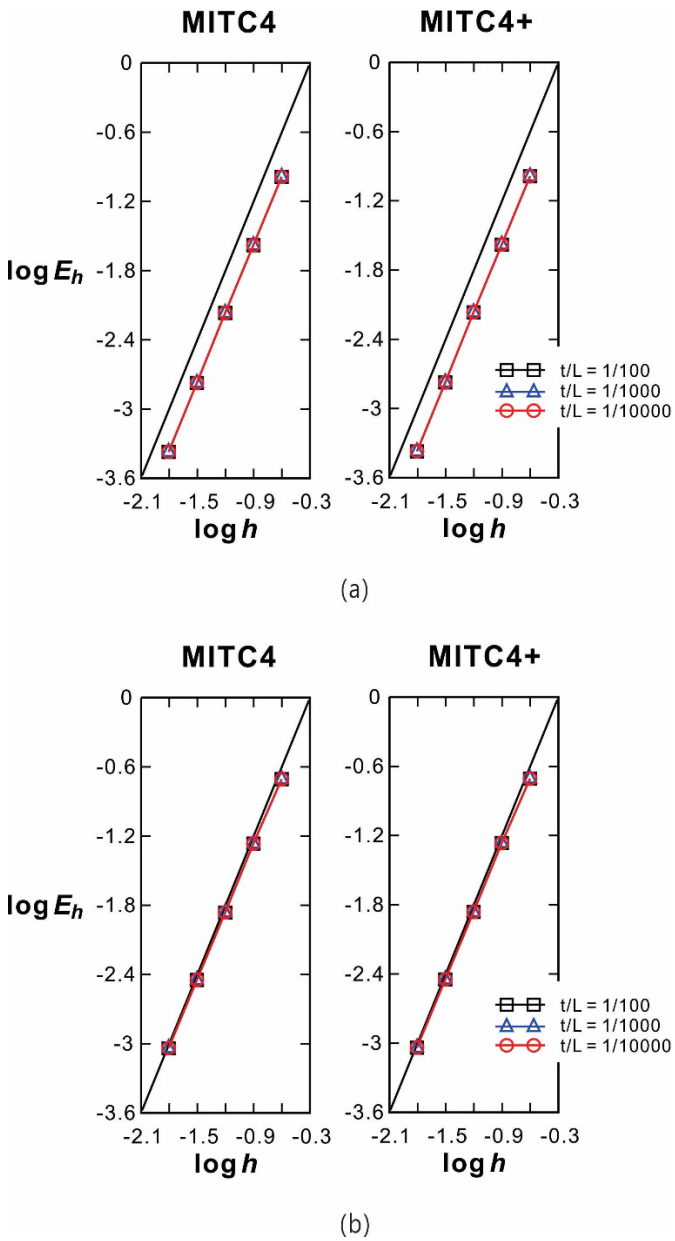


Fig. 2.8 Convergence curves for the fully clamped square plate problem, (a) the regular and (b) distorted meshes shown in Fig. 2.7.

Fig. 2.8 gives the convergence curves of the MITC4 and the MITC4+ shell elements. . A 72×72 element mesh of MITC9 shell elements was used to obtain the reference solution. We use $N \times N$ element meshes ($N = 4$, 8, 16, 32, and 64) to calculate the solutions. The element size in the convergence curves is $h =$

L/N. The performance of the elements is uniformly optimal in both the regular and distorted meshes. Note that membrane locking is not present in this plate bending problem.

Cylindrical shell problems

We consider the cylindrical shell of length $2L$, radius R and uniform thickness t as shown in Fig. 2.9. The loading is a smoothly varying pressure $p(\theta)$

$$p(\theta) = p_0 \, cos(\, 2\theta), \qquad (2.23)$$

We do not use a point load because point loads result in stress singularities. This shell structure shows two different asymptotic behaviors depending on the boundary conditions at its ends: bending-dominated behavior when both ends are free and membrane-dominated behavior when both ends are clamped.

Using symmetry, only the region ABCD is modeled. For the membrane-dominated case, the clamped boundary condition is imposed using $u_z = \beta = 0$ along AB, $u_y = \beta = 0$ along DC, and $u_x = \alpha = 0$ along BC, and $u_x = u_y = u_z = \alpha = \beta = 0$ along AD. For the bending-dominated case, the free boundary condition is imposed using $u_z = \beta = 0$ along AB, $u_y = \beta = 0$ along DC, and $u_x = \alpha = 0$ along BC. When using the clamped boundary condition, the regular mesh is graded with a boundary layer of width $2\sqrt{t}$, and when using the free boundary condition, with a boundary layer of width $0.5\sqrt{t}$, see D. Chapelle and K.J. Bathe [B]. We also perform the convergence studies with the distorted meshes shown in Fig. 2.9.

Fig. 2.10 gives the convergence curves in the solution of the clamped cylindrical shell problems. The reference solutions are calculated using a 72×72 element mesh of MITC9 shell elements. The solutions are obtained with $N \times N$ element meshes ($N = 4$, 8, 16, 32, and 64). The element size is $h = L/N$. In this problem, all shell elements present similarly good convergence behaviors.

Fig. 2.11 shows the convergence curves for the free cylindrical shell problems. Note that, in the regular meshes, all 4-node shell elements have a flat geometry, and thus membrane locking does not occur. However, in the distorted meshes, the element geometry is not flat, which induces membrane locking. In those cases, the performance of the MITC4 shell element severely deteriorates, while the MITC4+ shell element shows an excellent performance.

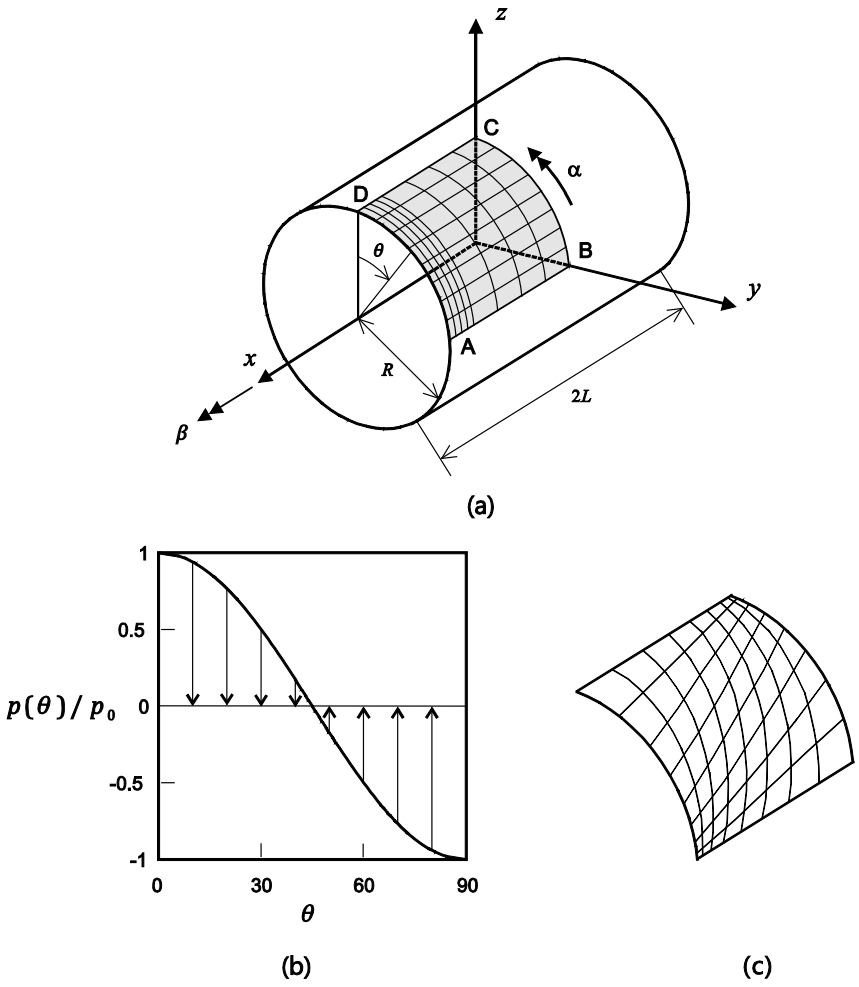


Fig. 2.9 Cylindrical shell problem ($L = R = 1.0$, $E = 2.0 \times 10^5$, $\nu = 1/3$ and $p_0 = 1.0$). (a) Problem description with the graded regular mesh for the clamped case (8×8 mesh). (b) Pressure loading. (c) Distorted mesh defined in Fig. 2.7 is used.

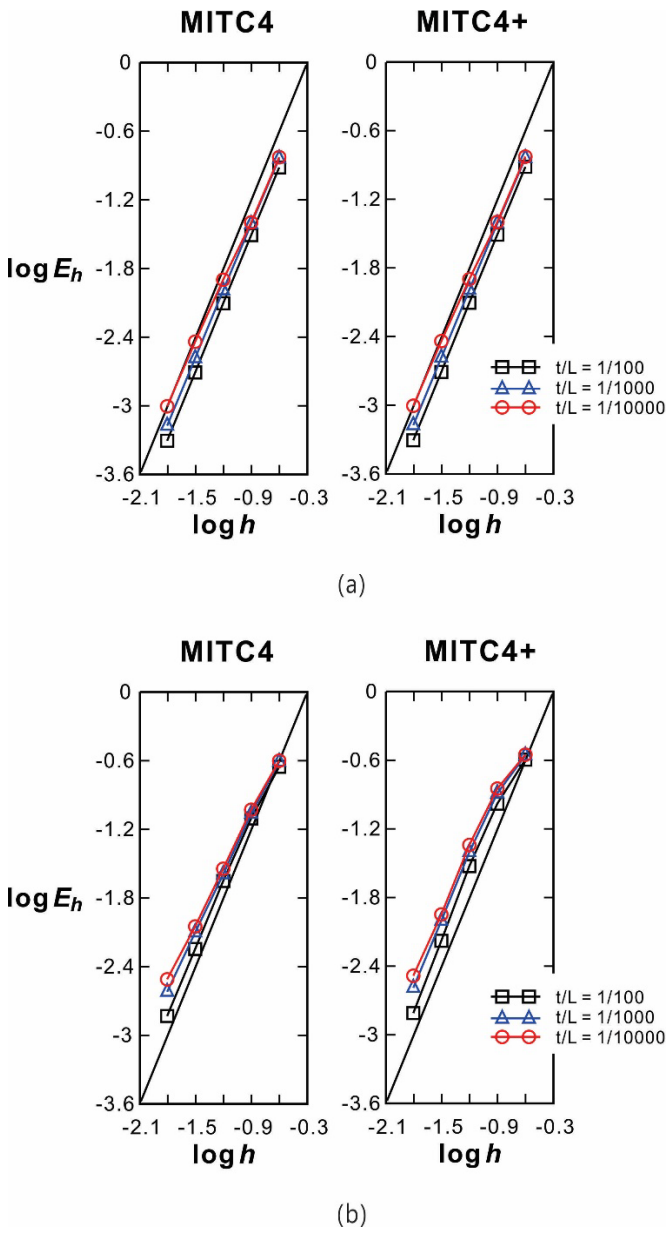


Fig. 2.10 Convergence curves for the clamped cylindrical shell problem with (a) the graded regular or (b) distorted meshes.

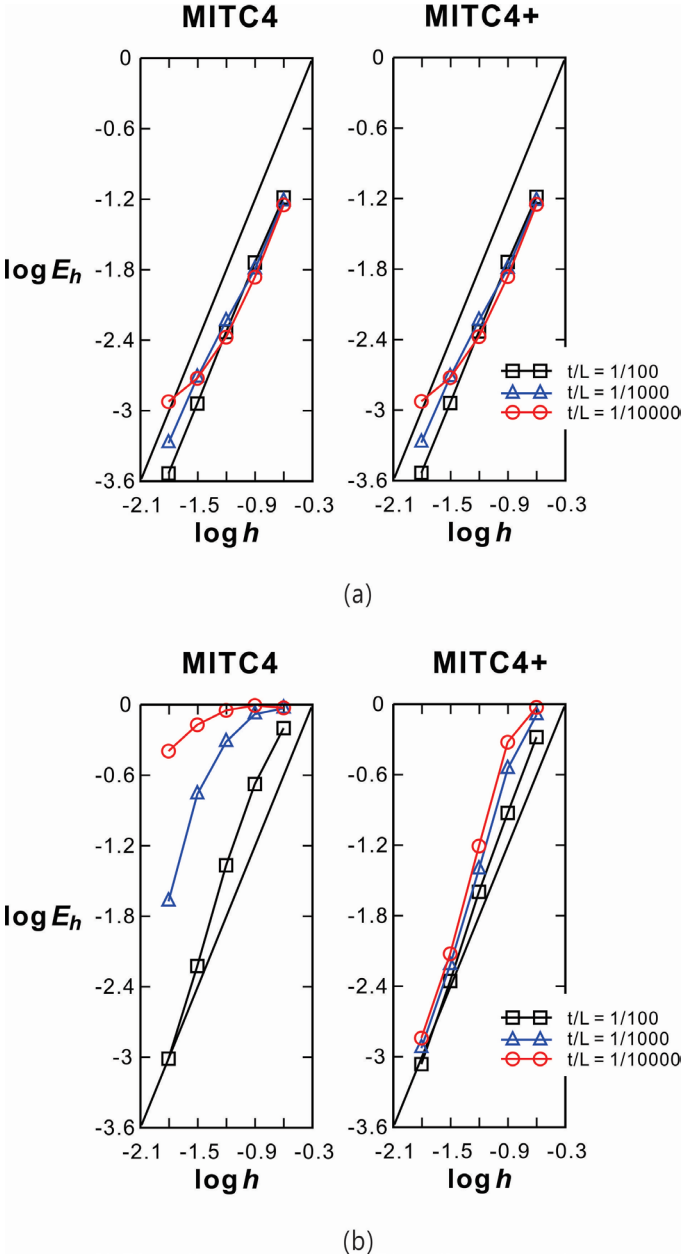


Fig. 2.11. Convergence curves for the free cylindrical shell problem with (a) the regular and (b) the distorted meshes.

Hyperboloid shell problems

It is important to consider a doubly curved shell and we solve the hyperboloid shell shown in Fig. 2.12 . The mid-surface of the shell structure is given by

$$x^2 + z^2 = 1 + y^2;\ y \in [-1,\quad 1]. \qquad (2.24)$$

and hence the shell contains two curvatures in opposite directions. As for the cylindrical shell problems, a smoothly varying pressure is applied.

A bending-dominated behavior is obtained with free ends and a membrane-dominated behavior is given with clamped ends, similar to the cases of the cylindrical shell.

Due to symmetry, only one-eighth of the structure corresponding to the shaded region ABCD in Fig. 2.12 is modeled for the analysis. For the membrane-dominated case, the clamped boundary condition is imposed: $u_z = \beta = 0$ along BC, $u_x = \beta = 0$ along AD, $u_y = \alpha = 0$ along DC, and $u_x = u_y = u_z = \alpha = \beta = 0$ along AB. For the bending-dominated case, the free boundary condition is imposed: $u_z = \beta = 0$ along BC, $u_x = \beta = 0$ along AD, and $u_y = \alpha = 0$ along DC.

In both cases, a 72×72 element mesh of MITC9 shell elements is used to obtain the reference solution. The solutions are calculated using $N \times N$ element meshes ($N = 4$, 8, 16, 32 and 64). The element size is $h = L/N$. For the clamped boundary condition, the regular mesh is graded in a boundary layer of width $6\sqrt{t}$ as shown in Fig. 2.12, see D Chapelle and KJ Bathe [A]. For the free boundary condition, the regular

mesh is graded in a boundary layer of width $0.5\sqrt{t}$ .We also perform the convergence studies with the distorted meshes shown in Fig. 2.12.

Fig. 2.13 shows the good convergence behavior of both elements in the solution of the clamped hyperboloid shell problem. Fig.2.14 shows the convergence curves for the solution of the free hyperboloid shell problem. While both shell elements behave well when using regular meshes, the convergence behavior of the MITC4 shell element deteriorates when using the distorted meshes. However, the new MITC4+ shell element displays an almost ideal convergence behavior.

Considering the *overall* predictive capabilities of the MITC4+ element, we see that in all these test problem solutions the MITC4+ shell element shows an almost optimal convergence behavior, even when the mesh is distorted in the bending-dominated situations.

Of course, the element might be further improved, to reach an even better convergence behavior, to simplify the formulation or to decrease the solution time, see for example Y. Ko, P.S. Lee and K.J. Bathe [B] and H.G. Choi and P.S. Lee [A]. In particular, the element may also use interpolation covers, see J. Kim and K.J. Bathe [A] and H. Jun, K. Yoon, P.S. Lee and K.J. Bathe [A], or be used as an overlapping finite element, like the elements for the analyses of solids described in Chapter 3.

An important further applicability of the MITC4+ element is the use of six degrees of freedom at each node, or at only certain nodes. Then the element can directly be coupled to beams (with 6 degrees of freedom at each beam element node) and also be used in the modeling of intersecting shells at significant angles. This development has been tackled in Y. Ko, X. Zhang and K.J. Bathe [A]. The resulting shell element shows all the important features of the MITC4+ element, that is, the element formulation is based on the fully coupled membrane and bending behaviors, passes the patch test, and shows good convergence behavior. Since the element also uses the interpolation for the additional rotational degrees of freedom (the drill degrees of freedom) a slightly improved membrane behavior is also obtained.

There are hence various research avenues available to enhance the MITC4+ shell element. Our objective was here to expose the fundamental ingredients in the formulation of the element – which might be further improved – and show the importance of appropriate testing, propose effective test problems and an error norm to use. While improvements in the element formulation and solution behavior can be expected to appear, the emphasis on appropriate testing and means to proceed (the use of the s-norm and the proposed test problems) can be expected to always be applicable.

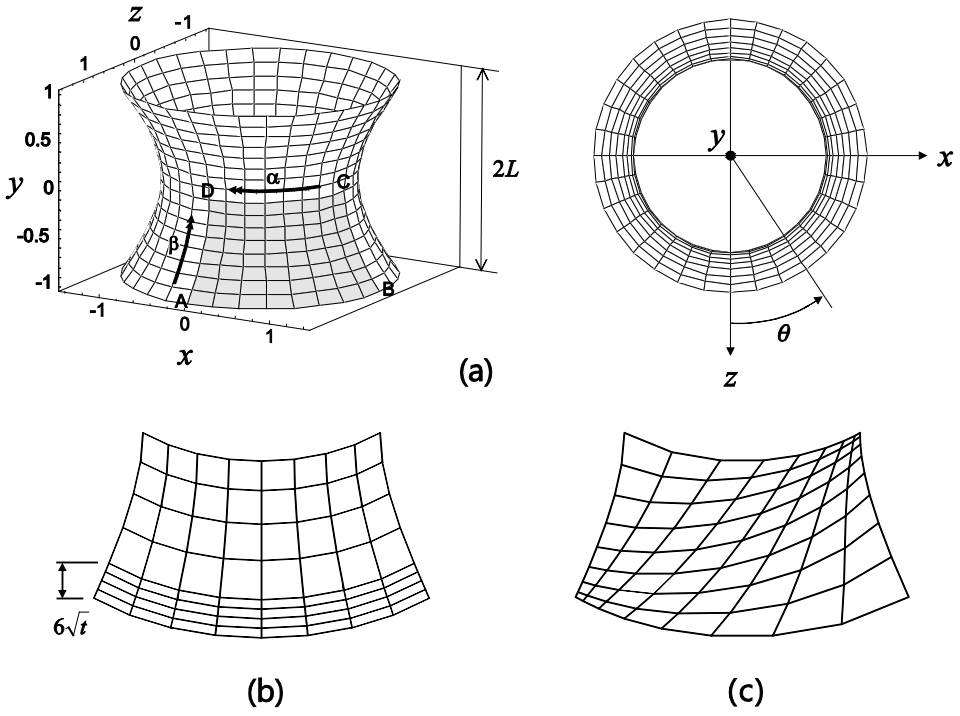
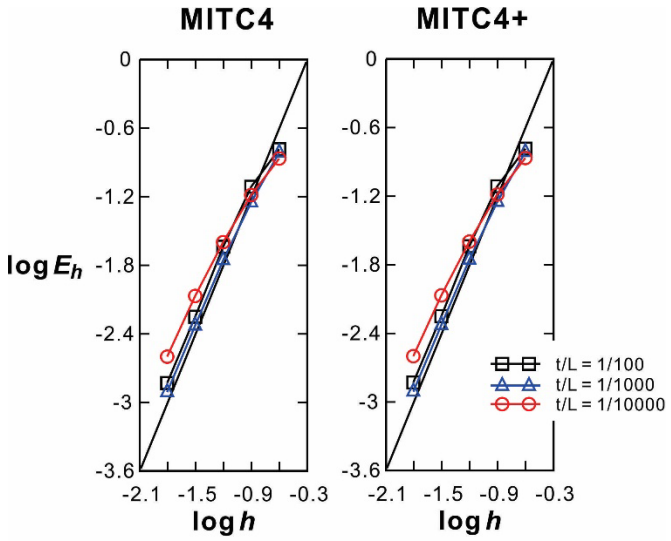
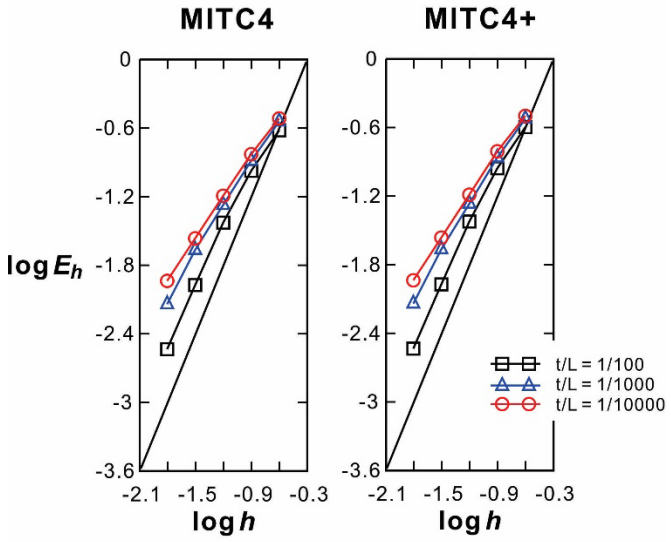


Fig. 2.12. Hyperboloid shell problem ($E = 2.0 \times 10^{11}$, $\nu = 1/3$, $L = 1.0$ and $p_0 = 1.0$). (a) Problem description. (b) Graded regular mesh for the clamped case (8×8 mesh). (c) Distorted mesh defined in Fig. 2.7 is used.



(a)



(b)

Fig. 2.13. Convergence curves for the clamped hyperboloid shell problem with (a) the graded regular and (b) the distorted meshes.

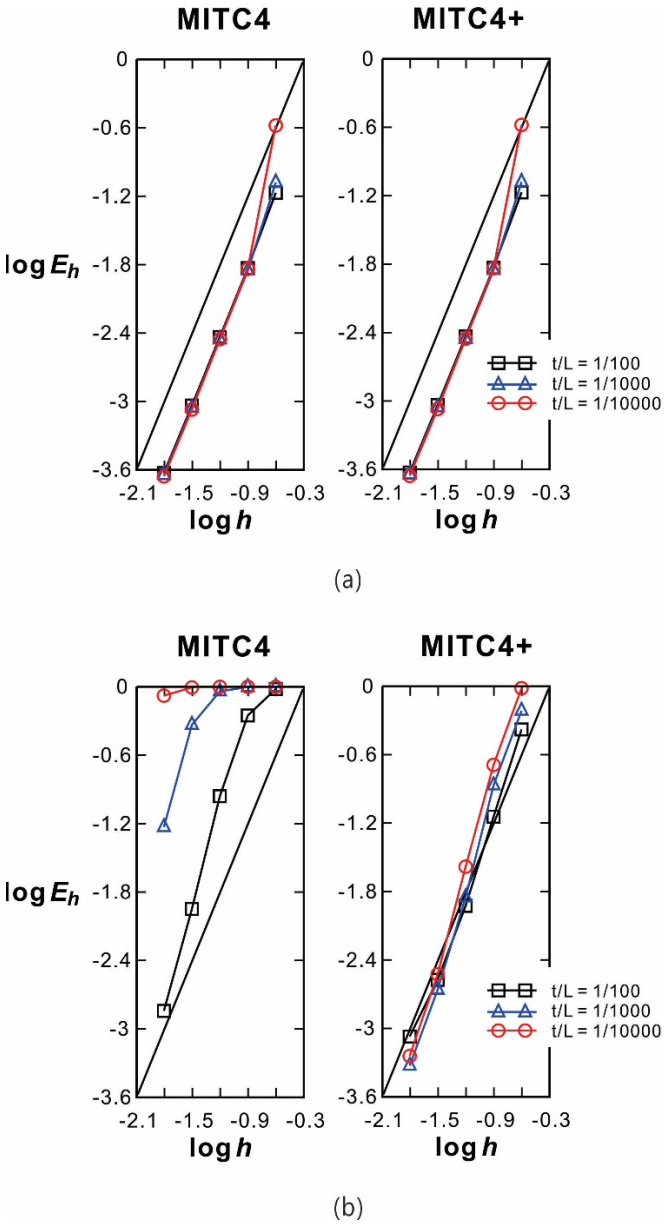


Fig. 2.14. Convergence curves for the free hyperboloid shell problem with (a) the regular mesh and (b) the distorted mesh.

2.3 The MITC3+ shell element

In practical engineering analyses, 4-node quadrilateral shell elements are mostly used and, in particular, if the elements can be geometrically distorted without serious deterioration of their performance. However, 3-node triangular shell elements can be of value if the use of 4-node elements would result in too distorted elements or if mesh grading needs to be used to calculate high stress gradients.

Since the MITC4+ interpolations are successful in removing shear and membrane locking for the 4-node element, it is natural to investigate whether also the formulation of the MITC3 shell element can be improved to reach a more effective 3-node element. Indeed, such improved formulation with enhanced solution capabilities is possible and resulted in the development of the MITC3+ shell element, Y. Lee, P.S. Lee and K.J. Bathe [A] and Y. Lee, H.M. Jeon, P.S. Lee and K.J. Bathe [A].

2.3.1 The formulation of the MITC3+ shell element for general shell analyses

The geometry interpolation of the standard 3-node shell element degenerated from continuum mechanics is like for the 4-node element, see Eq. (2.2)

$$\boldsymbol{x}(r,s,t) = \sum_{i=1}^{3} h_i(r,s)\boldsymbol{x}_i + \frac{t}{2}\sum_{i=1}^{3} a_i h_i(r,s)\boldsymbol{V}_n^i, \qquad (2.25)$$

where the $h_i(r,s)$ are the two-dimensional interpolation functions of the standard 3-node isoparametric element corresponding to node i, $\boldsymbol{x}_i$ is the position vector of node i in the global Cartesian coordinate

system, and a_i and $\boldsymbol{V}_n^i$ denote the shell thickness and the director vector at the node, respectively, see Fig. 2.15. In Eq. (2.25), we have $h_1 = 1 - r - s$, $h_2 = r$, $h_3 = s$, see K.J. Bathe [A].

Let us briefly recall the MITC3 shell element formulation, see also P.S. Lee and K.J. Bathe [D], because this element formulation provides the basis from which the improved element, the MITC3+ shell element, is reached.

Using Eq. (2.25), we obtain the displacement interpolation of the displacement-based 3-node element as

$$\boldsymbol{u}(r,s,t) = \sum_{i=1}^{3} h_i(r,s)\boldsymbol{u}_i + \frac{t}{2}\sum_{i=1}^{3} a_i h_i(r,s)(-\boldsymbol{V}_2^i\alpha_i + \boldsymbol{V}_1^i\beta_i), \tag{2.26}$$

in which $\boldsymbol{u}_i$ is the nodal displacement vector in the global Cartesian coordinate system, $\mathbf{V}_1^i$ and $\boldsymbol{V}_2^i$ are unit vectors orthogonal to $\boldsymbol{V}_n^i$ and to each other, and α_i, β_i are the rotations of the director vector $\boldsymbol{V}_n^i$ about $\mathbf{V}_1^i$ and $\boldsymbol{V}_2^i$, respectively, at the node i. Since the displacement-based element locks, we pursue the use of the MITC interpolation scheme to obtain a more effective element.

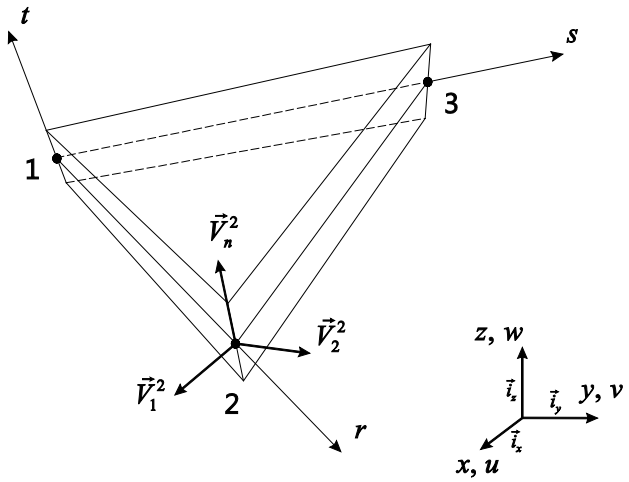


Fig. 2.15 Geometry of 3-node shell element, P.S. Lee and K.J. Bathe [D].

As for all displacement-based shell elements, the linear terms of the displacement-based covariant strain components are given by

$$e_{ij} = \frac{1}{2}(\boldsymbol{g}_i \cdot \boldsymbol{u}_{,j} + \boldsymbol{g}_j \cdot \boldsymbol{u}_{,i}), \tag{2.27}$$

in which $\boldsymbol{g}_i = \frac{\partial \boldsymbol{x}}{\partial r_i}$, $\boldsymbol{u}_{,i} = \frac{\partial \boldsymbol{u}}{\partial r_i}$ with $r_1 = r, r_2 = s, r_3 = t$.

Since the 3-node triangular shell finite element is flat, the covariant in-plane strain components are calculated using Eqs. (2.25) to (2.27). However, the covariant transverse shear strain field is established using the MITC scheme – as in the approach of formulating the MITC4 shell element.

The transverse shear strain field of the MITC3 shell element is based on assuming constant covariant transverse shear strain conditions along the edges, to have

$$\tilde{e}_{rt}^{MITC3} = e_{rt}^{(1)} + cs, \qquad \tilde{e}_{st}^{MITC3} = e_{st}^{(2)} - cr \tag{2.28}$$

where

$$c = (e_{rt}^{(3)} - e_{rt}^{(1)}) - (e_{st}^{(3)} - e_{st}^{(2)}) \tag{2.29}$$

and the tying points are shown in Fig. 2.16.

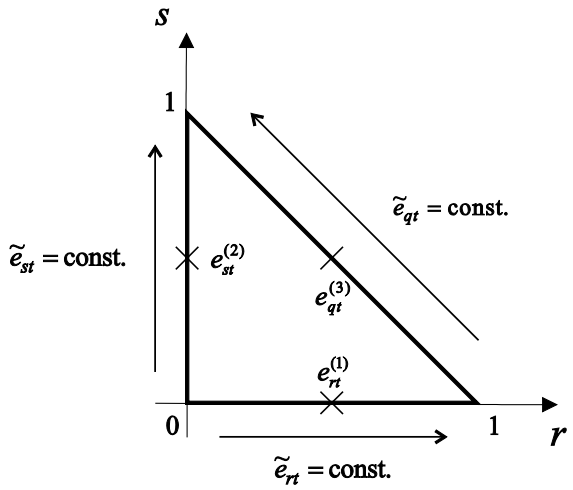
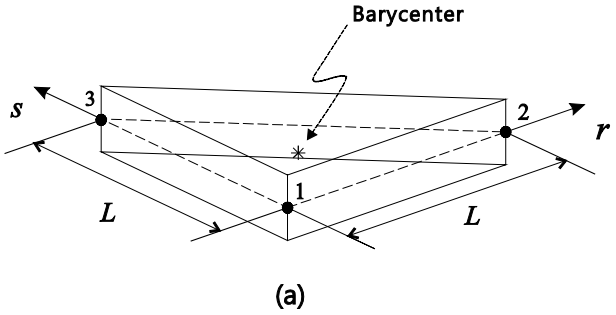
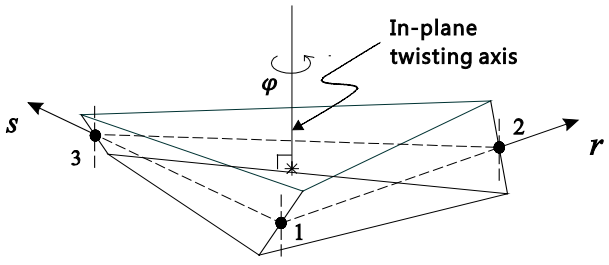


Fig. 2.16 Tying positions for the assumed transverse shear strain field of the MITC3 shell element. Along the edges, constant transverse shear strain conditions are imposed, P.S. Lee and K.J. Bathe [D].

While effective in some solutions, the above formulation needs to be improved, because the MITC3 element is too stiff for the use of larger element sizes. Specifically, the element is too stiff in the twisting mode shown in Fig. 2.17



(a)



(b)

Fig. 2.17 In-plane twisting mode of 3-node element. Regarding Figs. 2.17 to 2.32 and further information, see Y. Lee, P.S. Lee and K.J. Bathe [A].

Hence to reach a more effective element, we introduce a fourth node at the center of the element corresponding to a cubic bubble function $f_4 = 27rs(1-r-s)$. Correspondingly, the geometry of the MITC3+ shell element is given by

$$\boldsymbol{x}(r,s,t) = \sum_{i=1}^{3} h_i(r,s)\boldsymbol{x}_i + \frac{t}{2}\sum_{i=1}^{4} a_i f_i(r,s)\boldsymbol{V}_n^i \qquad (2.30)$$

with $$a_4\boldsymbol{V}_n^4 = \frac{1}{3}(a_1\boldsymbol{V}_n^1 + a_2\boldsymbol{V}_n^2 + a_3\boldsymbol{V}_n^3),$$

in which the $f_i(r,s)$ are two-dimensional interpolation functions that include the cubic bubble function f_4 corresponding to the internal node 4

$$f_1 = h_1 - \frac{1}{3}f_4, \quad f_2 = h_2 - \frac{1}{3}f_4, \qquad f_3 = h_3 - \frac{1}{3}f_4, \qquad (2.31)$$

The displacement interpolation of the MITC3+ shell element is thus

$$\boldsymbol{u}(r,s,t) = \sum_{i=1}^{3} h_i(r,s)\boldsymbol{u}_i + \frac{t}{2}\sum_{i=1}^{4} a_i f_i(r,s)(-\boldsymbol{V}_2^i\,\alpha_i + \boldsymbol{V}_1^i\beta_i), \qquad (2.32)$$

in which α_4 and β_4 are the rotational degrees of freedom at the bubble node, which carries no displacement degrees of freedom.

The bubble node is positioned on the flat element surface defined by the three corner nodes of the element. The purpose of introducing the bubble function is to enrich the bending and transverse shear strain fields. Since the bubble function pertains only to the element considered, the bubble degrees of freedom can be condensed out on the element level after forming the element matrices.

For the MITC3+ transverse shear strain interpolation, we identify modes of constant and linearly varying shear strain contributions and interpolate the total shear strains as

$$\hat{e}_{rt} = \frac{2}{3}(e_{rt}^{(B)} - \frac{1}{2}e_{st}^{(B)}) + \frac{1}{3}(e_{rt}^{(C)} + e_{st}^{(C)}) + \frac{1}{3}\hat{c}(3s - 1) \qquad (2.33)$$

$$\hat{e}_{st} = \frac{2}{3}(e_{st}^{(A)} - \frac{1}{2}e_{rt}^{(A)}) + \frac{1}{3}(e_{rt}^{(C)} + e_{st}^{(C)}) + \frac{1}{3}\hat{c}(1 - 3r) \qquad (2.34)$$

where

$$\hat{c} = (e_{rt}^{(F)} - e_{rt}^{(D)}) - (e_{st}^{(F)} - e_{st}^{(E)}). \qquad (2.35)$$

and the sampling points (A) to (F) are shown in Fig. 2.18. The *rt*- and *st*-components of the strains are calculated from the *1t- , 2t-* and *3t-* components referred to in Fig. 2.18 (b).The details of the derivation are given in Y. Lee, P.S. Lee and K.J. Bathe [A] , where also the distance *d*, mentioned in Fig. 2.18 (c) is determined. The value of *d* needs to be small (like *d* = 1/10,000) to avoid excessive stiffness in the twisting mode shown in Fig. 2.17(b) but not be zero because then the element matrix contains a spurious zero eigenvalue, see Fig. 2.19.

An important point is that the MITC3+ shell element passes all “basic numerical tests”, that is, the element is spatially isotropic, it does not contain any spurious zero energy mode, and passes the membrane, bending and shearing patch test (see above similar comments regarding the MITC4+ shell element).

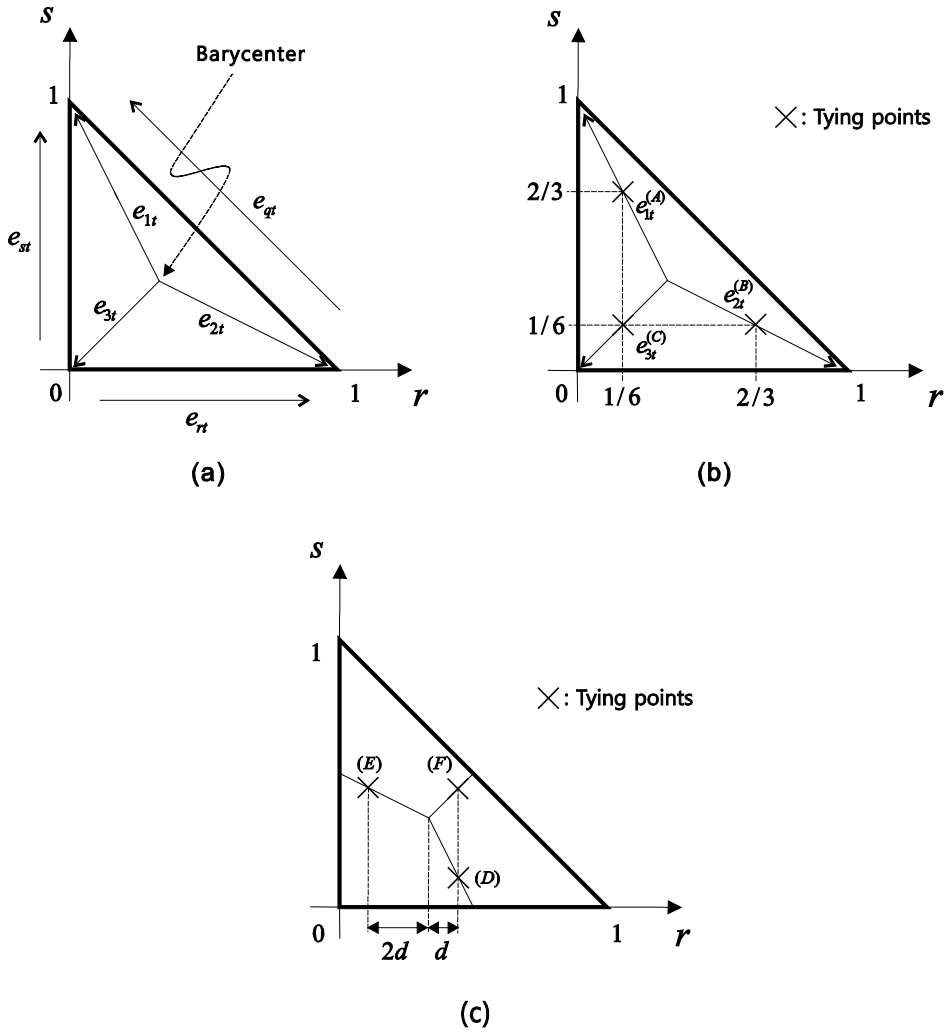


Fig. 2.18 Transverse shear strain components e_{1t}, e_{2t} and e_{3t}, the covariant components are measured in the directions shown and at the tying positions (A), (B), (C), (D), (E) and (F) for the MITC3+ assumed transverse shear strain field.

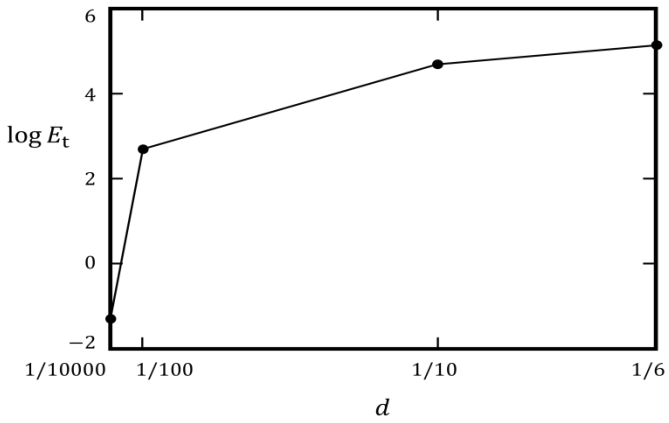


Fig. 2.19 Strain energy in twisting mode as a function of distance *d* in Fig. 2.1.

Since the MITC3+ shell element contains the bubble function, the calculation of the element matrices is computationally more expensive than using the MITC3 element. For the MITC3+ element, 7-point Gauss integration might be used to calculate the element matrices but using 3-point Gauss integration all basic numerical tests are also passed and similar convergence behaviors are observed.

Some additional insight into the behavior of the element is given in Lee Y, Jeon HM, Lee PS, and KJ Bathe [A], and the MITC3+ shell element is available in the ADINA program, see Bentley Systems, Inc. [A].

2.3.2 The testing of the MITC3+ element

Since – as with the MITC4+ shell element – a deep mathematical analysis of the MITC3+ shell element for its general convergence properties in complex shell analyses seems out of reach, we need to test the element numerically. As we discussed above, a "proper testing" needs to be conducted and for that purpose we perform the same tests as for the MITC4+ shell element, and we also use the generally applicable s-norm to measure convergence, see J.F. Hiller and K.J. Bathe [A] and D. Chapelle and K.J. Bathe [B].

Considering Eq. (2.22), for a 3-node shell element to be uniformly optimal, the value of C must be constant, in particular, independent of the shell thickness, and k should be equal to 2.

In the figures given below, the optimal convergence rate is in all cases shown by a bold line, and well-converged reference solutions obtained using fine meshes of the MITC9 shell elements are used, because the MITC9 shell element satisfies the ellipticity and consistency conditions and shows a good convergence behavior, see K.J. Bathe, P.S. Lee and J.F. Hiller [A].

Fully clamped square plate problem

We solve the plate bending problem shown in Fig. 2.20. We solved the same problem using the MITC4+ element, see Fig. 2.6, and refer to the description given there, also regarding the boundary conditions used.

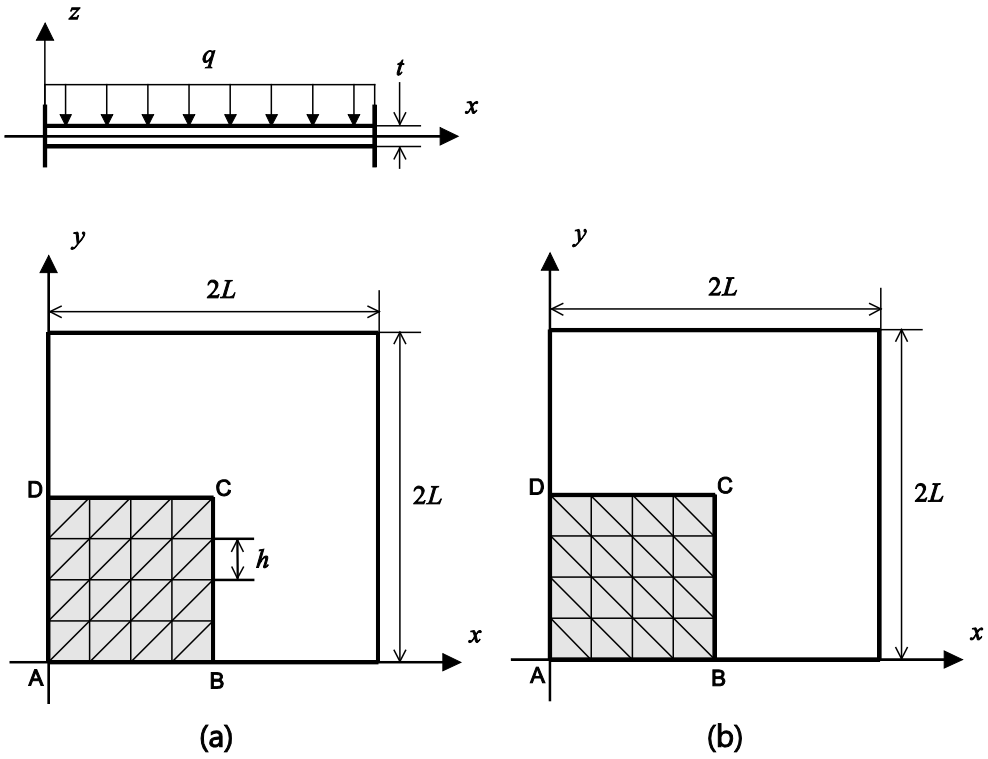


Fig. 2.20 Fully clamped square plate problem under uniform pressure p ($L = 1.0$, $E = 1.7472 \times 10^7$, p =1 and $\nu = 0.3$) with 2 different 4×4 mesh patterns in (a) and (b).

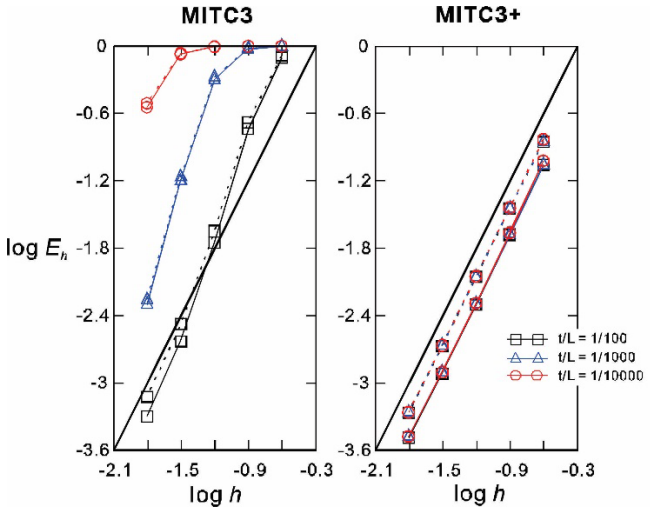


Fig. 2.21 Convergence curves for the fully clamped square plate problem. The solid and dotted lines correspond to the results obtained using the mesh patterns in Figs. 2.20 (a) and (b), respectively. Y. Lee, P.S. Lee and K.J. Bathe [A]

Figure 2.21 gives the convergence curves of the MITC3 and MITC3+ elements. A uniform mesh of 96×96 MITC9 shell elements is used to obtain the reference solution. We use $N \times N$ element meshes ($N = 4$, 8, 16, 32, and 64) to calculate the solutions. The element size in the convergence curves is $h = L/N$. The performance of the MITC3+ shell element is very good and, practically, uniformly optimal. Note that for the triangular elements we use two different mesh patterns, see Fig. 2.20, to see whether the mesh pattern affects the convergence of the element. Fig. 2.21 shows that the response is hardly affected using either element. More generally, provided reasonably distorted meshes are employed in the solution of this problem, the results using the MITC3+ element are hardly affected.

Cylindrical shell problems

We consider the cylindrical shell of length $2L$, radius R and uniform thickness t as shown in Fig. 2.22. The loading is a smoothly varying pressure $p(\theta) = p_0 \cos(2\theta)$. We solved this problem already using the MITC4+ element and obtained the results given in Figs. 2.9 to 2.11.

As pointed out earlier, when both ends are free, a bending dominated problem is solved, and when both ends are clamped, a membrane dominated problem is considered.

As in the solution using the MITC4+ element, using symmetry, we only model the region ABCD in Fig. 2.22, and impose the boundary conditions as in that solution.

Figs. 2.23 and 2.24 give the convergence curves of the MITC3 and MITC3+ shell elements for the clamped and free cylindrical shell problems, where we did not grade the mesh in the boundary layers (like we did in the testing of the MITC4+ element) because good results are obtained without special attention to the boundary layers. The reference solutions are calculated using a 96×96 element mesh of MITC9 shell elements. The solutions using the MITC3 and MITC3+ shell elements are obtained with non-distorted $N \times N$ element meshes ($N = 4$, 8, 16, 32, and 64). The element size is $h = L/N$. In these problem solutions, using non-distorted meshes, the MITC3 and MITC3+ shell elements show similarly good convergence behaviors.

We also give the results of convergence studies using the distorted meshes shown in Fig. 2.25. When an $N \times N$ element mesh is used, each edge is discretized in the following ratio: $L_1 : L_2 : L_3 : \ldots\ldots L_N = 1 : 2 : 3 : \ldots\ldots N$. Figs. 2.26 and 2.27 show the convergence curves for the clamped and the free shell structures,

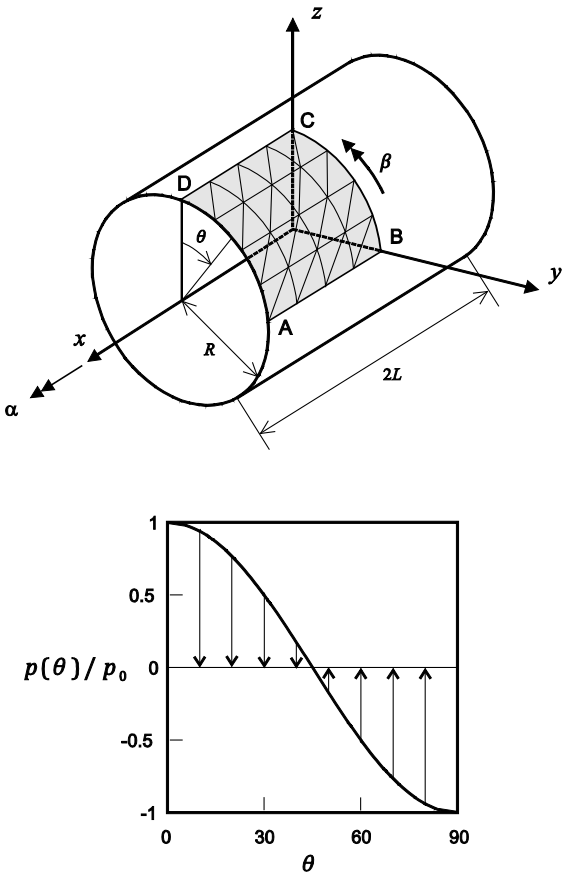


Fig. 2.22. Cylindrical shell problem (4×4 mesh, $L = R = 1.0$, $E = 2.0 \times 10^5$, $\nu = 1/3$ and $p_0 = 1.0$)

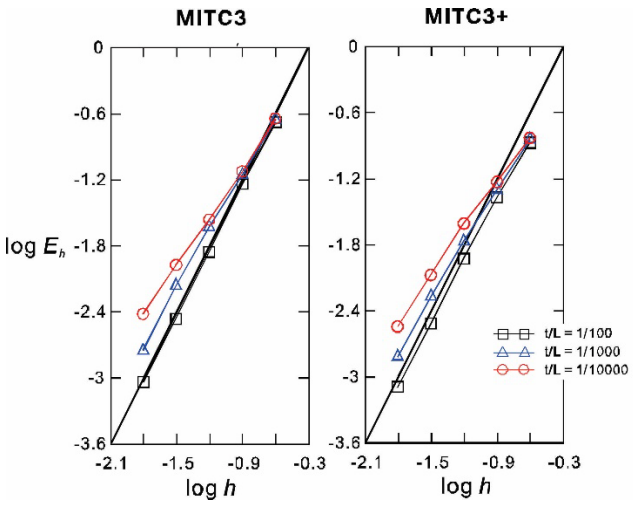


Fig. 2.23. Convergence curves for the clamped cylindrical shell problem using non-distorted meshes.

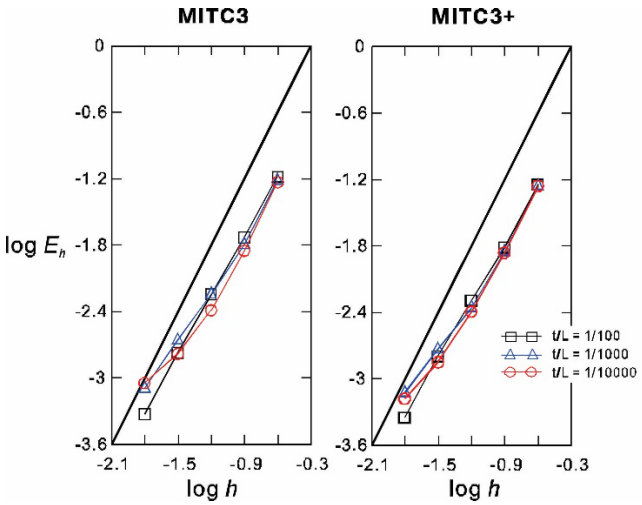


Fig. 2.24. Convergence curves for the free cylindrical shell problem using non-distorted meshes.

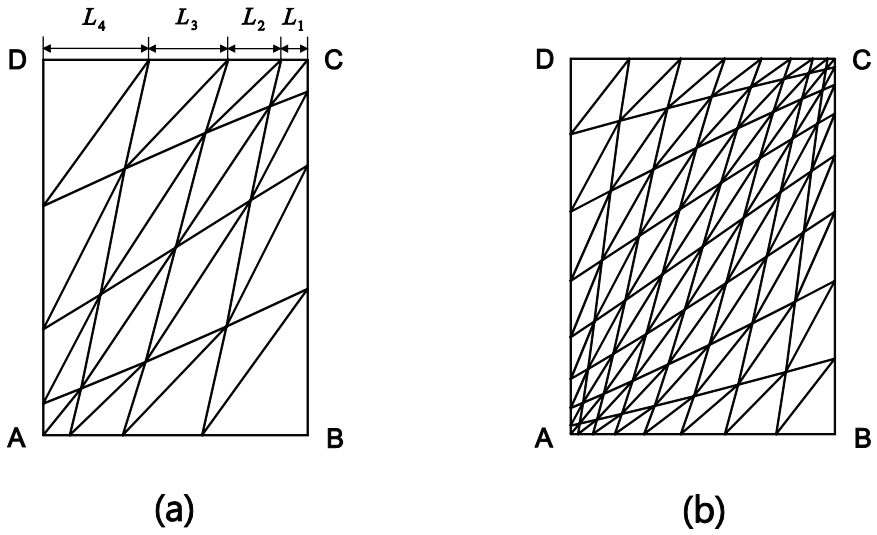


Fig. 2.25. Distorted mesh patterns (a) for $N = 4$ and (b) for $N = 8$.

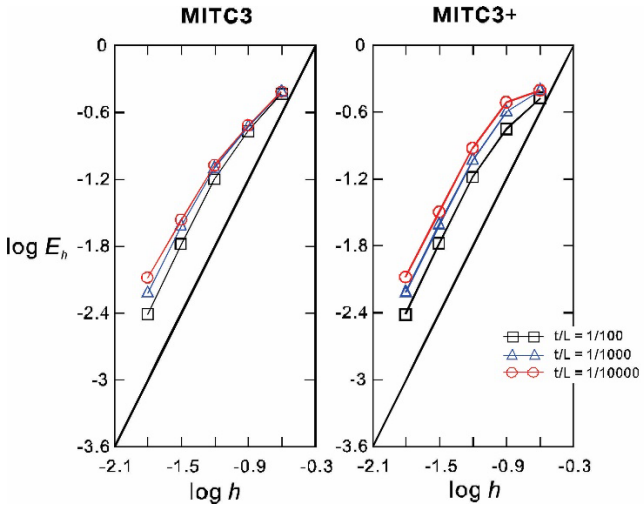


Fig. 2.26. Convergence curves for the clamped cylindrical shell problem with the distorted meshes shown in Fig. 2.25.

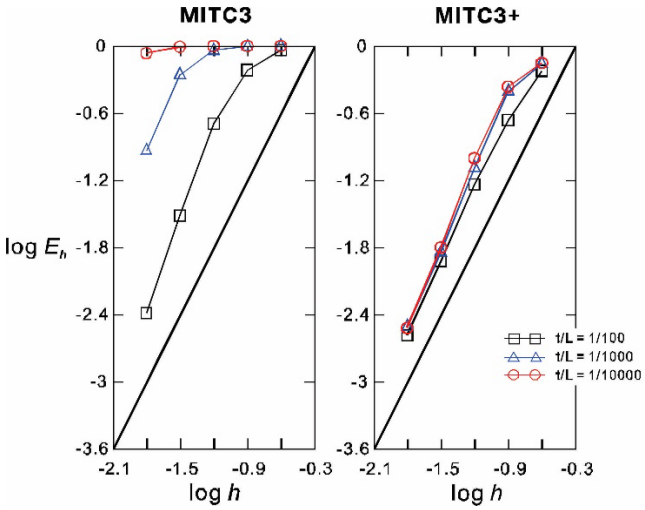


Fig. 2.27. Convergence curves for the free cylindrical shell problem with the distorted meshes shown in Fig. 2.25.

respectively. When using the distorted meshes, the MITC3+ shell element continues to show an excellent performance and significantly better than the MITC3 element for the free edge condition, see Fig. 2.27.

Hyperboloid shell problems

As in Section 2.2.3, we consider the hyperboloid shell shown in Fig. 2.28. The mid-surface of the shell structure is given by $x^2 + z^2 = 1 + y^2;\ \ y \in [-1,\ \ 1\ ]$ and the smoothly varying applied pressure is $p(\theta) = p_0\, cos(\,2\theta)$.

A membrane dominated behavior is given with clamped ends and a bending dominated behavior is obtained with free ends.

Due to symmetry, the analyses are performed using again one-eighth of the structure corresponding to the shaded region ABCD in Fig. 2.28 (and Fig. 2.12). We use the same boundary conditions as for the discretizations with the MITC4+ shell element.

In both cases, a 96×96 element mesh of MITC9 shell elements is used to obtain the reference solutions. The solutions of the MITC3 and MITC3+ shell elements are calculated using $N \times N$ element meshes ($N =$ 4, 8, 16, 32 and 64). The element size is $h = L/N$. In the clamped hyperboloid shell case, a boundary layer of width $6\sqrt{t}$ is considered for half of the mesh, see Fig. 2.28. For the free hyperboloid shell case, the thin boundary layer is not specially considered because good results are obtained using the MITC3+ element even when not modeling the boundary layer.

Figs. 2.29 and 2.30 show the convergence curves for the clamped and free hyperboloid shell problems when non-distorted meshes are used. We see that in the free hyperboloid shell case, the MITC3+ shell element shows much better convergence behavior than the MITC3 shell element.

Figs. 2.31 and 2.32 give the convergence curves of the MITC3 and MITC3+ shell elements when the distorted meshes in Fig. 2.25 are used. Even when the meshes are distorted, the MITC3+ shell element displays an excellent convergence behavior.

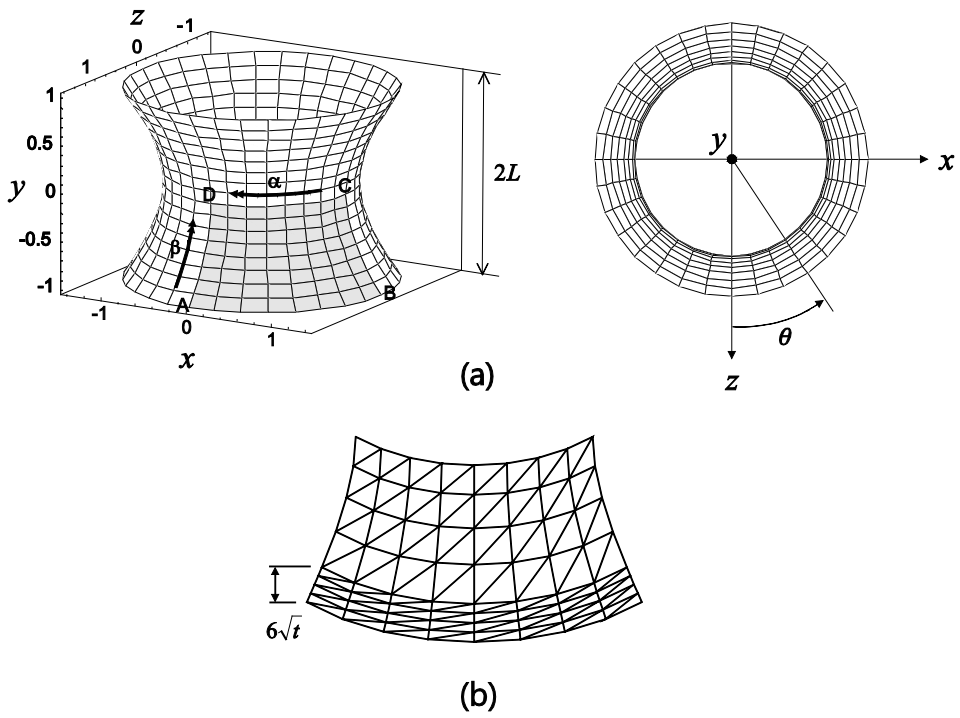


Fig. 2.28. Hyperboloid shell problem ($E = 2.0 \times 10^{11}$, $\nu = 1/3$ and $p_0 = 1.0$). (a) Problem description. (b) Graded mesh for the clamped case (8×8 mesh).

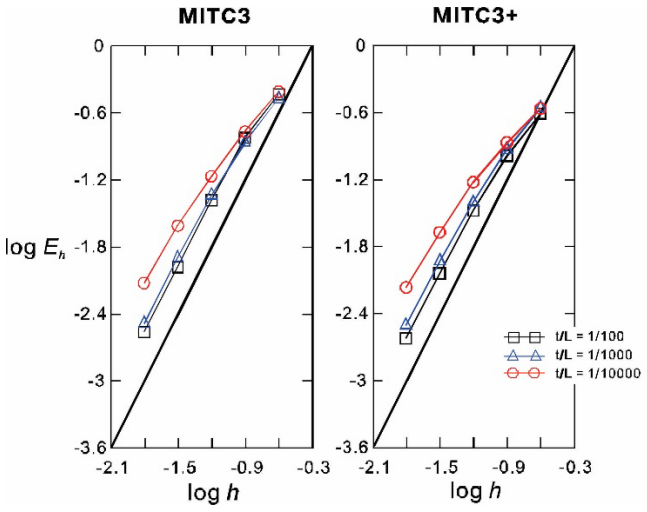


Fig. 2.29. Convergence curves for the clamped hyperboloid shell problem using non-distorted meshes.

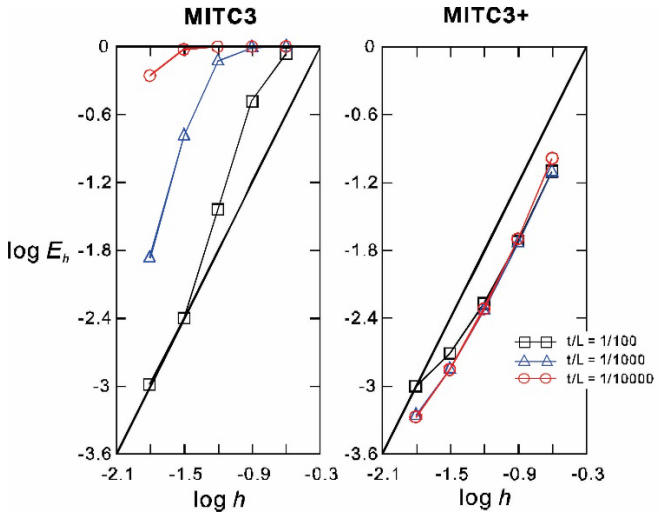


Fig. 2.30 Convergence curves for the free hyperboloid shell problem using non-distorted meshes.

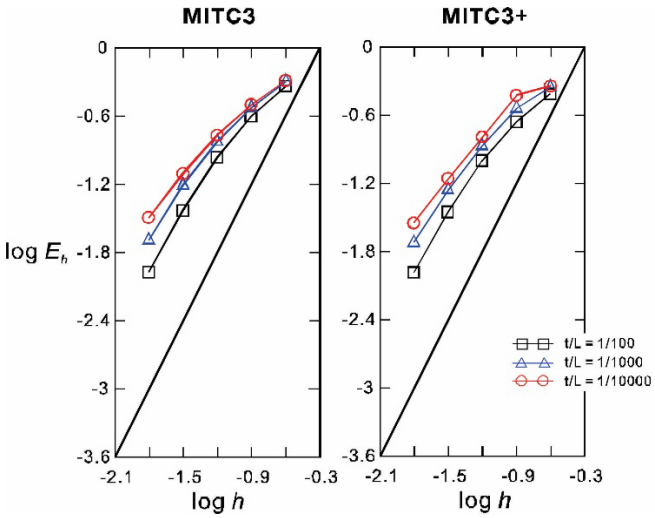


Fig. 2.31 Convergence curves for the clamped hyperboloid shell problem with the distorted meshes shown in Fig. 2.25.

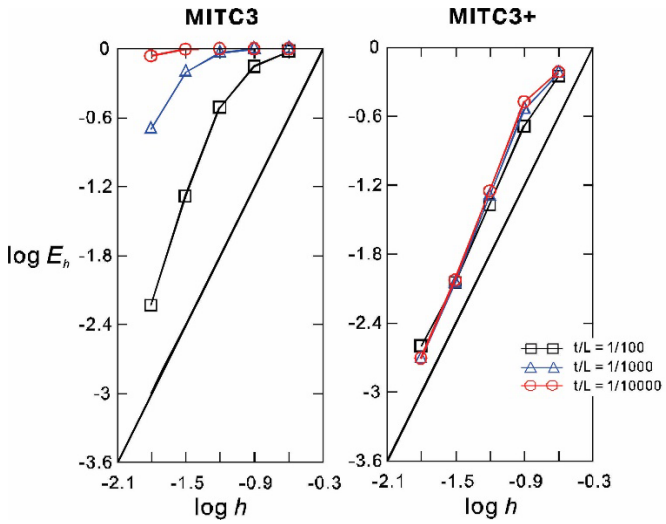


Fig. 2.32 Convergence curves for the free hyperboloid shell problem with the distorted meshes shown in Fig. 2.25.

As for the MITC4+ shell element presented in Section 2.2, improvements in the shell elements can surely be reached. Different tying schemes may be used, interpolation covers may be employed, see J. Kim and K.J .Bathe [A, B] and H. Jun, K. Yoon, P.S. Lee and K.J. Bathe [A], strain smoothing techniques may be used, see C Lee and PS Lee, simplifications of the element formulations may be reached, the applicability of the MITC3+ element may be increased by introducing the drilling rotational degree of freedom, as for the MITC4 elements, and the element may be used in the formulation of overlapping elements, see Chapter 3.

The aim in the presentation of the MITC3+ and MITC4+ elements was to give the procedures used to apply the MITC scheme for the formulation of the elements, and to show that effective shell elements have been reached. The elements are inherently stable, the basic tests are passed, and the solutions of the recommended shell analysis problems to measure convergence show very good convergence behaviors for non-distorted and distorted meshes.

In fact, to have stability of the elements, without the use of numerical factors, and very good convergence behaviors as displayed by the MITC3+ and MITC4+ shell elements, makes these elements excellent candidates for practical use. We might recall here that other elements, like those based on reduced integration, or on "enhancing strains" may not be inherently stable and need stabilization schemes. A particular example of such elements are the "incompatible mode elements", see E.L Wilson and A. Ibrahimbegovic [A], and J. Simo and S. Rafai [A], which can be valuable but contain instabilities and need to be used with care, see T. Sussman and K.J. Bathe [A], and Y. Ko and K.J. Bathe [A].

2.4 A three-dimensional MITC element for shells – the 3D-MITC8/s element

While the MITC approach of formulating elements was "invented" for the analysis of shells, first to remove shear locking and then also element distortion effects and membrane locking, the scheme can of course also be applied to formulate two-dimensional and three-dimensional elements for the analysis of solids and fluids. For some developments with that objective, see for example, E. N. Dvorkin and S.I. Vassolo [A], R. Radovitzky and E.N. Dvorkin [A], T. Sussman and K.J. Bathe [B], and Y. Ko, P.S. Lee and K.J. Bathe [B]. We consider below the development of a three-dimensional MITC element for shell analyses.

2.4.1 The formulation of the three-dimensional 3D-MITC8 and 3D-MITC8/s elements

It is of value to focus on the formulation of a three-dimensional solid element because such formulation also reveals how we may "physically think" when pursuing any MITC element development. Indeed, physical thinking was the basis of the development of the MITC4 shell element where for the Dvorkin-Bathe *Ansatz* (to model the transverse shear behavior of the shell element), the four edges of the shell element were thought of as four 2-node isoparametric beams connected at the shell element corner nodes. Since the 2-node displacement-based beam element does not lock in shear when the transverse shear is assumed to be constant along the length of the beam, the transverse shear for the MITC4 shell element is assumed to be constant along the edges and varying linearly across the shell element. We use such physical approach in the development of the three-dimensional 3D-MITC8 element.

Consider the 8-node three-dimensional element shown in Fig. 2.33. This element is frequently formulated as a displacement-based element but because it is then too stiff in bending and shear actions, the formulation is amended to include incompatible modes or enhanced strains. Also, reduced numerical integration is used, see e.g. KJ Bathe [A]. But these remedies may cause instabilities in the element behavior that need to be addressed with stabilization schemes.

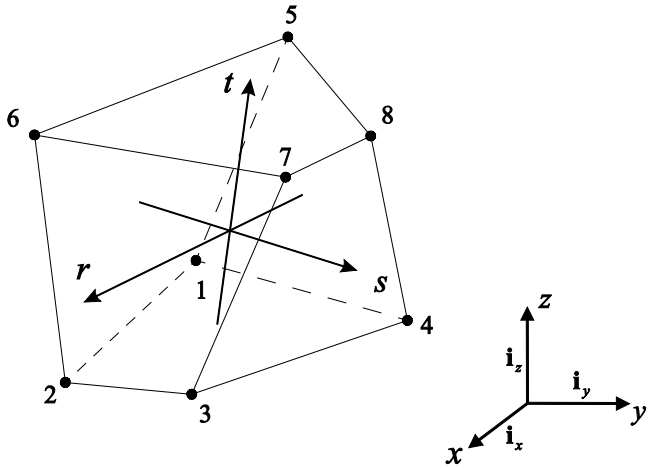


Fig. 2.33 A standard 8-node hexahedral three-dimensional solid element, Y. Ko and K.J. Bathe [A].

As mentioned already in Sections 2.2 and 2.3, and demonstrated in those sections, using the MITC approach of formulation can result in elements that are stable and show good convergence behavior.

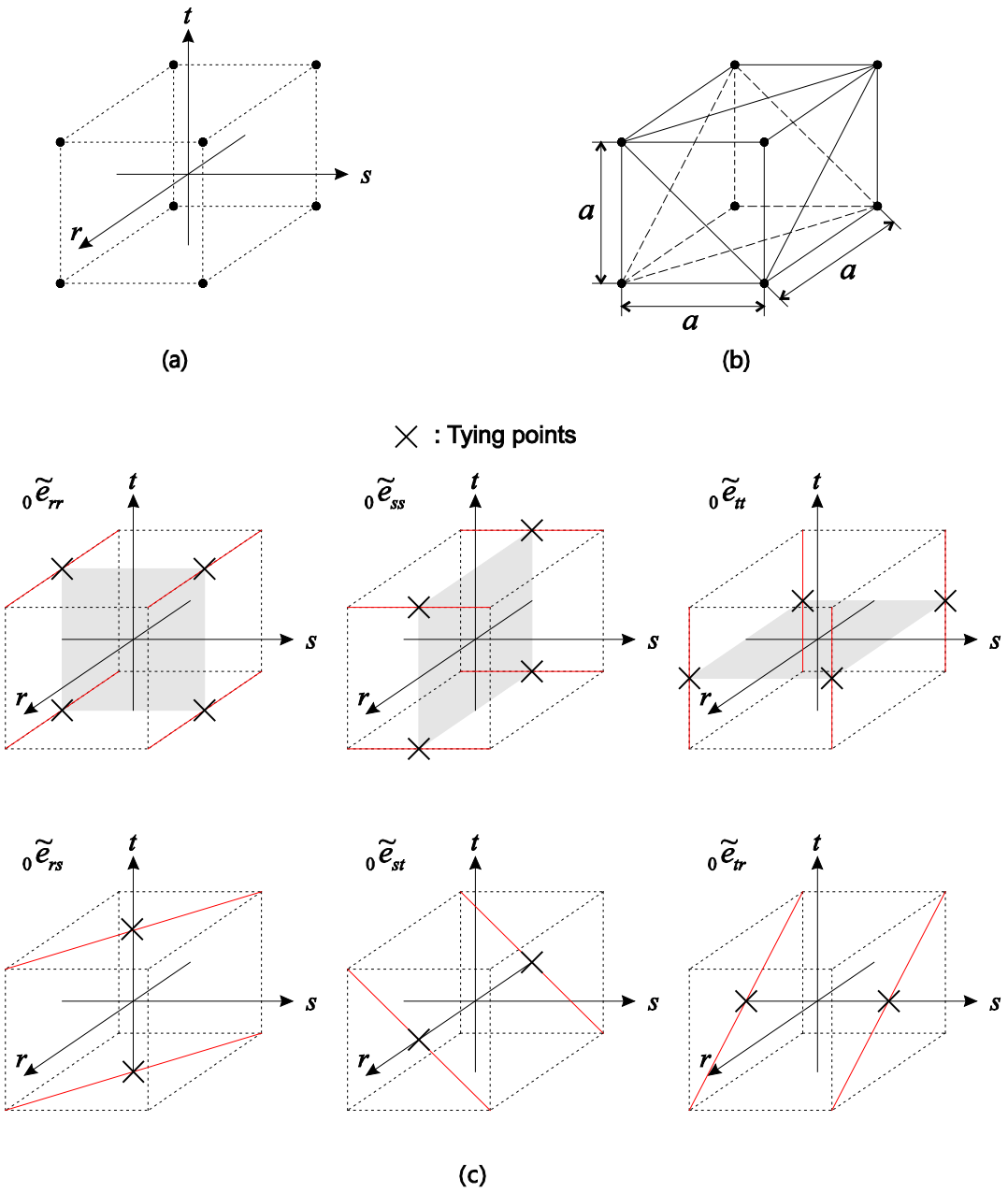


Fig. 2.34 Selection of tying strains and positions for the 3D-MITC8 element, Y. Ko and K.J. Bathe [A]. (a) The eight nodes or "joints" on the solid. (b) A stable truss structure of eighteen 2-node truss elements. (c) Tying positions for the assumed strain components.

One way to proceed in the formulation of the three-dimensional MITC element is to think physically of a truss structure replacing the solid shown in Fig. 2.33. If the truss structure consists of *the minimum number of 2-node truss elements to be stable, for any loading applied, we can expect the structure to not be too stiff when subjected to any loading*, hence a "locking stiffness" might not be observed. Such a truss structure is shown in Fig. 2.34 and we use this structure to formulate the three-dimensional 8-node MITC element.

Let us note that the truss structure contains one 2-node element for each edge of the solid, a total of twelve 2-node elements. These elements will carry the normal forces (that is, normal stresses) applied to the truss structure (representing the 8-node solid element). In addition, the truss structure has one diagonal 2-node truss element on each of the six faces of the structure. Clearly, this truss structure is stable with the minimum number of 2-node elements. Using more 2-node elements in the structure (for example, placing more diagonals on the faces) would render the structure stiffer and over-constrain its behavior for our purposes.

Our aim is now to project the behavior of the truss structure in Fig. 2.34 onto the solid element of Fig. 2.33. For this purpose we use the forces in the truss elements (with each force constant in each 2-node element) for a "judicious" interpolation of the stresses over the solid element in the following way:

For bending with the normal stress activated in the r-direction, we interpolate the r-direction normal strain linearly over the st-plane. This prevents locking in bending for that bending action. We similarly interpolate the normal strains in the s- and t-directions, see Fig. 2.34. This procedure corresponds to using the forces in the truss structure of the 2-node edge elements and interpolating these forces over the solid domain.

The shearing actions are resisted by six 2-node diagonal elements in the truss structure and hence we interpolate the shearing strains linearly over the solid as shown in Fig.2.34.

Altogether, we have the following equations for the assumed strain field of the 3D-MITC8 element

$$\tilde{e}_{rr} = A_{rr}^0 + A_{rr}^1 s + A_{rr}^2 t + A_{rr}^3 st$$
$$\tilde{e}_{ss} = A_{ss}^0 + A_{ss}^1 t + A_{ss}^2 r + A_{ss}^3 tr$$
$$\tilde{e}_{tt} = A_{tt}^0 + A_{tt}^1 r + A_{tt}^2 s + A_{tt}^3 rs \qquad (2.36)$$
$$\tilde{e}_{rs} = A_{rs}^0 + A_{rs}^1 t$$
$$\tilde{e}_{st} = A_{st}^0 + A_{st}^1 r$$
$$\tilde{e}_{tr} = A_{tr}^0 + A_{tr}^1 s$$

in which the A_{ij}^k are the unknown strain coefficients. The constants A_{ij}^k (k = 0, 1, 2, 3 or 0, 1) and corresponding interpolations allow for overall stability in linear analyses. The details of how these constants are determined using the concepts discussed above are given in Y. Ko and K.J. Bathe [A]. However, while this Ansatz is good for linear analysis, if the element shall also be employed in nonlinear analysis, it is necessary to include additional terms in the interpolations of Eq. (2.36). These terms are identified and various solutions to linear and nonlinear problems using the 3D-MITC8 element are given in Y. Ko and K.J. Bathe [A], where the results are also compared to those obtained with the displacement-based element without and with incompatible modes.

The above considerations address the prevention of locking in bending and shear actions. If the three-dimensional element is to be used in the analysis of shells, it is also necessary to prevent "locking in pinching" because the normal strain through the element thickness is included in the three-dimensional formulation. To also prevent locking in pinching, we simply interpolate the normal strain through the element shell thickness as, see D. Chapelle and K.J. Bathe [B] and Y. Ko and K.J. Bathe [A],

$$\tilde{e}_{tt} = \frac{1}{4}(1-r)(1-s)e_{tt}^{(-1,-1,0)} + \frac{1}{4}(1+r)(1-s)e_{tt}^{(1,-1,0)} + \frac{1}{4}(1+r)(1+s)e_{tt}^{(1,1,0)} + \frac{1}{4}(1-r)(1+s)e_{tt}^{(-1,1,0)} \qquad (2.37)$$

where we use the *t*-direction as the normal direction and the superscripts denote the (*r, s, t*) positions in the shell element. We refer to the resulting element as the 3D-MITC8/s element, the "/s" signifying "shell".

2.4.2 Inf-sup testing of three-dimensional elements considering shell solutions

Inf-sup testing is a very general procedure for the evaluation of finite element formulations and used to identify whether a formulation performs well, for example, whether a formulation does not lock in the analysis of incompressible media. In plate and shell analyses, the formulation should not lock, with an emphasis on shear and membrane actions.

It is of particular value to test the 3D-MITC8/s element for locking in shell analyses. This is achieved by establishing and evaluating the inf-sup condition. Let the three-dimensional shell problems using the element for solution be in general given by, see K.J. Bathe [A] and F. Brezzi and K.J. Bathe [A]

$$\text{Find } \mathbf{v} \text{ such that } a(\mathbf{w},\mathbf{v}) = (\mathbf{w},\mathbf{f}) \quad \forall \mathbf{w} \tag{2.38}$$

where $a(\cdot,\cdot)$ and $(\cdot,\cdot)$ denote the applicable bilinear and linear forms, $\mathbf{f}$ represents the loading, and the solution $\mathbf{v} \in V$ and the variation $\mathbf{w} \in V$ are members of the following solution space

$$V = \{\, \mathbf{v} \mid \mathbf{v} \in L^2(\Omega), \frac{\partial v_i}{\partial x_j} \in L^2(\Omega), v_i = 0 \text{ on } S_U \text{ with } i,j = 1,2,3 \,\} \tag{2.39}$$

Here we use the notation $v_1 = v_x$, $v_2 = v_y$, $v_3 = v_z$ to denote the components of the displacement vector $\mathbf{v}$ in the global Cartesian coordinates $x_1 = x$, $x_2 = y$, $x_3 = z$, and S_U is the surface on which the displacements are prescribed to be zero.

The finite element discretization using the 3D-MITC8/s element then gives the finite-dimensional problem

$$\text{Find } \bar{\mathbf{v}}_h \text{ such that } \bar{\mathbf{w}}_h^T \bar{\mathbf{A}}_h \bar{\mathbf{v}}_h = \bar{\mathbf{w}}_h^T \bar{\mathbf{F}}_h \quad \forall \bar{\mathbf{w}}_h \tag{2.40}$$

where $\bar{\mathbf{A}}_h$ is the matrix representation of the mesh using the element formulation (the stiffness matrix **K**) and $\bar{\mathbf{F}}_h$ is the vector of external forces (the load vector **R**) , h denotes the "average element" size used in the discretization, and we have that the solution vector $\bar{\mathbf{v}}_h$ and vector $\bar{\mathbf{w}}_h$ are vectors of the nodal degrees of freedom of the finite element solution space.

The condition of continuity on the discretization using the 3D-MITC8/s element (and the other elements considered below in the figures) to solve a problem is satisfied. The reliability of the elements is then checked by studying the inf-sup condition given by

$$\inf_{\bar{\mathbf{w}}_h} \sup_{\bar{\mathbf{v}}_h} \frac{\bar{\mathbf{w}}_h^T \mathbf{A}_h \bar{\mathbf{v}}_h}{\|\bar{\mathbf{w}}_h\| \|\bar{\mathbf{v}}_h\|} \geq \gamma_A \tag{2.41}$$

and hence we focus on this condition. In essence, the condition requires that as a mesh is refined, the value of the inf-sup expression will not continue to decrease but be bounded from below by a positive constant, which we denote here by γ_A. If the inf-sup condition is satisfied, the element is stable and indeed optimal in its approximation properties. We denote the value of the inf-sup expression for a particular mesh as γ_h and thus have the condition that $\gamma_h \geq \gamma_A$ for all h.

Ideally, it would be possible to prove analytically that the condition in Eq. (2.41) holds when using the 3D-MITC8/s element in the analysis of general complex shell conditions. However, with such proof absent, we resort to using the inf-sup test proposed by D. Chapelle and K.J. Bathe [B, D], see Y. Ko and K.J. Bathe [B]. In this test, for a given shell problem, a mesh and its uniform refinements are considered and for each mesh case, the left-hand side of (2.41) is evaluated through an eigenvalue solution. The test has been applied abundantly to evaluate element formulations, see also K.J. Bathe [B]. For the details to perform an inf-sup test we refer to D. Chapelle and K.J. Bathe [C], K.J. Bathe [A, B], and K.J. Bathe, A. Iosilevich and D. Chapelle [A].

In the tests of the 3D-MITC8/s element, we are particularly interested in possible locking effects due to mesh distortions when the analysis of curved shells in bending-dominated conditions is considered. We have seen in the above convergence studies, see Sections 2.2.3 and 2.3.2, that shell elements may not give good results in such solutions. Hence, we focus the inf-sup tests on the cylindrical and hyperboloid shell structures when these are in bending-dominated conditions.

Cylindrical shell problem

We consider the cylindrical shell problem shown in Fig. 2.35. Utilizing symmetry, only one-eighth of the cylinder is modeled using $N \times N$ elements with $N = 4$, 8, 16, 32 and 64, and the shell thickness normalized by the overall dimension of the structure is $t / L = 1/10$, 1/100, 1/1000 and 1/10000. The element size is $h = L / N$. We consider the bending dominated case, that is, the boundary conditions are free at the edge AD, because then, if locking is present, it is most severe. In all analyses, using the three-dimensional elements, we use only one layer of elements in the shell thickness direction.

When using the regular mesh in Fig. 2.35, the mesh near the free end is graded in a boundary layer of width $0.5\sqrt{t}$, see D Chapelle and KJ Bathe [B]. We also perform the convergence studies with distorted meshes, a typical such mesh is shown in Fig. 2.35 (b).

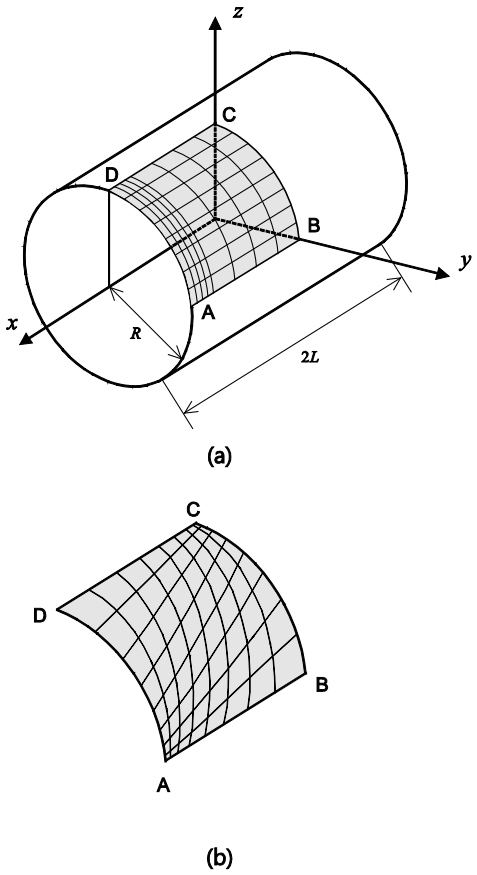


Fig. 2.35 Cylindrical shell for the inf-sup testing ($L = R = 1.0$, $t = 1/10$, $1/100$, $1/1000$ or $1/10000$, $E = 1.0$ and $v = 0.0$). (a) Problem description with the graded regular mesh (8×8 elements). (b) A distorted mesh used (8×8 elements). Regarding Figs.2.35 to 2.38 and more discussion, see Y. Ko and K.J. Bathe [B].

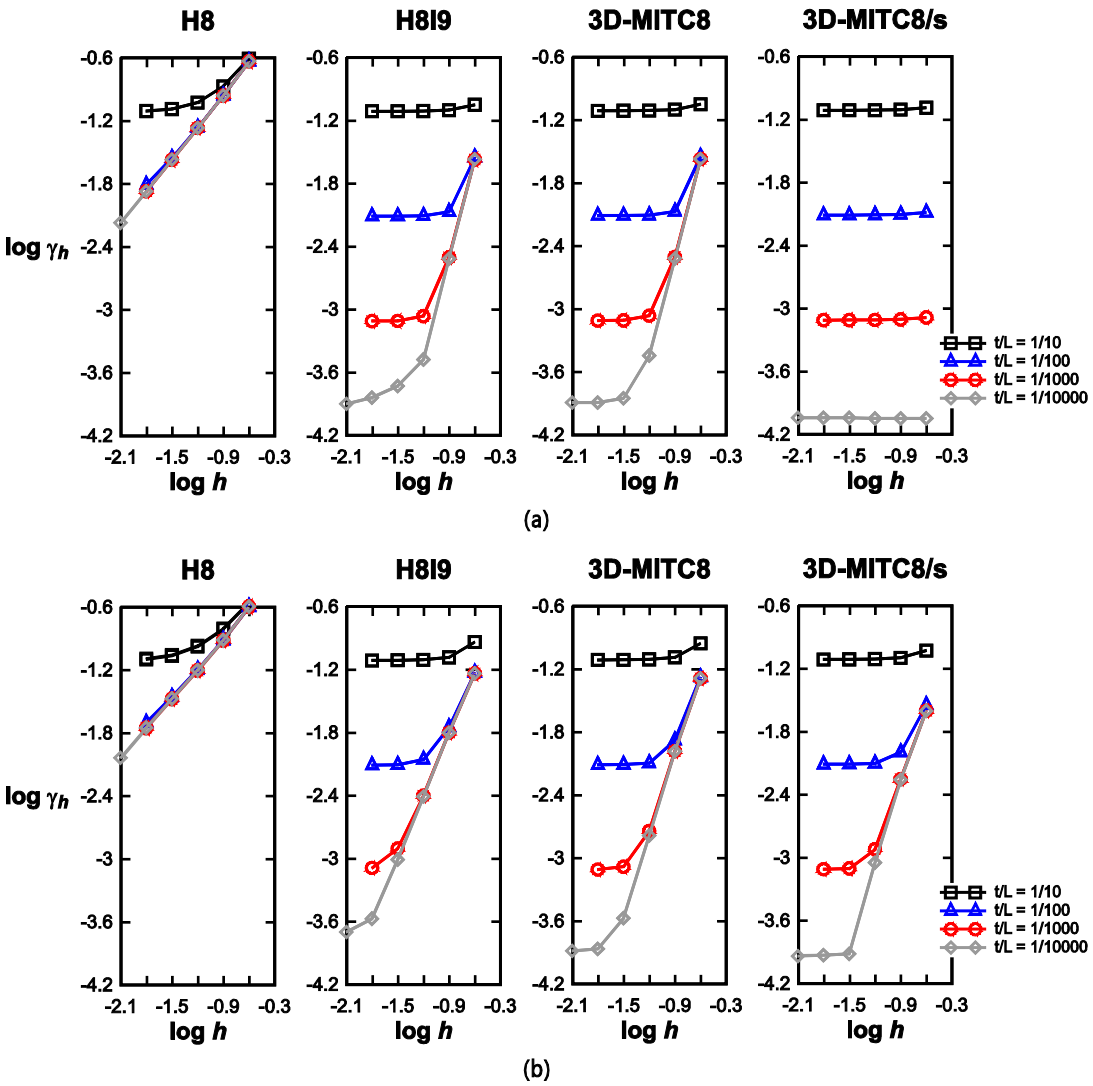


Fig. 2.36 Inf-sup convergence curves for the cylindrical shell using (a) regular and (b) distorted meshes.

Figure 2.36 shows the convergence of the inf-sup values. We denote the 8-node displacement-based element as H8, and the element when incompatible modes are included as H8I9. The inf-sup values of the 8-node displacement-based element continue to decrease, signifying, as expected, locking of the element. The H8I9 element performs better but also shows locking, in particular when the distorted meshes are used. The 3D-MITC8 element is also not effective when the very thin shell is considered. However, using the 3D-MITC8/s element, the inf-sup values are well bounded for the regular and the distorted meshes.

Since the only difference in the formulations of the 3D-MITC8 and 3D-MITC8/s elements is the use of the interpolation in Eq. (2.37) to prevent thickness locking, we can conclude that for the 3D-MITC8 element the thickness locking is causing the inf-sup value to continue to decrease when the shell is very thin.

Considering the results for the 3D-MITC8/s element, it is interesting to note that the stable inf-sup values decrease by an order of magnitude for each order of thickness decrease, see in particular Fig. 2.36 (a). This behavior corresponds to the decrease in the stiffness as the thickness decreases (for this bending-dominated problem).

Hyperboloidal shell problem

We next consider the doubly curved hyperboloidal shell problem shown in Fig. 2.37. The mid-surface of the shell structure is given by $x^2 + z^2 = 1 + y^2; \quad y \in [-1, \quad 1]$.

Using symmetry, only one-eighth of the structure is modeled with $N \times N$ elements and we use $N = 4$, 8, 16, 32 and 64. We consider $t/L = 1/10$, 1/100, 1/1000 and 1/10000 to see whether we have shear, membrane or thickness locking. The element size is $h = L/N$. To consider the bending-dominated case, the boundary conditions correspond to the free shell at the edge AB, with $u_z = 0$ along BC, $u_x = 0$ along AD, and $u_y = 0$ along DC. We again use only one layer of three-dimensional elements in the shell thickness direction.

When using the regular mesh shown in Fig. 2.37, the mesh near the free edge is graded in the boundary layer of width $0.5\sqrt{t}$. We also perform the inf-sup test with distorted meshes, see Fig. 2.37.

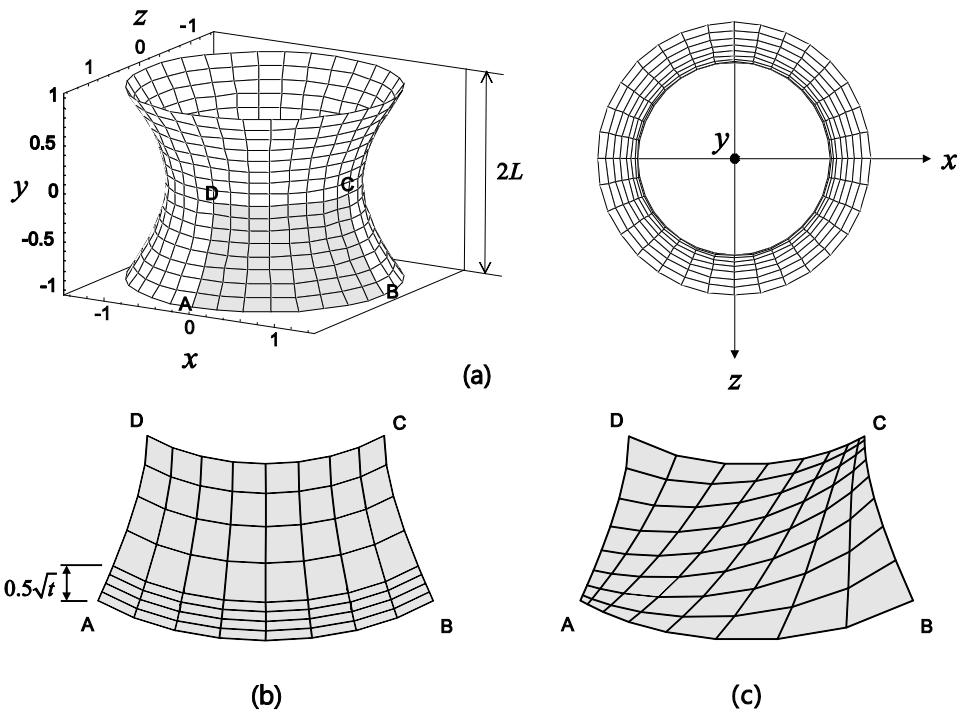


Fig. 2.37 Hyperboloid shell for inf-sup test ($L=1.0$, $t=1/10$, $1/100$, $1/1000$ or $1/10000$, $E=1.0$ and $v=0.0$). (a) The shell (b) Graded regular mesh (8×8 elements) (c) A distorted mesh used (8×8 elements).

Figure 2.38 shows the convergence of the inf-sup values. The behavior of the inf-sup values using the H8I9 element for the very thin shell implies locking. Considering the results using the 3D-MITC8 element, some locking is also present when considering the very thin shell, whereas better results are obtained using the 3D-MITC8/s element.

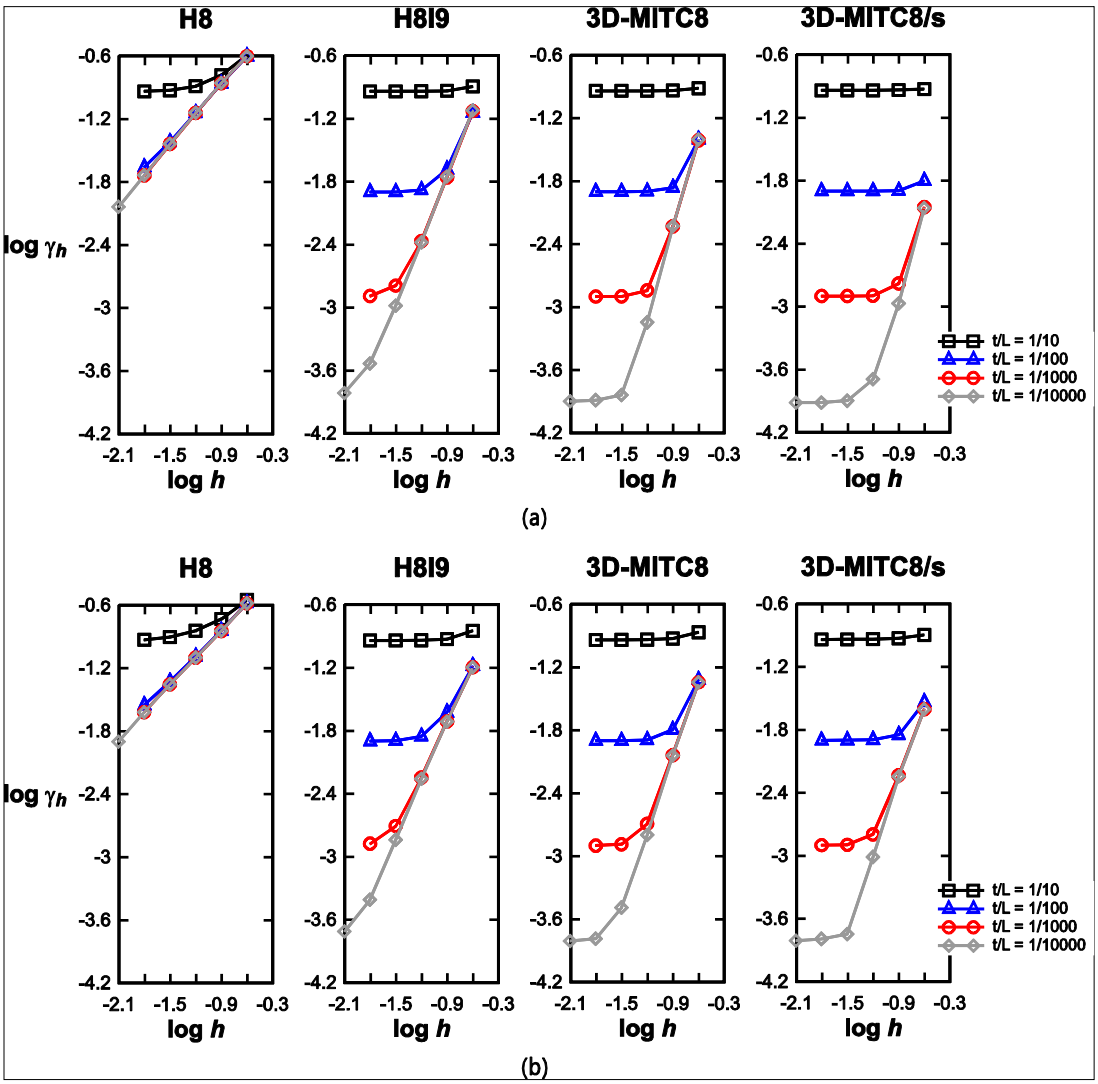


Fig. 2.38 Inf-sup convergence curves for the hyperboloid shell using (a) regular and (b) distorted meshes.

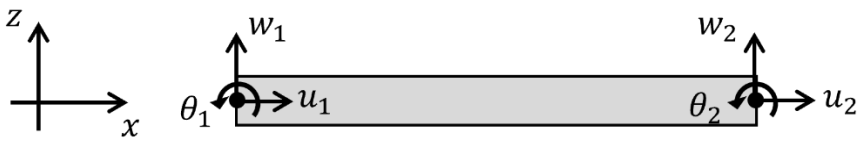
2.5 Exercises

We suggest these Exercises to obtain further insight into the solution schemes discussed in this chapter. To complete an Exercise, the material given in KJ Bathe [A] may need to be known.

2.5.1

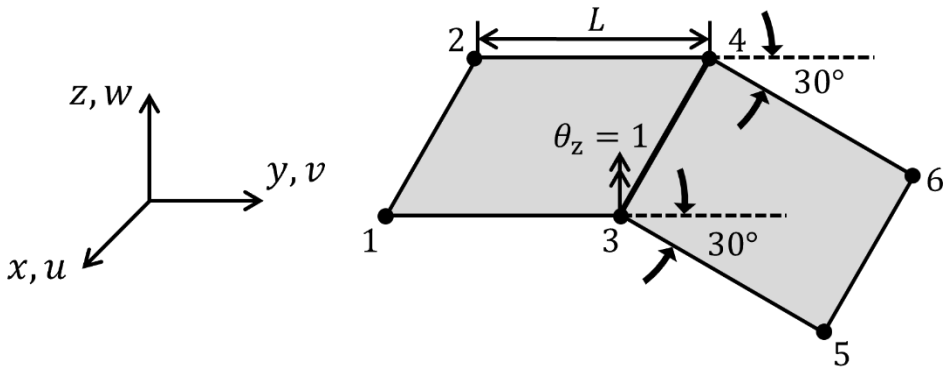
Consider a 2-node beam element acting in the x-z plane with three degrees of freedom at each node.

Show that the element when based on the Bernoulli beam theory does not lock, but that the element locks when based on the Timoshenko beam theory.



2.5.2

Consider the two identical flat MITC4 shell elements below fully constrained at all nodes except free to displace and rotate at node 3. The elements are formulated using five degrees of freedom at each node, and are connected along line 3 - 4 at an angle of 30 degrees. Identify whether an incompatibility in displacement occurs when $\theta_z = 1.0$ is applied at node 3.



2.5.3

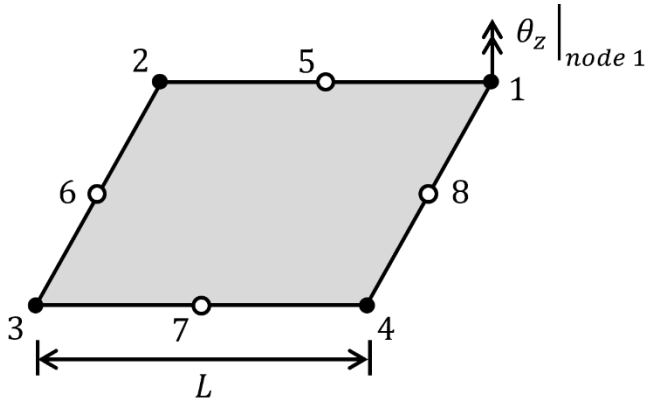
Show explicitly that the use of Eqs. (2.8) for the MITC4 flat rectangular shell element prevents shear locking of the element.

2.5.4

Prove that if the MITC4+ shell element geometry is flat, the membrane strains are those of the displacement-based plane stress element. Consider a square element.

2.5.5

Consider the 4-node square shell element shown below. Formulate how to include the drilling rotation at node 1 in the formulation. Use the fictious nodes 5 and 8 at the mid-side points shown and the linear / parabolic membrane displacement assumption as explained in the text.



2.5.6

Show that the boundary conditions used in the analysis using symmetry of the hyperboloid shell problem are correct for the analysis case considered.

Chapter 3 The Overlapping Finite Elements

The effectiveness of a finite element analysis largely depends on the finite element discretization used. Hence various finite element formulations have been proposed recently in addition to those given and referred to in K.J. Bathe [A], in particular, the generalized finite element method, see C.A. Duarte, I. Babuška, and J.T. Oden [A] and the variants thereof, finite elements with covers, see for example J.. Kim and K.J. Bathe [A], and finite element methods involving smoothing, see for example G.R. Liu, T. Nguyen-Thoi and K.Y. Lam [A]. In all these developments and those presented in this chapter, the reliability and efficiency of a proposed method is of great importance for engineering use, which requires a focus on mathematical analyses *and – very importantly -- details to identify whether the method is reliable, efficient and easy to use in engineering practice*.

In this chapter we focus on a new approach of analysis, namely the use of "overlapping finite elements" which was also only recently proposed, see K.J. Bathe [C]. We briefly mention as well overlapping finite element meshes, see J. Huang and K.J. Bathe [B], which would be used with traditional finite elements or overlapping elements. As presented below, the overlapping finite elements are a further development of the Method of Finite Spheres, a meshless method, see S. De and K.J. Bathe [A, B] and J.W. Hong and

K.J. Bathe [A]. We show that the use of overlapping finite elements can lead to very effective solutions, an observation which indicates a significant potential for their further development and their use in engineering and the sciences.

In the discussion below, we focus on the analysis of solids, but the concepts can of course also be used in the solution of fluid flows, see W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R.C. Mesquita [D], because the incompressibility of the fluid is modeled like in the analysis of incompressible solids.

3.1 The problem of traditional elements "not to overlap"

Despite the well-known success of the finite element method in the analysis of solids and structures, there is still the burden that – using conventional elements and more recently proposed discretizations – an analyst needs to establish a "good quality" mesh for accurate solution results. The main point is that *the elements used in a mesh must not overlap and not be too distorted*, that is, in a two-dimensional analysis, quadrilaterals should be close to squares or rectangles and triangular elements should be close to equilateral triangles, in particular all elements should not be distorted to be thin sliver elements. In addition, for higher-order elements, the internal nodes should be placed at their natural positions, see N.S. Lee and K.J. Bathe [A] and K.J. Bathe [A].

Exceptions are given when special stress conditions, like those encountered in fracture mechanics, are analyzed. For example, using an 8-node element for simulations of two-dimensional fracture, an element side may be collapsed, and the adjacent mid-side nodes be moved to the quarter-points to model a stress singularity, see K.J. Bathe [A].

Powerful meshing algorithms have been developed to achieve "good quality" meshes with largely undistorted elements by, for example, aiming that each element internal angle satisfies criteria to not have a distorted element shape. However, in solutions when very complex geometries need to be meshed, these

meshing schemes will not give undistorted elements throughout the geometry. Frequently, quite distorted elements are still encountered in certain areas of the meshed domain.

The important point is that traditional elements when geometrically distorted lose for most analyses their predictive capability, see K.J. Bathe [A]. That is, element distortions in general deteriorate the analysis results. Indeed, the solution accuracy may decrease drastically due to element geometric distortions. The decrease in accuracy depends of course on various factors, like the type of elements used, their actual distortions, where in the domain of analysis the elements are located and how large a domain is covered by distorted elements. To improve the analysis predictions, *error estimation schemes with remeshing* and *stress improvement direct solution schemes* have been proposed. The error estimation schemes, see e.g. O.C. Zienkiewicz and J.Z. Zhu [A] and T. Grätsch and K.J. Bathe [A, B], estimate the error of an obtained finite element solution and the magnitude of error can then be used with remeshing schemes, possibly in iterations, to establish better solutions, e.g. R. Boussetta, T. Coupez and L. Fourment, L [A]. A direct stress improvement scheme can without iterations significantly (and directly) improve stress predictions, see e.g. D.J. Payen and K.J. Bathe [A], but of course could also be used with remeshing and mesh iterations to further improve solutions.

The direct stress improvement scheme in essence uses the solution obtained with the given mesh and projects that solution onto an “enriched space” to reach an improvement in the stress prediction. The enriched space used depends on the elements employed and whether these are internal or near the boundary of the geometric domain. The scheme can improve a stress prediction significantly, see D.J. Payen and K.J. Bathe [A] and K.J. Bathe [F], but the results depend of course inherently on the stresses used as an input to the algorithm. Hence the final predicted stresses are still affected by the element distortions in the mesh for the original solution, although the effect may not be large. As already mentioned above, if the solution error is still deemed too large, remeshing schemes could be used in conjunction with the direct stress improvement scheme.

A related approach is to smooth analysis results within elements and over element sides. The scheme of smoothing is frequently performed within and across defined cells involving parts of the finite elements

and mesh. There are many variations of smoothing, with the objective to establish improved strain fields within lower order elements and avoid undue stress jumps over element boundaries. The solutions may correspond to a "softer" behavior than a traditional displacement-based solution of the continuum, which can be of benefit, but may also contain undesirable spurious non-zero energy modes. see e.g. G.R. Liu, T. Nguyen-Thoi and K.Y. Lam [A], C. Lee and P.S. Lee [A], and the references therein.

In principle, of course all the above schemes to improve stress predictions can be combined, and in various ways, but the resulting procedure may be complicated and not be sufficiently robust.

However, these approaches -- the error estimation schemes with remeshing, the direct stress improvement schemes and strain smoothing techniques -- can encounter difficulties in the material nonlinear analyses of solids. Consider the analysis of a reinforced concrete structure with highly nonlinear behavior due to the nonlinear stress-strain relationship including cracking (in tension), stress softening, and crushing (in compression), and the interaction with reinforcing bars. The smoothing of calculated stresses may not result in better stress predictions and to establish a more effective mesh requires a mathematically firm and helpful error measure. Also, a developed procedure may computationally be very costly to use.

Because of the difficulties encountered in reaching an effective mesh and in performing remeshing, meshless discretization methods have been developed, notably the element-free Galerkin method, see T. Belytschko, Y.Y. Lu and L. Gu [A] and also the method of finite spheres, see S. De and K.J. Bathe [A, B], J.W. Hong and K.J. Bathe [A], and W. L. Nicomedes, K. J. Bathe, F. J. S. Moreira and R.C. Mesquita [A, B, C]. In these developments, for the practical use of a meshless method, it is best that the method is not based on using artificial numerical factors, like it is the case for the method of finite spheres, because the reliability and robustness of a scheme are very important in numerical simulations, see K.J. Bathe [A].

If we consider the use of the method of finite spheres, we see that the procedure will cover the complete analysis domain with *overlapping* spheres (disks in two-dimensional analyses) and because of the inherent overlapping of the spheres, there are no thin or highly distorted elements as can occur in the traditional

finite element meshing. However, while elegant in the formulation, a major difficulty in the use of the method of finite spheres is the computational expense of the numerical integration – typically encountered with meshless methods not using artificial numerical factors. This expense is relatively high for use in engineering practice, see for example B. Lai and K.J. Bathe [A].

Instead of using a meshless method, another approach is to develop "finite elements" that overlap and are "distortion-insensitive", see K.J. Bathe [C] and K.J. Bathe and L. Zhang [A]. As we discuss below, the overlapping finite elements are in various regards a further development of the method of finite spheres. The approach is mathematically well-founded and effective and can be used in linear and general nonlinear analyses, including statics and dynamics. Specifically, the strong mathematical foundation of the overlapping finite elements and their insensitivity to mesh and element distortions render their use very attractive -- the meshing can be performed more efficiently and better accuracy in the solution results can be expected.

Furthermore, due to the distortion-insensitivity, the elements may also be very useful for the prediction of large deformations. In such analyses, the solution domain can deform significantly and hence the elements discretizing the original domain may undergo large distortions which may require remeshing when using traditional finite elements. The use of overlapping finite elements may significantly decrease the need for remeshing.

3.2 Overlapping finite element formulation

In the following we first present the basic considerations to formulate overlapping finite elements (FN1) and then we illustrate the use of the elements in analyses.

(FN1) It is interesting to note that the term "overlapping finite element" was earlier used in W.J.T. Daniel [A] but in a quite different context, namely, to establish high-continuity (HC) elements, see also M. Aristodemo [A].

3.2.1 Fundamental considerations and formulation

As we mentioned above, the overlapping finite elements are a further development of the method of finite spheres. The objective is to keep the excellent predictive properties of the method of finite spheres but render the new procedure computationally effective. This was achieved in several research steps.

In the first achievement, a triangulation underlying the disks (considering two-dimensional analysis) was identified and used. Each 3-node triangle in the triangulation is seen to be a low-order overlapping element. The displacement interpolation is using the rational functions of the method of finite spheres, but the interpolation is valid only over the domain of the overlapping element. Hence the numerical integration of the element stiffness and mass matrices is only performed over the overlapping element, see K.J. Bathe and L. Zhang [A], and L. Zhang and K.J. Bathe [A].

In a subsequent achievement, the rational functions used to express the displacements are interpolated over the overlapping element using the one order higher traditional interpolation functions with virtual nodes for the higher-order functions. The virtual nodes do not carry degrees of freedom and are only used for the interpolation of the rational functions. This step reduces the effort of numerical integration because only polynomial functions are used in the basic displacement interpolations. Also, the bandwidth of the stiffness and mass matrices is reduced because no node outside the overlapping element is involved in the displacement interpolation over the overlapping element. The interpolation uses the values of the rational functions at the nodes (including the virtual nodes) of the overlapping element, see Eqs. (3.1a,b,c) below and L. Zhang, KT Kim and KJ Bathe [A], KT Kim, L Zhang and KJ Bathe [A] and J Huang and KJ Bathe [A].

Then, finally, the evaluations of the rational functions at the nodes are no longer needed, and instead, the displacement interpolation functions are directly established. This step further simplifies the analysis

procedure, see S. Lee and KJ Bathe [A].

An important capability of the presented overlapping finite elements is that we can directly use these elements also with harmonic functions to solve wave propagation problems, see K.T. Kim, L. Zhang and K.J. Bathe [A], Y. Chai and K.J. Bathe [A] and K.T. Kim and K.J. Bathe [B] and Section 3.3.4, as with the method of finite spheres, S. Ham, B. Lai and K.J. Bathe [B].

A powerful use of the overlapping finite elements is found in the AMORE analysis procedure, with AMORE being an acronym for Automatic Meshing with Overlapping and Regular Elements, see K.J. Bathe [D, N]. In the solution procedure mostly undistorted (regular) traditional low-order finite elements are used and these are coupled to overlapping finite elements discretizing the domain where these elements are particularly useful, see Section 3.6. The coupling is performed similarly to how traditional finite elements can be coupled to disks or spheres of the method of finite spheres, see J.W. Hong and K.J Bathe [A], K.J. Bathe and L. Zhang [A] and J. Huang and K.J. Bathe [A]. The details of the formulation of coupling elements are given in Section 3.2.3

In the presentation below, we first discuss the developments that lead to obtain valuable insight in the final formulation of the overlapping finite elements and then focus on the current use of the overlapping elements, their formulations and illustrative solutions.

Let us consider a general triangularization, using triangular cells, of a two-dimensional geometry. Figure 3.1 shows a patch of a triangularization and how three polygonal elements with center nodes 1, 2, and 3 are formed by triangles. In each case all triangles attached to the center node *m* make up the corresponding polygonal element *m*. In this way we obtain polygonal elements 1 and 3 with 5 sides and polygonal element 2 with 6 sides. The overlapped region of the three polygonal elements is the grey triangular

domain, which we define to be an *overlapping 3-node finite element*. We note that with this approach, each of the triangular cells in Fig. 3.1 becomes a triangular 3-node overlapping finite element.

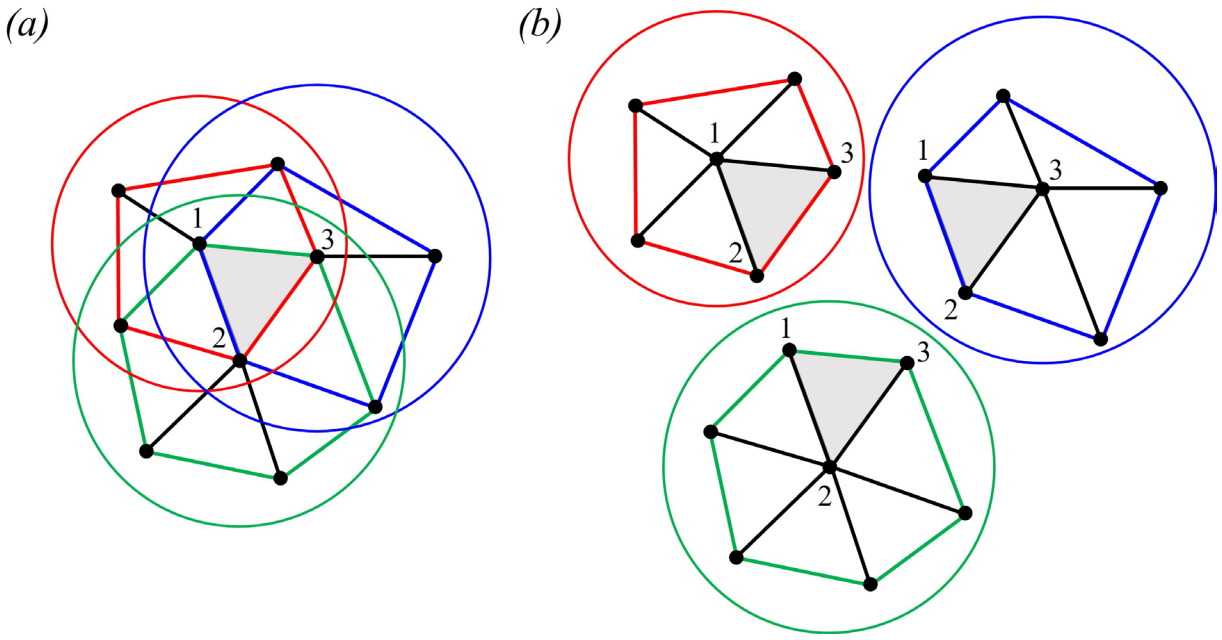


Fig. 3.1 (a) A triangular overlapping finite element with nodes 1, 2, and 3; (b) the overlapping finite element corresponds to the overlapped region of the three polygonal elements with center nodes 1, 2, and 3; the circles indicate the corresponding finite disks for the overlapping finite element. Regarding Figs. 3.1 to 3.3 and more discussion, see S. Lee and K.J. Bathe [A].

Hence, the name “overlapping elements” relates to the observation that disks overlap in the method of finite spheres (in three-dimensional analysis, spheres instead of disks), and here polygonal elements within the disks overlap from which we extract the “overlapping 3-node element”.

The above discussion already shows that a mesh of overlapping elements can be obtained using the usual mesh generation schemes of traditional finite elements. This is an important point for the practical use of overlapping elements and holds for all overlapping elements here presented. This point will become even more apparent below.

Assume that we are seeking to calculate a scalar field *u* (like a displacement or temperature) over a geometric domain. In the overlapping finite element method, the field is interpolated using the local fields for the overlapping finite elements. For a 3-node element, as shown in Fig.3.1, we have

$$u = \sum_{I=1,2,3} h_I \, \psi_I \qquad (3.1a)$$

where h_I is the low-order finite element function for node *I* of the overlapping element, and ψ_I is the local field associated with node *I*. Using the concepts of the method of finite spheres, we use

$$\psi_I = \sum_{K=1,2,3} \phi_K^I \, u_K \qquad (3.1b)$$

where ϕ_K^I is the rational function of the method of finite spheres. To satisfy the rigid body mode criterion we need that for each *I* in Eq. (3.1b) $\sum_{K=1,2,3} \phi_K^I = 1.0$.

In principle, different local fields can be employed but in the development of the overlapping elements the focus was on using those employed in the method of finite spheres because of the good solution accuracy obtained and the observed distortion insensitivity. However, whereas in the method of finite spheres, all nodes within the disk of node *I* are included in the definition of the displacement field of the disk, the local field of the 3-node overlapping element is here only constructed from the three nodes of the element, see Eqs. (3.1a) and (3.1b). This leads to a much-reduced bandwidth of the stiffness and

mass matrices.

As already pointed out above, a disadvantage of these local fields is still that rational functions are used, but a polynomial interpolation of the rational functions, local over the overlapping element considered, can be designed. For the interpolation of the local field over the 3-node overlapping elements considered above, we use

$$\phi_K^I = \sum_{i=1}^{6} \hat{h}_i \hat{\phi}_{Ki}^I \qquad (3.1c)$$

where the $\hat{h}_i$ are the interpolation functions of the higher-order 6-node element corresponding to the 3-node element. The mid-side nodes of the 6-node element are called "virtual nodes" because these nodes carry no degrees of freedom and are only used for the interpolation of rational functions.

The $\hat{\phi}_{Ki}^I$ in Eq. (3.1c) are the values of the rational function at the six element nodes. Here too, the condition $\sum_{K=1,2,3} \phi_K^I = 1.0$ is satisfied for each node *I*. The details are given in L. Zhang, K.T. Kim, and K.J. Bathe [A], K.T. Kim, L. Zhang and K.J. Bathe [A], J. Huang and K.J. Bathe [A] and K.J. Bathe [D]. Using this interpolation, the numerical evaluation of the overlapping finite element matrices is significantly less costly.

These two solution characteristics -- the much smaller bandwidth and the less costly numerical integration -- render the overlapping element much more effective than the method of finite spheres.

The overlapping elements satisfy of course the usual convergence requirements, namely compatibility, the rigid body mode and patch tests, and the convergence of these elements can be analyzed, see J. Huang and K.J. Bathe [C].

For the polygonal elements in Figure 3.1, we note that all overlapping elements are 3-node elements. Similarly, we also have 4-node quadrilateral overlapping elements in two-dimensional analysis and the corresponding three-dimensional elements. Furthermore, these elements can also be used in overlapping finite element meshes, see J. Huang and K.J. Bathe [B, C].

Illustrative solutions using these overlapping finite elements are given in the above references. While these elements are quite effective regarding the numerical integration of the element matrices and are (almost) distortion-insensitive, simpler but closely related interpolation functions are proposed in S. Lee and K.J. Bathe [A]

$$\rho_K = h_K + \beta \sum_J (h_J - h_K)\hat{h}_{JK} \qquad (3.2)$$

with

$$u = \sum_K \rho_K u_K \qquad (3.3)$$

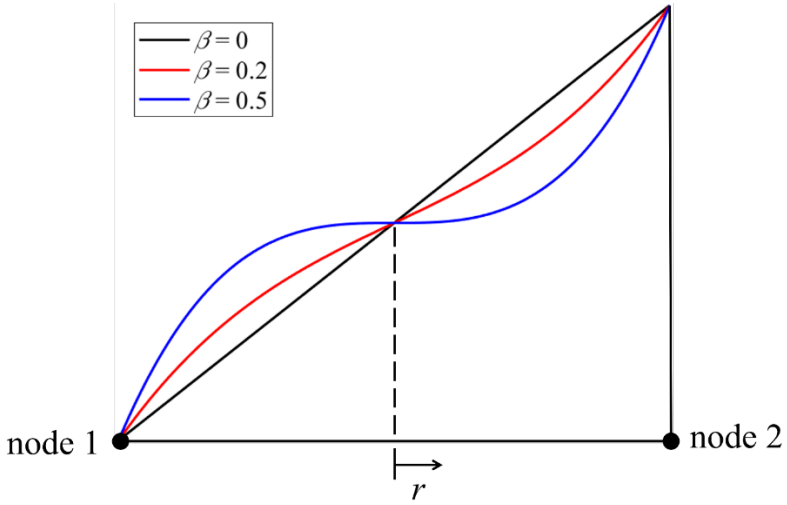
In Eq. 3.2, the h_J and h_K are the low-order traditional interpolation functions of the 3-node and 4-node elements in two-dimensional analysis, for nodes *J* and *K* (and the corresponding elements in three-dimensional analysis), *J* varies over all the nodes directly connected by an element edge to the node *K,* $\hat{h}_{JK}$ is the higher-order parabolic interpolation function corresponding to the mid-point of that element edge, and β arises because of the interpolation of the rational functions in the formulation. It is important that values of β could be established using a specific rational function and that different rational functions could be employed, for details see S. Lee and K.J. Bathe [A]. In the method of finite spheres, the rational functions vary depending on the radius of the sphere. The use of one value for β for a complete element assemblage is more effective because it is a constant value for all overlapping

elements, and it is directly chosen to have good conditioning of the assembled stiffness matrix, see Section 3.3.1.

The overall aim in these developments was to retain the valuable properties of the method of finite spheres but obtain a much more effective solution procedure. The expression in Eq. (3.2) shows that only polynomial functions are used to establish the interpolations ρ_K and only the traditional element nodes are used to define the element interpolations – just like in a traditional finite element analysis. As already mentioned, the overlapping elements can therefore directly be implemented in a traditional finite element program, the numerical integrations of the element matrices are more effective and the bandwidth of the resulting matrices is smaller than when using the method of finite spheres.

In Eq. (3.3) the u_K are the nodal degrees of freedom defined below in Eq. (3.4). The interpolation functions ρ_K are used for the displacements while in linear analysis, for the interpolation of the geometry, the traditional functions h_K are employed. The illustrative solutions given below have been obtained with these interpolations.

To illustrate the functions used in Eqs. (3.2) and (3.3), Fig. 3.2 shows the simple case of a 2-node one-dimensional element in its natural coordinate system. The interpolation functions h_K, K =1, 2 are those of the isoparametric 2-node element, and the interpolation function $\hat{h}_{JK}$, giving $\hat{h}_{21}$= $\hat{h}_{12}$, is the parabolic function with end-nodes 1 and 2, as shown in the figure.



$$\rho_2 = h_2 + \beta(h_1 - h_2)\,\hat{h}_{12}$$
$$= \frac{1}{2}(1 + r) - \beta r(1 - r^2)$$

Figure 3.2 Interpolation function for node 2 of overlapping 2-node element

Figure 3.3 shows an interpolation function of the 3-node overlapping finite element. In Figures 3.2 and 3.3 , ρ_K is shown for rather large values of β merely to illustrate the function.

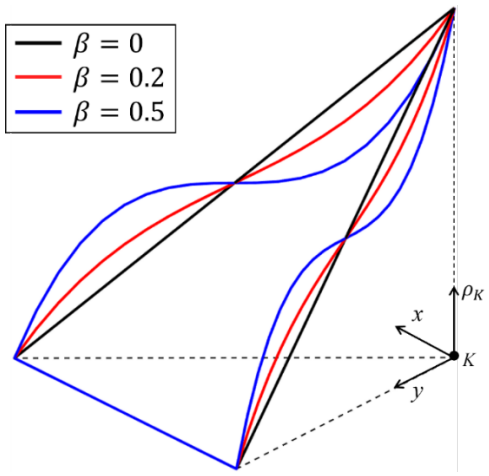


Figure 3.3 Interpolation function ρ_K for a 3-node overlapping finite element

In the use of these interpolations, the numerical integration of the element matrices is performed to a sufficiently high order of accuracy to not have any element spurious zero energy mode. This is important for the reliability of the solution scheme (like in all finite element methods), see K.J. Bathe [A]. The numerical integration is addressed in Section 3.2.2.

The final consideration pertains to the degrees of freedom used in u_K. If the nodal degrees of freedom are just those employed in the traditional finite element analysis, the elements are not powerful and we need to include at least the linear basis as nodal degrees of freedom, also to pass the patch test. Hence, we use additional degrees of freedom. Considering only one continuous variable to be solved for, here *u* (x, y, z) where x, y, and z are the Cartesian coordinates for the geometry under consideration, we use at each node *K*

$$u_K = \alpha_{K1} + \alpha_{K2}\hat{x} + \alpha_{K3}\hat{y} + \alpha_{K4}\hat{z} + \alpha_{K5}\hat{x}\hat{y} + \ldots \qquad (3.4)$$

where the $(\hat{x}, \hat{y}, \hat{z})$ are the nodal Cartesian coordinates measured from the node *K* and the α_{Ki}, *i* = 1, 2, 3, 4 … are nodal unknowns to be solved for. The coordinate values $(\hat{x}, \hat{y}, \hat{z})$ may in practice be normalized by some representative length to allow for different sizes of elements.

In Eq. (3.4), α_1 corresponds to the traditional degree of freedom. Of course, the more degrees of freedom are used at the nodes, the more powerful will be the element, in general, but also the computational cost per element and the bandwidth of the stiffness and mass matrices increase.

Based on the discussion given above, we can now summarize the important characteristics of an overlapping finite element and add further observations --

- While we introduce higher-order functions, there are no degrees of freedom at the mid-points of the element edges. The only degrees of freedom are those at the element corner nodes (as in traditional finite element analysis).
- The interpolation functions pertain only to the overlapping element considered and are compatible over the element edges with the functions of the adjacent overlapping elements. The coupling of the overlapping elements to traditional elements is discussed in Section 3.2.3.
- The sum of the interpolation functions is equal to 1. That is, for all elements

$$\sum_K \rho_K = 1 \quad (K \text{ summing over all nodes of the element}) \qquad (3.5)$$

- Equations (3.3) and (3.5) imply that the rigid body modes of translations and rotations and the constant strain states can be represented by the element when both the displacements and the geometry are interpolated using the ρ_K interpolation functions. The proof is given in Section 5.3.3 of K.J. Bathe [A] by simply using in that section the ρ_K for the interpolations instead of the h_K. Since, as mentioned already above, we use at least the linear basis as nodal degrees of freedom and the element is compatible, also the patch test (involving the different stress conditions) is passed. The same also holds true when in linear analysis, the ρ_K functions are used for the interpolation of the displacements and the h_K functions are used for the interpolation of the geometry.

- Since the fundamental ingredients of the overlapping finite elements and the traditional finite elements are the same, the solution scheme based on the use of overlapping finite elements is

stable as in the use of traditional finite elements. However, for the analyses of incompressible media appropriate pairs of element spaces need to be chosen (like in the traditional u/p formulation of finite elements, see K.J. Bathe [A]), see Section 3.4.

- The degrees of freedom used in Eq. (3.4), in fact, correspond to various degrees of approximations desired. Hence an element performance may simply be increased by using a higher-order polynomial for the degrees of freedom, and the order used may even change from node to node over each element in the mesh. Since this process is performed using the given mesh, considerable flexibility is present to increase the accuracy of a solution without any remeshing (as may be needed with meshes of traditional elements). This is an additional strength of the overlapping elements to the possible use of special functions for the solution of specific problems (like functions to allow for warping, see K.J. Bathe and A. Chaudhary [A], ovalization displacements of pipe cross-sections, see K..J Bathe and C. Almeida [A, B, C], harmonic functions for wave propagations, see e.g. S. Ham and K.J. Bathe [A] and K.T. Kim and K.J. Bathe [A], or functions to model high stress gradients, see e.g. T.P. Fries and T. Belytschko [A]).

Usually, the nodal degrees of freedom are polynomials. Then considering the convergence of the solution scheme, the formulation achieves the 2p-th order convergence of strain energy error (assuming that the problem solved has a sufficiently smooth solution)

$$E - E_h \leq c\, h^{2p} \tag{3.6}$$

where E and E_h are the strain energies of the exact and numerical solutions, respectively, h is the element size, and c is a constant independent of h but dependent on the material properties,

see J. Huang and K.J. Bathe [C],

In addition to the flexibility inherent in the possible use of different nodal degrees of freedom, the overlapping finite elements can also be employed together with traditional finite elements. This flexibility is effective in analyses taking advantage of using non-distorted traditional elements when possible and overlapping elements when distorted elements should naturally be used. We exploit this approach in the AMORE scheme, see Section 3.6.

The strengths of the overlapping elements pointed out above -- the *strong mathematical foundation*, *distortion insensitivity*, *inherent flexibility in their use due to suitable nodal degrees of freedom, use with traditional meshing schemes*, *use with error estimates to improve solutions without remeshing* -- and the relative ease of implementation render the overlapping finite elements very attractive for use in engineering practice, in static and dynamic analyses. Although not considered here, the overlapping finite elements are also very effective in general nonlinear analyses.

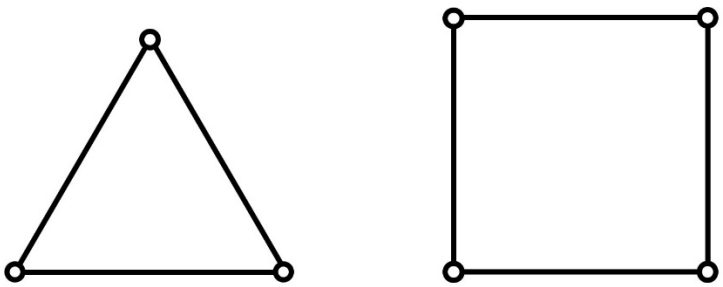
We next consider the available overlapping elements and illustrate their use. More illustrations are given in the references given above.

3.2.2 Available elements for two- and three-dimensional analyses

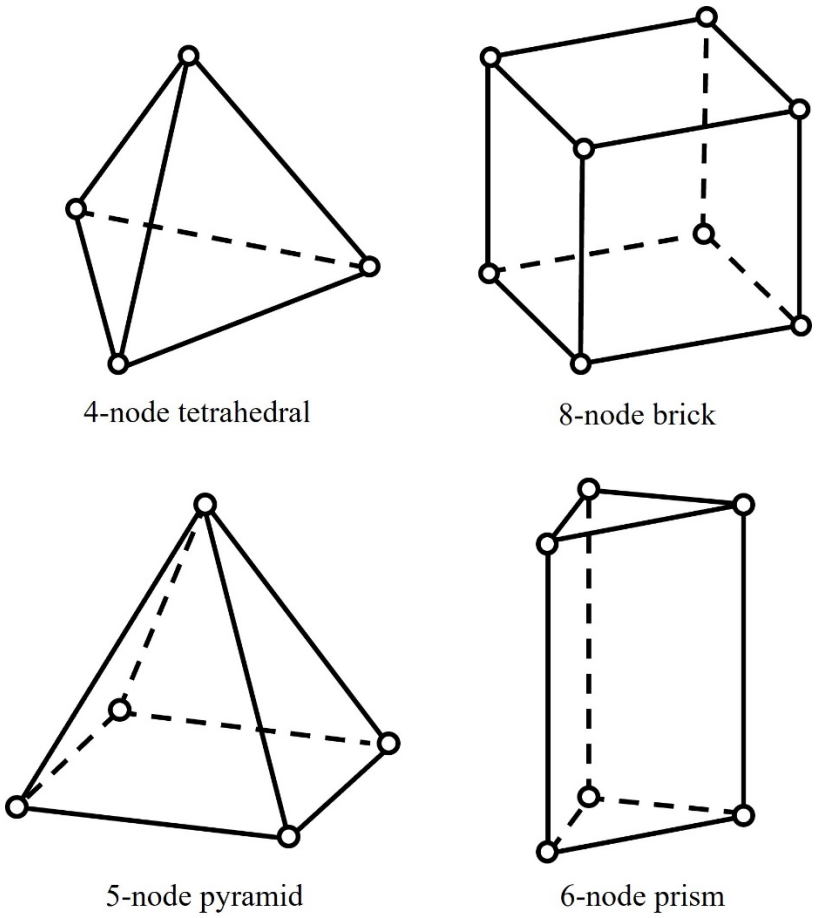
The elements used as overlapping elements are the traditional low-order elements, but the elements carry of course in general additional degrees of freedom at the nodes. Fig. 3.4 summarizes the elements and shows the nodes of the elements that carry the (physical) degrees of freedom.

These element matrices are evaluated using numerical integration. The order of the integration needs to be in each case high enough to not have any spurious zero energy mode in the element. Table 3.1 lists the integration orders that can be used when the nodal degrees of freedom correspond to a linear polynomial.

While the integration order is higher than usually employed for the low-order traditional elements, the computational cost to calculate the element matrices is still small when compared to the total cost to solve for a solution of an element assemblage. Of course, a higher integration order needs to be employed if the degrees of freedom include higher-order functions, see S. Lee and KJ Bathe [A, B].



(a) Elements for two-dimensional analyses



(b) Elements for three-dimensional analyses

Figure 3.4 Overlapping elements used in two- and three-dimensional analyses of solids. Regarding Figs. 3.4 to 3.11 and further information, see S. Lee and K.J. Bathe [B].

Table 3.1 Integration orders used for some overlapping elements, assuming a linear polynomial basis for the nodal degrees of freedom

Element	Numerical integration rule
3-node triangular	6-point rule by G.R. Cowper [A]
4-node quadrilateral	3 × 3 Gauss quadrature
4-node tetrahedral	11-point rule by P. Keast [A]

8-node brick	3 × 3 × 3 Gauss quadrature	
5-node pyramid (see Fig. 3.5)	*r*- and *s*-directions: 2 × 2 3 × 3 Gauss quadrature 3 × 3	*t*-direction: 3-point Gauss quadrature
6-node prism (see Fig. 3.6)	*r*- and *s*-directions: 6-point rule by G.R. Cowper [A]	*t*-direction: 3-point Gauss quadrature

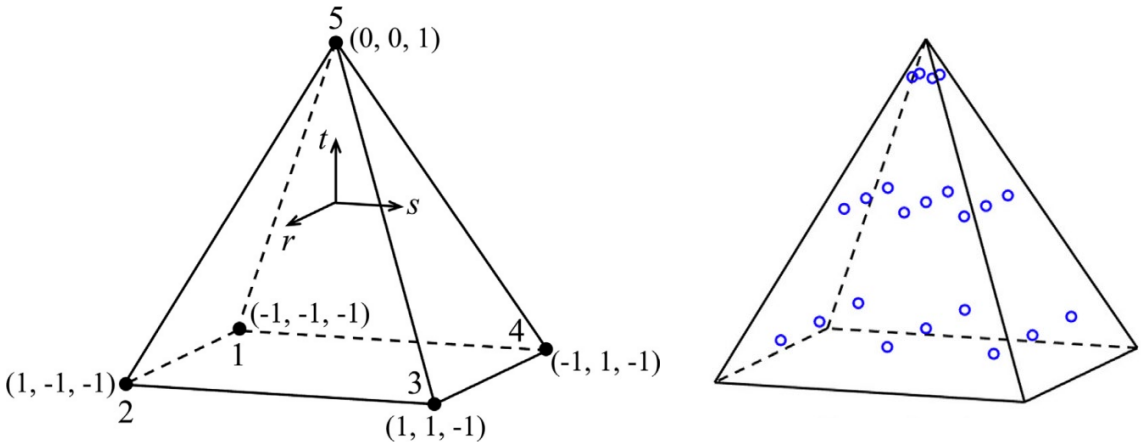


Figure 3.5 Reference element for the geometry interpolation of the pyramid overlapping element (left) and positions of the numerical integration points (right)

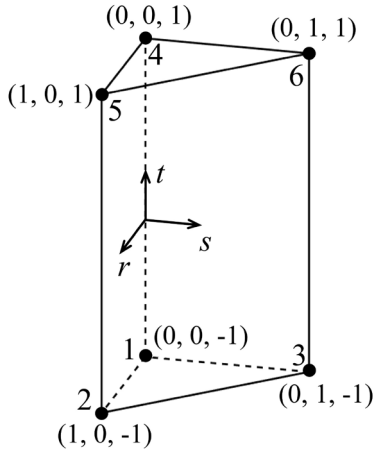


Figure 3.6 Reference element for the geometry interpolation of the prism overlapping element.

3.2.3 Coupling elements

As mentioned already, the overlapping elements (with overlapping element nodes) can be used in conjunction with traditional elements (with traditional nodes). However, since the overlapping elements use different interpolation functions from the functions used for traditional finite elements and carry additional nodal degrees of freedom, we need coupling elements, that is, elements which couple the traditional and overlapping elements. Figure 3.7 gives an illustration of a mesh with coupling elements. We see that an element containing at least one overlapping node and at least one traditional node is a coupling element.

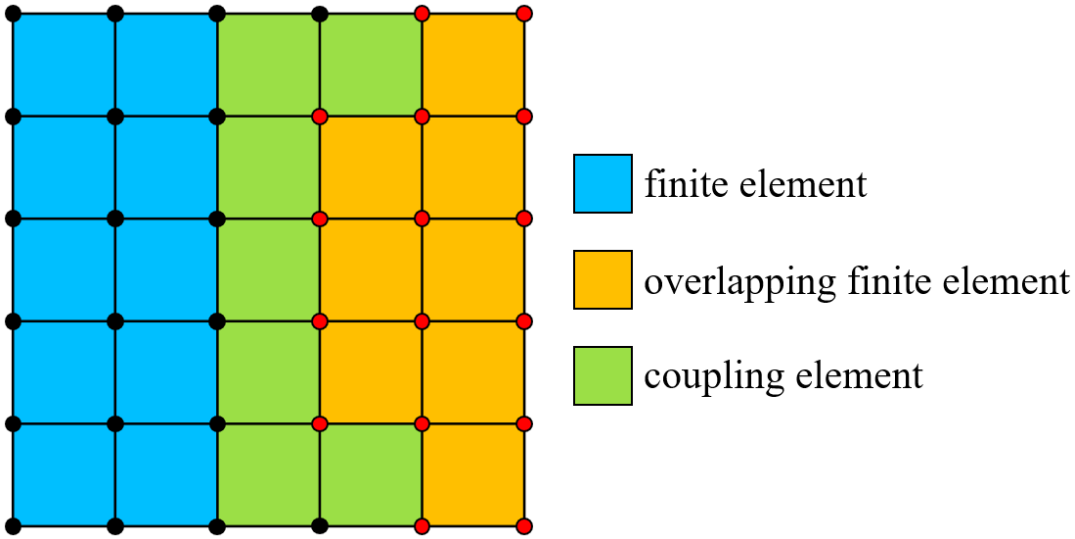


Figure 3.7 Coupling elements for transitioning from traditional to overlapping elements

In addition to the transition between traditional and overlapping elements as considered in Fig. 3.7, the coupling elements are also useful when we need to impose boundary conditions. Then we can use traditional element nodes on the boundary and overlapping element nodes to couple into overlapping elements, see Fig. 3.8 for an example.

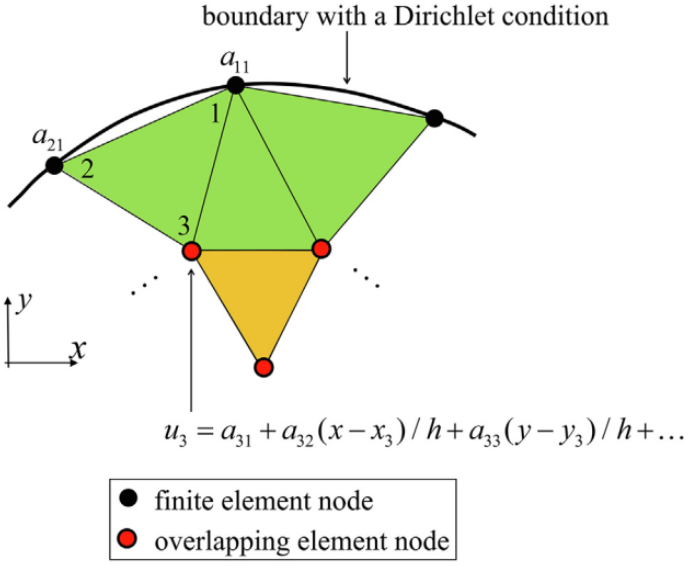


Fig. 3.8 Coupling element used to impose displacement boundary condition

In this approach, we can directly impose the displacement (Dirichlet) boundary conditions, and if desired also the traction (von Neumann) boundary conditions in the usual way onto the boundary defined by the traditional element nodes. On the other hand, the traction boundary conditions can also be applied onto a boundary containing only overlapping finite element nodes. With this approach, the resulting discretization is always stable, hence the corresponding stiffness matrix is always positive definite.

However, with the interpolation functions of the traditional and overlapping elements available, we still need to establish the functions for the coupling elements. These interpolation functions must account for the use of traditional and overlapping element nodes in the element. To formulate the element interpolation functions we resort back to the basic equation, Eq. (3.1), and use $u = \sum_{I=1}^{N} h_I \psi_I$ where

$$\psi_I = \alpha \;\; ; \quad I \in FEn$$
$$\psi_I = \sum_{K \in OFEn} \phi_K^I u_K + \sum_{K \in FEn} \phi_K^I \alpha \quad ; \quad I \in OFEn \tag{3.7}$$

and

$$\alpha = \sum_{K=1}^{N} h_K \, a_{K1} \tag{3.8}$$

with $K \in$ OFEn and $K \in$ FEn summing, respectively, over the number of overlapping and traditional finite element nodes and N is the total number of nodes of the coupling element.

The ϕ_K^I have been defined in Eq. (3.1b, c) and the a_{K1} are defined in Eq. (3.4), that is, they are the traditional degrees of freedom. For example, for the 3-node coupling element with local nodes 1-2-3 shown in Fig. 3.8, we have $a_{11} = u_1$, $a_{21} = u_2$ and u_3 is given in Eq. (3.4). These interpolation functions also satisfy the basic requirements for convergence.

3.3 Testing of the overlapping finite elements

The testing of the overlapping elements consists of first investigating whether some fundamental properties hold, and then using the elements in various problem solutions.

3.3.1 Basic considerations

For the overlapping finite elements to be effective, various basic properties need to hold and we investigate these properties in this section.

Positive definiteness of coefficient stiffness matrix

We use in practice $\beta > 0$ in Eq. (3.2), and since then each overlapping element does not contain a spurious zero energy mode, assuming that a sufficiently high order of numerical integration is used, see Section 3.2.2, the assemblage of elements in an arbitrary mesh leads to a positive definite stiffness matrix provided proper displacement boundary conditions are imposed. Whenever any boundary condition needs to be imposed on the surface of an analysis part, we model the volume at that surface with one layer of coupling elements to impose the boundary condition of displacements or tractions, see Fig. 3.8. Hence the boundary conditions are applied on traditional finite element nodes (which carry only the usual degrees of freedom as in traditional finite element analysis). Then using 3-node triangular two-dimensional elements and 4-node tetrahedral three-dimensional elements, we can even have that $\beta = 0$ because the overlapping finite elements and coupling elements then reduce to finite elements with covers, see J Kim and KJ Bathe [A, B]. There is then no "linear dependency problem" in the governing equations as observed in generalized finite element methods, see C.A. Duarte, I. Babuška and J.T. Oden [A] and R. Tian, G. Yagawa and H. Terasaka [A].

Derivation of β and considerations on stability and accuracy

The magnitude of the parameter β affects the solution accuracy and the conditioning of the governing stiffness matrix. Since with $\beta = 0$ we have the method of finite elements with interpolation covers, we see that the elements reproduce polynomial fields of one order higher than the nodal polynomial used (and see the above comments regarding the use of triangular and tetrahedral elements). Hence, as $\beta \to 0$, in general we can expect a higher solution accuracy.

The interpolation functions using the parameter β are derived and studied in S. Lee and K.J. Bathe [A]. In this publication, it is shown how the functions arise when rational functions are interpolated, and the solution accuracy and conditioning numbers of the stiffness matrices using β in test problems are given. These studies and those in S. Lee and K.J. Bathe [B, C] and W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R C.Mesquita [D] and industrial solutions using ADINA, see KJ. Bathe [N], show that a good value to choose is $\beta = 0.01$, but tests were conducted also using $\beta = 0.03$ and 0.10. However, as mentioned already, it is important to use for better matrix conditioning coupling elements to impose the displacement boundary conditions. If ill-conditioning were to arise, the value of β should simply be increased to using $\beta = 0.10$, then good solutions are still obtained and the conditioning may then even be better than using traditional higher-order elements.

Distortion-insensitivity and convergence rates

A traditional geometrically distorted element may lose its polynomial completeness and hence predictive capability, see N.S. Lee and K.J. Bathe [A] or K.J. Bathe [A]. This loss in solution accuracy can be

significant, in particular when a coarse mesh is used in engineering practice.

However, the overlapping finite elements are distortion-insensitive because the proposed overlapping elements keep their polynomial completeness even when geometrically highly distorted, see L. Zhang, K.T. Kim, and K.J. Bathe [A], J. Huang and K.J. Bathe [A] and S. Lee and K.J. Bathe [A].

The distortion insensitivity also means that the convergence rates are little affected by element distortions. For example, when the p-th order polynomial basis is used, the formulation still achieves the $2p$-th order convergence of strain energy, as given in the relation (3.6).

3.3.2 Static analyses

In this section we briefly illustrate the fundamental characteristics of the overlapping finite elements. Further solutions can be found in S. Lee and K.J. Bathe [A, B], J. Huang and K.J. Bathe [A, B], L Zhang and K.J. Bathe [A, B] and see L. Zhang, K.T. Kim, and K.J. Bathe [A].

Thin beam problem

The thin beam problem in two dimensions considered in Fig. 3.9 is used to identify the effect of mesh distortion on the predictive capability of the element. We use the length e to represent the degree of distortion and compare the solutions obtained with the 4-node overlapping finite element using the quadratic basis with those obtained using the traditional 9-node element.

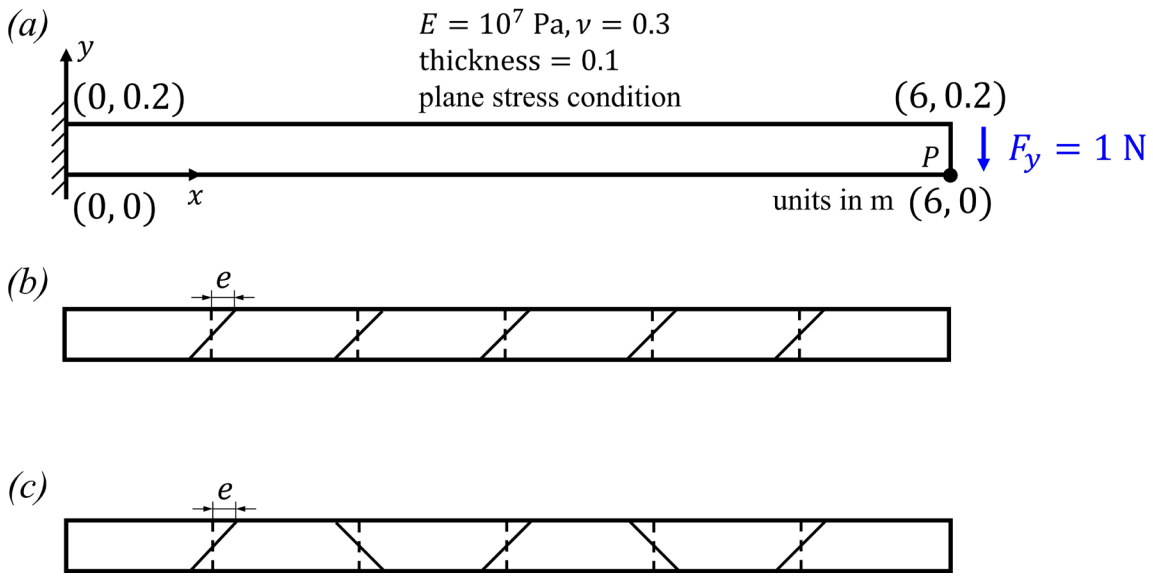


Fig. 3.9 Thin beam problem; (a) Description of the bending problem; total applied force is 1 N; (b) Parallelogram mesh used; (c) Trapezoidal mesh used.

Table 3.2 Normalized *y*-direction displacement at the tip (*x*=6, *y*=0); for the normalization, the reference displacement is -0.1081m.

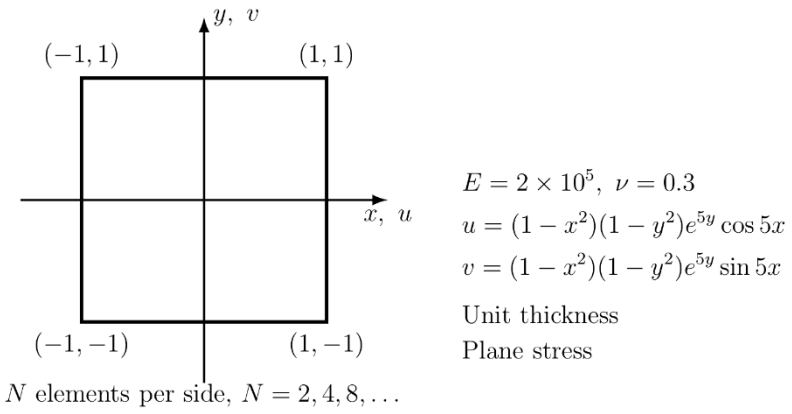
9-node FE (72 dofs)	*e = 0 m*	*0.1*	*0.2*	*0.3*	*0.4*
parallelogram	0.9901	0.9813	0.9397	0.8770	0.8252
trapezoidal		0.9811	0.9234	0.8422	0.7966
4-node OFE (156 dofs)	0	0.1	0.2	0.3	0.4
parallelogram	0.9909	0.9917	0.9925	0.9920	0.9905
trapezoidal		0.9913	0.9910	0.9903	0.9906

Table 3.2 lists the predicted *y*-displacement of point *P*. We see that the solutions using the overlapping element are uniformly accurate with the same small error regardless of the magnitude of *e* which does not hold when using the 9-node element.

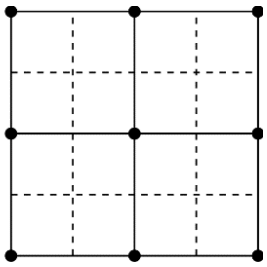
A convergence study

To numerically study the convergence reached using the overlapping finite elements (OFE), we consider the ad-hoc problem shown in Fig. 3.10. The solution for *u* and *v* is zero all around the support. This is a good problem to use for testing a finite element solution scheme because the exact analytical displacement solution of the problem is given, and we apply the body forces corresponding to these displacements, see K.J. Bathe [A].

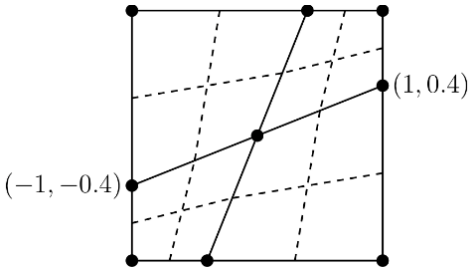
The numerical predictions of the strain energy are plotted in Fig. 3.11. For the distorted meshes, the mean value of *h* is used to denote the element size. We see that the convergence rates obtained agree with the theoretical convergence rate in Eq. (3.8).



a) Problem description



(b) Regular mesh



c) Distorted mesh

Figure 3.10 An ad-hoc problem

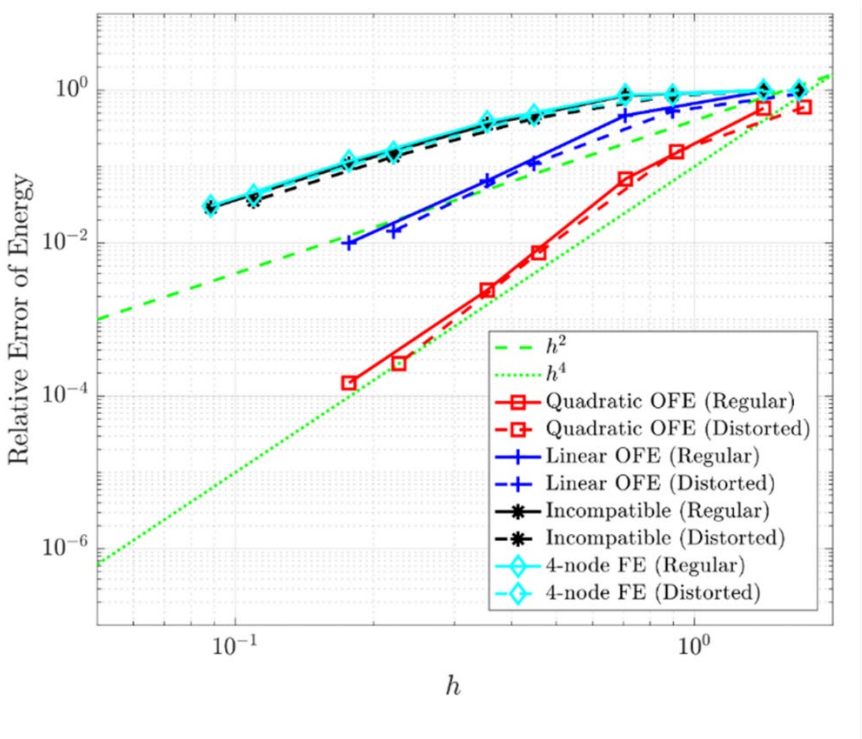


Figure 3.11 The convergence rates for the ad-hoc example problem

3.3.3 Solution of generalized eigenvalue problems

Like in the use of the traditional finite element method, we have the generalized eigenvalue problem

$$\begin{aligned} &\mathbf{K}\mathbf{v}_i = \lambda_i \mathbf{M}\mathbf{v}_i; \quad 0 \le \lambda_1 \le \lambda_2 \le \ldots \le \lambda_n \\ &\mathbf{v}_i^T \mathbf{K}\mathbf{v}_j = \delta_{ij}\lambda_i \\ &\mathbf{v}_i^T \mathbf{M}\mathbf{v}_j = \delta_{ij} \end{aligned} \tag{3.9}$$

where n is the order of the symmetric matrices $\mathbf{K}$ and $\mathbf{M}$, λ_i and $\mathbf{v}_i$ are the i-th eigenvalue and corresponding eigenvector, and δ_{ij} is the Kronecker delta. Of course, the eigenvector now corresponds to the nodal degrees of freedom used in the overlapping finite element formulation. To indicate the effectiveness that we may see when using the overlapping finite elements, we consider the following

problem solutions, see also S. Lee and K.J. Bathe [C]. For all frequency and mode shape solutions, the Bathe subspace iteration has been used, see Chapter 5, K.T. Kim and K.J. Bathe [C].

Frequency analysis of a three-dimensional block

We consider the block shown in Fig. 3.12 to study the effectiveness of using the overlapping finite elements

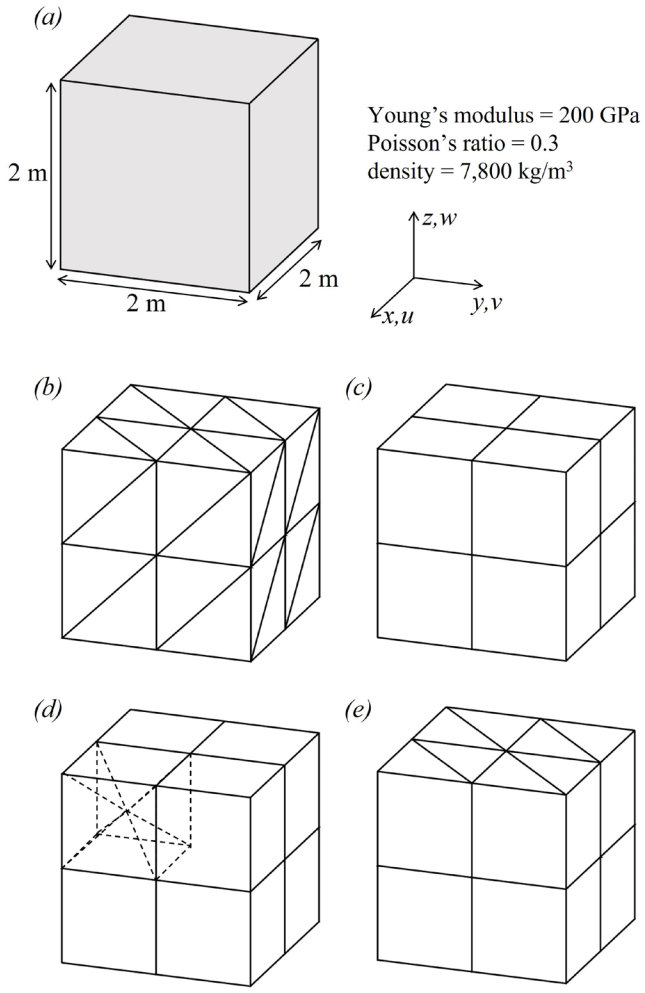


Fig. 3.12 Free vibration analysis of an unsupported cube; (a) Description of the analysis domain; (b) Mesh using tetrahedral elements; (c) using brick elements; (d) using pyramid elements; as shown, a set of six pyramid elements forms a sub-hexahedral domain; (e) Mesh using prism elements; All shown meshes

correspond to $N = 2$ where N is the number of elements per side of the analysis domain. In the models of Table 3.3 we use for N different values, S. Lee and K.J. Bathe [B].

in solving for the ten lowest frequencies. The objective is to solve for the lowest frequency with an error of less than 0.1 percent. To achieve this solution error different mesh densities need to be selected for the different elements used. Table 3.3 gives the results of the study, when using different elements, here "ICM" stands for "incompatible modes used" and "Linear OFE" for "overlapping finite element with the linear basis as nodal degrees of freedom". All solution times are normalized to the largest solution time used (here employing the incompatible mode elements).

Table 3.3 Solution of cube for smallest 10 frequencies; solution times for obtaining the relative error less than 0.1% for the smallest nonzero frequency; the numerical and reference solutions of the frequency are denoted as ω_h^* and ω^*, respectively; all times are normalized by the actual total CPU time of the 8-node incompatible mode element (ICM) solution.

Element (N of mesh)	$\log_{10}\left(\frac{\omega_h^*}{\omega^*}-1\right)$	Numerical integration time	Total CPU time	Half-bandwidth of **K**	Total degrees of freedom
8-node ICM (32)	-3.02	1.9×10^{-3}	1.0×10^{0}	9,510	107,811
4-node linear OFE (8)	-3.06	1.4×10^{-4}	4.7×10^{-3}	792	8,748
5-node linear OFE (5)	-3.01	6.9×10^{-5}	2.3×10^{-3}	1,644	4,092
6-node linear OFE (6)	-3.13	6.1×10^{-5}	1.9×10^{-3}	852	4,116
8-node linear OFE (4)	-3.01	2.1×10^{-5}	3.7×10^{-4}	744	1,500

The results in the table show that the numerical integration time to establish the governing matrices is in all cases small measured on the total solution time. The results also show that in this problem solution, the overlapping finite elements are very effective to solve the problem.

Free vibration analysis of a three-dimensional ring

We consider the solution of the 20 smallest natural frequencies of the ring shown in Fig. 3.13. We use the 8-node linear overlapping finite element with $\beta = 0.01$ and measure the solution time to reach the relative error 0.1% for the smallest frequency. The meshes used contain $m \times m \times 6m$ elements, with the larger number into the circumferential direction.

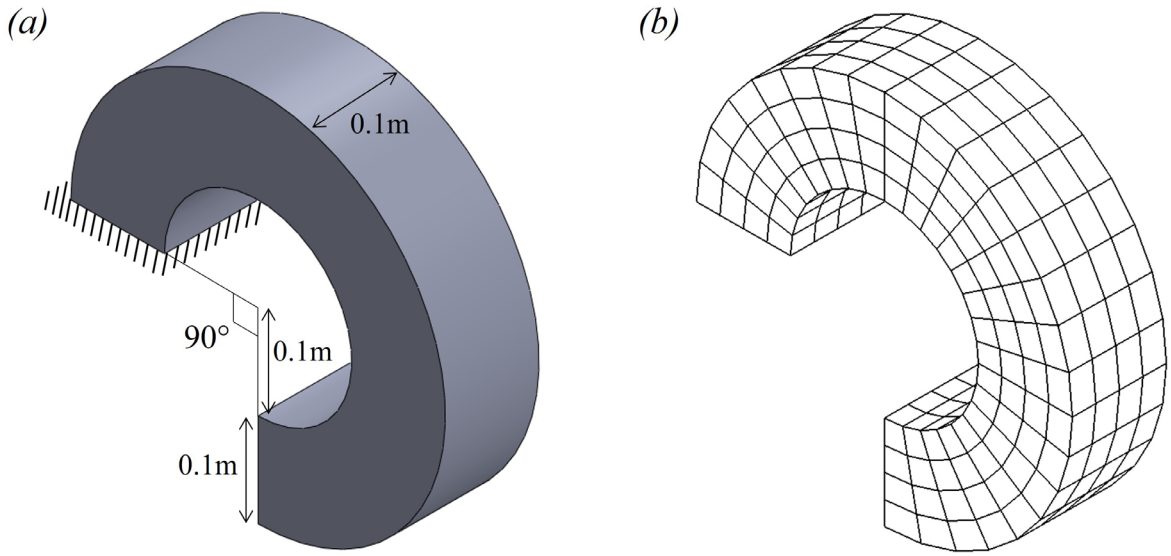


Fig. 3.13 A three-dimensional slender ring; Young's modulus = 200 GPa, Poisson's ratio = 0.3, density = 7,800 kg/m^3; (a) The ring (b) The mesh using $m = 4$, S. Lee and K.J. Bathe [B].

Table 3.4 gives the results obtained where the reference solution is obtained using a very fine mesh of 27-node three-dimensional traditional elements with 198,450 degrees of freedom. The table shows that the use of the overlapping finite element is in this problem solution also effective.

Table 3.4 Solution of ring for smallest 20 frequencies; solution times for obtaining the relative error less than 0.1% for the smallest frequency; the numerical and reference solutions of the frequency are denoted as ω_h^* and ω^*, respectively; all times are normalized by the total CPU time used for the 8-node incompatible mode element (ICM) solution.

Element ($m \times m \times 6m$)	$\log_{10}\left(\frac{\omega_h^*}{\omega^*}-1\right)$	Numerical integration time	Total CPU time	Half-bandwidth of **K**	Total degrees of freedom
8-node ICM (24 × 24 × 144)	-3.06	9.0×10^{-3}	1.0×10^{0}	5,553	270,000
8-node linear OFE (10 × 10 × 60)	-3.07	3.1×10^{-3}	2.1×10^{-1}	3,795	87,483

For additional solutions we refer to S Lee and KJ Bathe [C].

3.3.4 Wave propagation transient analyses

The use of overlapping finite elements can be particularly effective in the solution of special problems because special functions can be used in the nodal degrees of freedom. One such problem area is the transient solution of waves, see K.T. Kim, L. Zhang and K.J. Bathe [A], Y. Chai and K.J. Bathe [A], and K.T. Kim and K.J. Bathe [A].

The accurate solution of wave propagation problems is difficult, or even impossible to achieve, using a traditional finite element discretization because in general waves travel in any direction through the continuum, and for different kinds of waves we have different wave velocities. Using a traditional finite element mesh, the directionality of the mesh and possibly also different sizes of elements impose difficulties because the ratio of element "size" to time step used should be constant. This should hold for a given wave velocity, throughout the solution domain, where the "size" of an element is measured as the distance travelled through the element. In addition, waves of different velocities may travel through the mesh.

Hence, just considering the case of different physical wave velocities, like seen for normal, shear and pressure waves, through the continuum, it is clear that the use of a constant CFL (and hence a constant time step) defined as

$$\mathrm{CFL} = \mathrm{c}\Delta t/\mathrm{h} \tag{3.10}$$

where h is the element size and Δt is the time step used in the time integration cannot be satisfied for different wave velocities c.

For these reasons, the spectral element method is frequently used when accurate solution results are needed, see e.g. P. Zakian and K.J. Bathe and the references therein [A].

Using the overlapping finite elements for two-dimensional wave propagation problems, we can use the bi-linear polynomial functions and trigonometric functions for the local basis, namely, with a local Cartesian coordinate system (x, y)

$$[p_1, p_2, \cdots] =$$

$$\left[\begin{array}{c} 1, x, y, xy, \\ \cos\left(\frac{2\pi x}{\lambda_x}\right), \sin\left(\frac{2\pi x}{\lambda_x}\right), \cos\left(\frac{2\pi y}{\lambda_y}\right), \sin\left(\frac{2\pi y}{\lambda_y}\right), \\ \cos\left(\frac{2\pi x}{\lambda_x}+\frac{2\pi y}{\lambda_y}\right), \sin\left(\frac{2\pi x}{\lambda_x}+\frac{2\pi y}{\lambda_y}\right), \cos\left(\frac{2\pi x}{\lambda_x}-\frac{2\pi y}{\lambda_y}\right), \sin\left(\frac{2\pi x}{\lambda_x}-\frac{2\pi y}{\lambda_y}\right), \\ \cos\left(2\cdot\frac{2\pi x}{\lambda_x}\right), \sin\left(2\cdot\frac{2\pi x}{\lambda_x}\right), \cos\left(2\cdot\frac{2\pi y}{\lambda_y}\right), \sin\left(2\cdot\frac{2\pi y}{\lambda_y}\right), \\ \cos\left(2\cdot\frac{2\pi x}{\lambda_x}+2\cdot\frac{2\pi y}{\lambda_y}\right), \sin\left(2\cdot\frac{2\pi x}{\lambda_x}+2\cdot\frac{2\pi y}{\lambda_y}\right), \cos\left(2\cdot\frac{2\pi x}{\lambda_x}-2\cdot\frac{2\pi y}{\lambda_y}\right), \sin\left(2\cdot\frac{2\pi x}{\lambda_x}-2\cdot\frac{2\pi y}{\lambda_y}\right), \\ \cdots \end{array}\right] \tag{3.11}$$

where λ_x and λ_y are the fundamental wavelengths in the x- and y-directions, respectively. For example, if we consider only a one-dimensional wave travel in the x-direction, the trigonometric functions used

for the x-axis direction wave travel are $\sin\left(\frac{2\pi x}{\lambda_x/n}\right)$, n = 1, 2, … and $\cos\left(\frac{2\pi x}{\lambda_x/n}\right), n = 1, 2, \ldots$ where n needs to be chosen. The use of the bi-linear polynomial functions ensures the linear consistency (reproducing the linear fields like rigid body modes and constant strain states), and the trigonometric polynomial basis enhances the solution accuracy.

This enhancement using the trigonometric functions in the interpolations can of course also be used in the traditional finite element method but the accuracy achieved is not as good as when using the enhancement in the method of finite spheres and in the overlapping finite elements, see S. Ham and K.J. Bathe [A, B]. Some disadvantages of using the method of finite spheres are that special meshing schemes would need to be used and the computational cost (as pointed out in Section 3.1) is very high.

An important point for the overlapping finite elements with the bi-linear polynomial and the trigonometric functions up to order n (we shall refer to them as OFE-TRIn) is that the dispersion error caused by the OFE-TRIn discretization is negligible and almost independent of the propagation direction, for wave modes with $k_h h/\pi < n$ where $k_h = \frac{2\pi}{\lambda_h}$ is the numerical wave number and h is the typical size of the overlapping elements. Therefore, we use this formula and employ the element size to be $h < n\,\lambda_h/2$. This formula has been reached in K.T. Kim and K.J. Bathe [A] and been used with the Bathe time integration method and CFL = 0.125, see also K.T. Kim, L. Zhang and K.J. Bathe [A]. In these solutions the method of finite spheres and overlapping finite elements have been employed.

An analysis using the Bathe time integration scheme in finite element solutions of wave propagations *without* enrichments is given in G. Noh, S. Ham and K.J. Bathe [A]. Here it is shown that in one-dimensional solutions due to the numerical damping properties, the Bathe time integration method with CFL = 1 requires $h < 0.3\,\lambda_h$ for an accurate solution. Hence, in a one-dimensional solution about three 2-node elements are required to model one wave length (and the smallest wave length to be modeled is used). This illustrates the effectiveness of the Bathe method. While accurate in one-dimensional analyses,

in two-dimensional (and three-dimensional) analyses larger errors arise due to the non-isotropy of the elements, even when a uniform mesh is used.

However, considering the overlapping finite elements, an important characteristic of a discretization is that the elements are quite insensitive to geometric distortions, hence to the directions of wave travels, even through a nonuniform mesh. When using the TRIn enrichment with n=3, we might use as the coarsest mesh one element size h per wavelength λ_h and will have 3 cycles of the harmonic functions in the element domain. Hence with CFL = 0.125 we have that the wave is integrated with 8 time steps to advance through the element length and each of these time steps advancing the response for (roughly) 1/3rd of a sine or cosine cycle of the harmonic assumption. In this approach we obtain a monotonic convergence of the solution for any wave direction when refining the spatial discretization and/or the time discretization.

In the following we demonstrate the solution accuracy obtained, see also K.T. Kim and K.J. Bathe [B].

A one-dimensional wave propagation problem

We consider a clamped-free elastic bi-material bar, with zero initial conditions, subjected to a unit step load $H(t)$ at its free end, as described in Figure 3.14. The bar has length $L = 4$ m, consists of two segments, each of the same length but with different materials for which the wave speeds are $c_1 = 40\sqrt{5}$ m/s and $c_2 = 20\sqrt{2}$ m/s, respectively, and we want to solve for the axial stress at the interface of the materials, point A. We consider the same problem solution also in Sections 4.2.3 and 4.3.3.

Figure 3.14 An elastic bi-material bar impacted by a unit step axial pressure. Regarding Figs. 3.14 to 3.20 and further information, see K.T. Kim and K.J. Bathe [B].

Figure 3.15 shows the solutions of the problem using the OFE-TRI3 discretization with the implicit β_1/β_2 - Bathe method, see Section 4.3 and M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran [A] and K.J. Bathe [B]. We see that an accurate solution is obtained using a time step size $\triangle t$ set by the smaller wave speed with CFL = 0.125, where it is clearly seen that spurious oscillations are almost not present. An even more accurate solution is obtained if the larger wave speed is used to select the time step size; see Fig. 3.15 (b). It is also important to note that an accurate solution can be achieved even using a non-uniform mesh, which may need to be employed in practical analysis; see Fig. 3.15 (c).

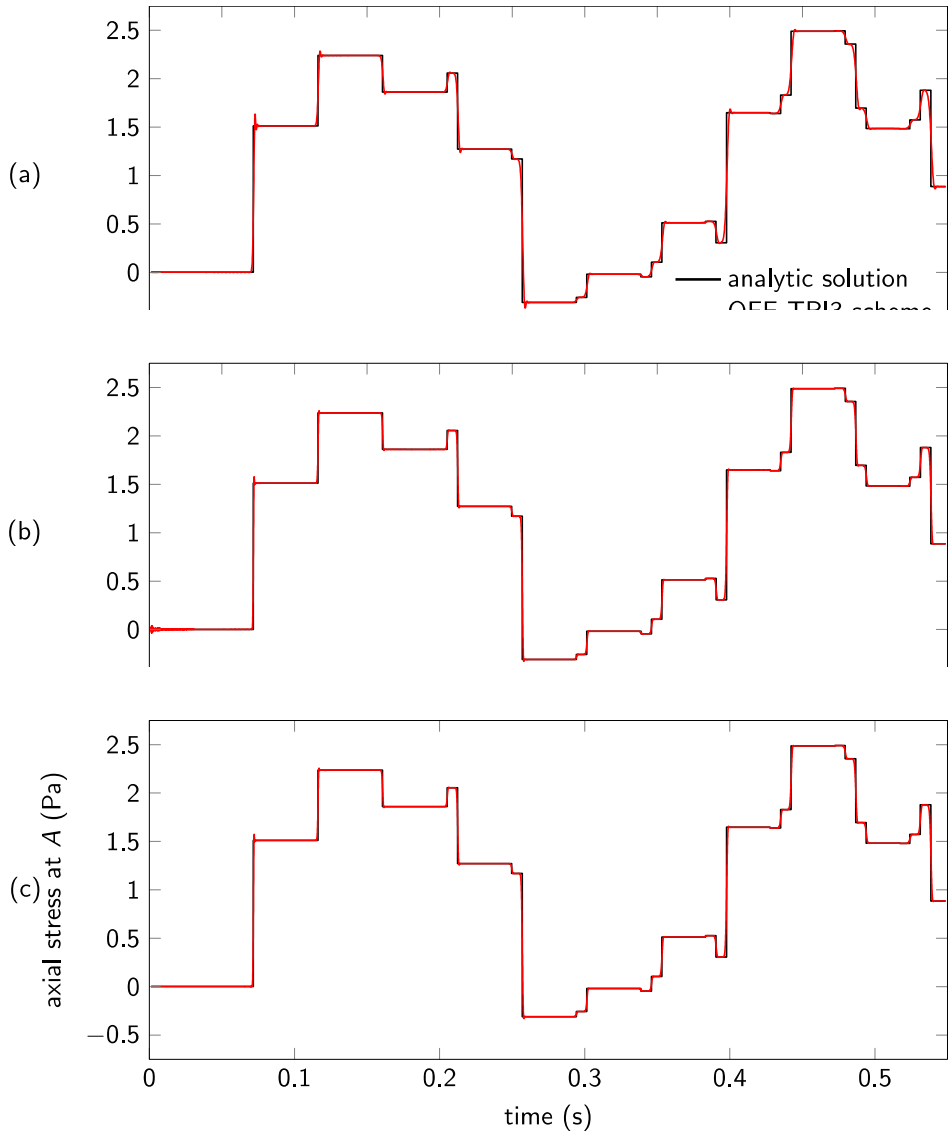


Figure 3.15 Predicted axial stress histories at the point A using the OFE-TRI3 with the β_1/β_2-Bathe method ($\gamma = 0.5$ and $\beta_1 = 0.35$ with $\beta_2 = 2\beta_1$), A uniform mesh of N = 160 elements (80 elements in each material) is used and the time step size is set using (a) the wave speed of the second material (the smaller wave speed) and (b) that of the first material (the larger wave speed) both with CFL = 0.125. (c) A non-uniform mesh of N = 160 elements (80 elements for each material) is used and the time step size is set using the larger wave speed with h = L / N and CFL = 0.125. The non-uniform mesh is constructed by randomly distributing the nodes under the condition that the size of each element is $0 < \Delta x < 2L/N$.

The analysis of Lamb's problem subjected to step loadings

A valuable test problem to consider is a concentrated line load applied to the surface of an infinite medium as shown in Fig. 3.16. The figure also lists the pressure, shear and Rayleigh wave speeds.

Fig. 3.16 Line load applied to the free surface of an infinite elastic domain; the pressure, shear and Rayleigh wave speeds are as shown

The line load $F_c(t)$ consists of three step functions, defined as

$$F_c(t) = 2 \times 10^6[H(0.15 - t) - 3H(0.1 - t) + 3H(0.05 - t)]$$

The discontinuous loading causes high-frequency wave modes, which renders the problem difficult to solve, in particular because of the three different wave speeds in the medium.

We use a uniform mesh of $N = 160 \times 160 \times 2 = 51{,}200$ elements and a non-uniform mesh of N = 30,446 elements where the nodes are increasingly denser toward the free surface. The calculated displacements at two receivers are given in Fig. 3.17, where it is seen that all elastic waves are well predicted with negligible oscillations using the OFE-TRI2 with the β_1/β_2 -Bathe method.

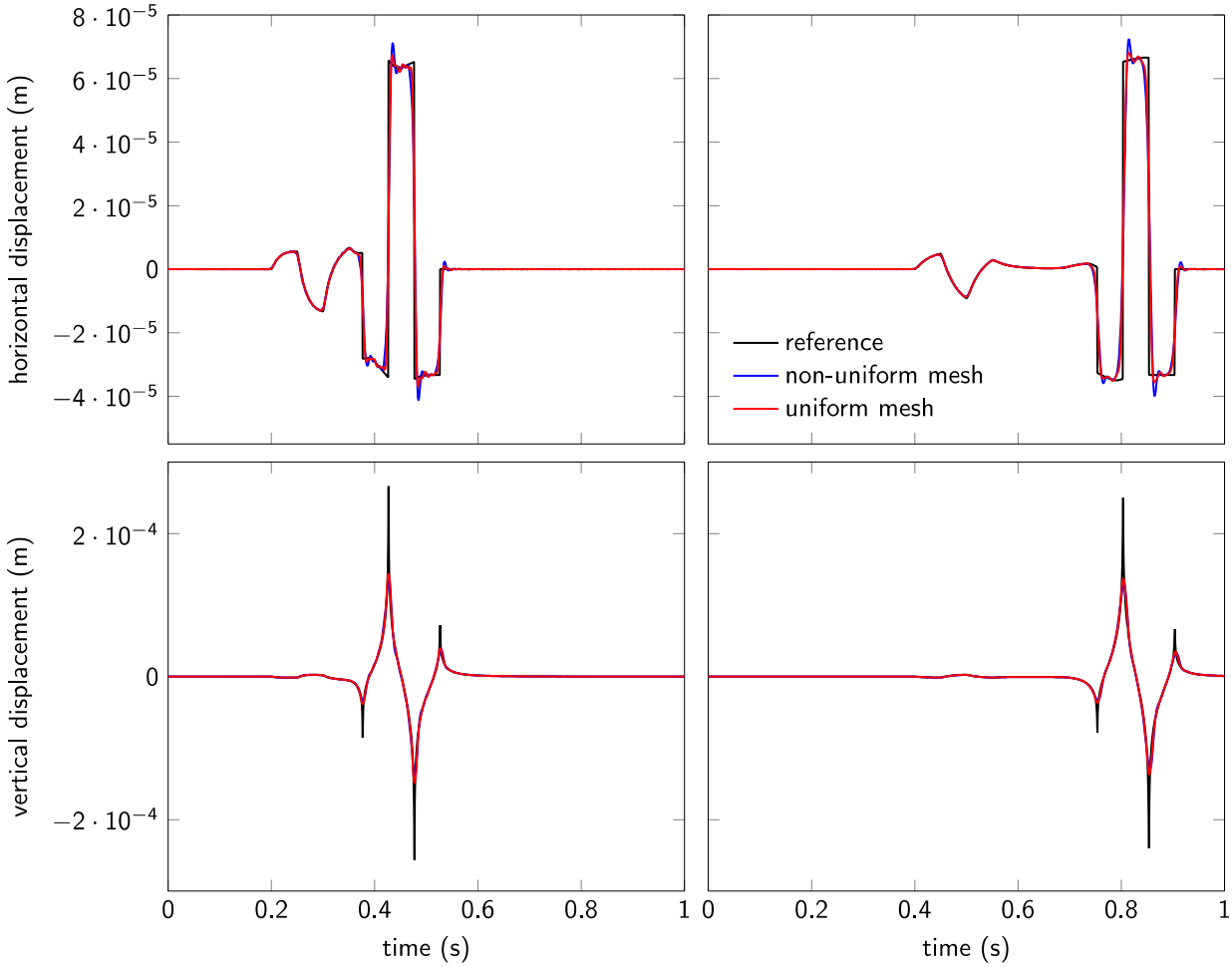


Figure 3.17 Horizontal and vertical displacements of the elastic medium subjected to the step loading on the free surface. The time histories of the displacements at **x** = (640,0) (left) and at **x** = (1280,0) (right) are shown. The OFE-TRI2 with uniform and non-uniform meshes (51,200 elements and 30,446 elements, respectively) and the β_1/β_2-Bathe method ($\gamma = 0.5$ and $\beta_1 = 0.35$ with $\beta_2 = 2\beta_1$) with CFL=0.125 are used. The analytical reference solution is based on the work in J Miklowitz [A].

The solution of wave propagations through two media with a crack on the interface

We consider here the solution of the problem described in Figure 3.18.

For this wave propagation problem, we do not have an analytical solution, and hence we use three meshes of triangular elements for solution, a coarse, finer and finest mesh. The element grading is even in the horizontal direction but increases in density towards the surface for each mesh, reaching at the surface about 1/3rd the element area used at the bottom.

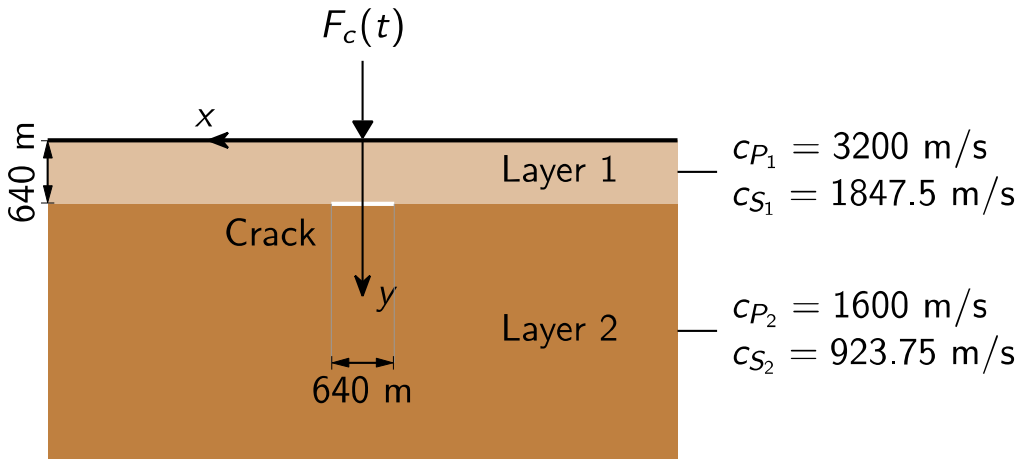


Figure 3.18 A semi-infinite medium consisting of two elastic layers with a crack in plane strain conditions; a concentrated line load is applied on the free surface.

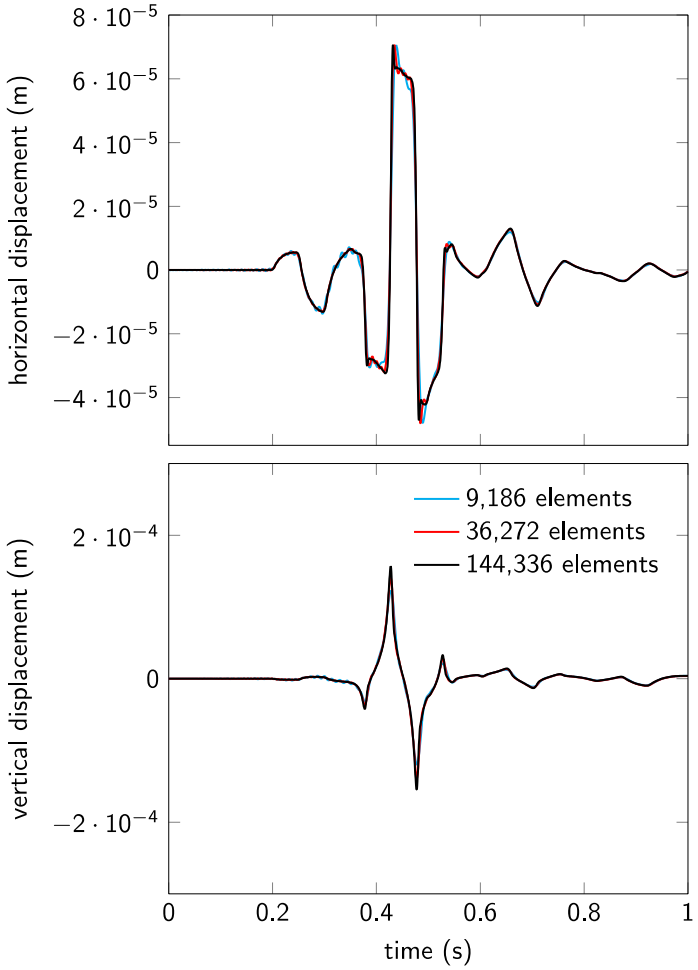


Figure 3.19 Horizontal and vertical displacements of the two-layered elastic medium with a crack subjected to the step loading on the free surface. The time histories of the displacements at $\boldsymbol{x} = (640,0)$ are shown. The OFE-TRI2 with non-uniform meshes of 9,186 elements, 36,272 elements, and 144,336 elements and the β_1/β_2 -Bathe method ($\gamma = 0.5$ and $\beta_1 = 0.35$ with $\beta_2 = 2\beta_1$) with CFL=0.125 are used.

Figures 3.19 and 3.20 show the results when the step loading (3.12) is applied. We see in Fig. 3.19 that even for this difficult problem, the solution scheme using the overlapping finite elements and the β_1/β_2 -Bathe method gives good results even when the coarse mesh is employed.

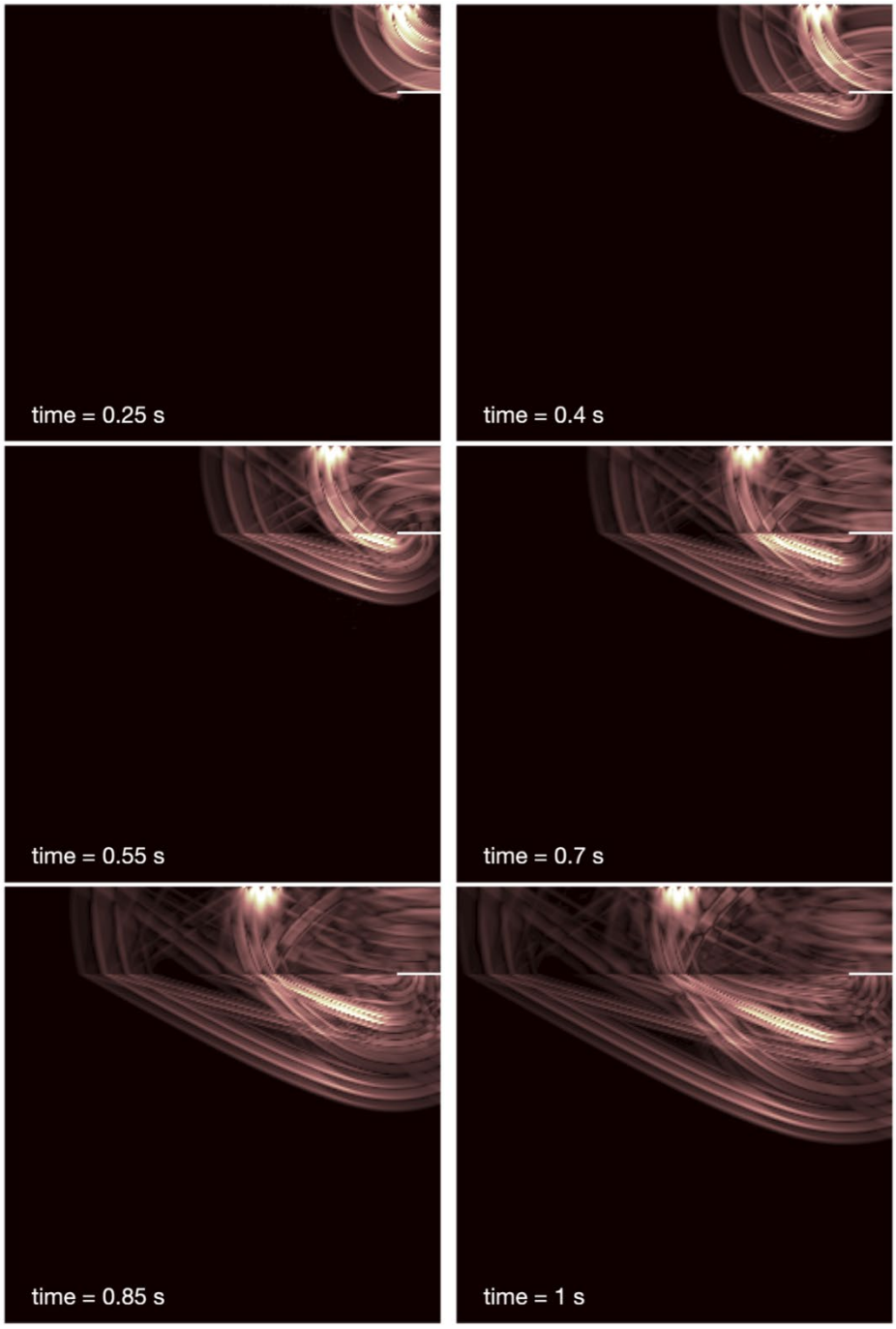


Figure 3.20 Snapshots of von Mises stress distributions of the two-layered elastic medium with a crack at various observation times; the step loading is applied on the free surface. The OFE-TRI2 with a non-uniform mesh of 36,272 elements and the β_1/β_2 -Bathe method ($\gamma = 0.5$ and $\beta_1 = 0.35$ with $\beta_2 = 2\beta_1$) with CFL=0.125 are used.

3.4 Overlapping elements for incompressible media

As we illustrated in the above sections, the overlapping elements perform very well in static and dynamic analyses. Major assets of these elements are that they are quite distortion insensitive, and they can be used in the same way as traditional elements. Hence traditional element meshes can directly be used for the overlapping elements. But more effective meshes may be used taking advantage of the insensitivity of element distortions, see Section 3.6.

However, one aspect, that we have not discussed yet, is the use of the overlapping elements in the analysis of incompressible media – which we encounter abundantly in solids and fluids. The objective of this section is to consider such media.

3.4.1 Formulation for incompressible solids and fluids

The formulation of the overlapping elements for incompressible solids (and low Reynolds number incompressible fluids) is as discussed in Section 3.2, except that we need to now add the constraint of incompressibility (the volume strain to be zero) to be satisfied at each point of the analysis domain, see KJ Bathe [A],

$$u_{i,i} = 0 \tag{3.12}$$

where the u_i are the unknown displacements, with i = 1, 2 in two-dimensional solutions, and i = 1, 2, 3 in three-dimensional solutions. In finite element analysis, we enforce the constraint in the weak sense using

$$\int_{Vol} \overline{p}\, u_{i,i}\, dVol = 0 \tag{3.13}$$

where Vol denotes the domain of analysis and $\overline{p}$ is the virtual pressure given by any function in $L^2(Vol)$. In the finite element solution, the constraint in Eq. (3.13) is satisfied over each finite element using the finite element interpolation spaces, that is, more specifically using the interpolation functions spanning these spaces.

In practice, the material may only be "almost" incompressible as seen by a large bulk modulus κ and Eq. (3.13) is then changed to

$$\int_{Vol} (\frac{p}{\kappa} + u_{i,i})\, \overline{p}\, dVol = 0 \qquad (3.14)$$

where p is the physical pressure to be calculated. In either case, the basic principle of virtual work equations governing the analysis of the solid are then written using the volumetric strain and pressure, and the deviatoric strains and stresses, see K.J. Bathe [A].

In the analysis of incompressible fluid flows, the same procedure is used but of course, the virtual work equations are then those governing the fluid flow, with the unknown velocities v_i to be solved for instead of the unknown displacements u_i, see K.J. Bathe [A].

Focusing on the overlapping finite elements, for the analysis of solids and fluids, the same virtual work equations hold but the interpolation functions of the overlapping elements for the displacements of solids or velocities of fluids need to be used, see Section 3.3 and W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira, and R.C. Mesquita [D].

The important point is next to identify suitable finite elements for the analysis of the incompressible media. Such elements are given for the traditional finite element solutions in K.J. Bathe [A], the elements are characterized by the interpolations used for the displacements (or velocities) and the pressure. Suitable 4-node overlapping finite elements for two-dimensional solutions are presented in W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira, and R.C. Mesquita [D]. The elements have been tested using the inf-sup test, see Section 3.4.2, with example solutions to problems.

Let us denote the finite element nodal degrees of freedom as given in Table 3.5. Here OFE denotes that we are considering an overlapping finite element and the right subscript gives the local basis used at the element nodes, hence, e.g., the "0" denotes that only the traditional nodal degree of freedom is used.

The degrees of freedom (defining the spaces) given in Table 3.5 are employed at the nodes for the displacements (or velocities) and the pressure. Typically, we would use the displacement / pressure nodal variables given by $\mathrm{OFE}_1/\mathrm{OFE}_0$.

The performance of various overlapping finite elements is presented in the next sections.

Table 3.5 Nodal degrees of freedom for the 4-node overlapping finite element

Space	Local bases
OFE_0	$\{1\}$,
OFE_1	$\{1, \hat{x}, \hat{y}\}$,
OFE_2	$\{1, \hat{x}, \hat{y}, \hat{x}^2, \hat{x}\hat{y}, \hat{y}^2\}$.

3.4.2 Inf-sup testing of overlapping finite elements

The conditions of continuity and coercivity of the overlapping element discretization are satisfied, and hence we can solve the governing equations, but we need to still check whether the relevant inf-sup condition is satisfied. If this condition also holds, then the element formulation will perform well, in particular will not lock in incompressible analysis. This condition is particularly applicable to mixed formulations, see K.J. Bathe [A].

We used the inf-sup test already in Section 2.4.2 to test the performance of the brick element, discussed in that section, for the analysis of solids and also in shell solutions. We use the same procedure to test the overlapping finite elements. Again, we perform the testing numerically as proposed by D. Chapelle and K.J. Bathe [D].

The overlapping finite elements discussed correspond to a u/p mixed formlation for which the inf-sup condition is:

There is some constant $\beta_h > 0$ such that

$$\inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in V_h} \frac{\int_V q_h \, div \, \mathbf{v}_h \, dV}{\|q_h\| \|\mathbf{v}_h\|} = \beta_h \geq \beta > 0 \qquad (3.15)$$

where Q_h is the space of assumed pressures, and V_h is the space of assumed displacements in the analysis of solids and assumed velocities in the analysis of fluid flows, that is

$$Q_h = \{ q_h \in L^2 \} \text{ with } \quad Q_h \subset Q \qquad (3.16a)$$

and

$$V_h = \{ \mathbf{v}_h \mid \mathbf{v}_h \in L^2, \frac{\partial (v_h)_i}{\partial x_j} \in L^2, (v_h)_i = 0 \text{ on } S_U \quad \text{with } i, j = 1,2,3 \} \qquad (3.16b)$$

with $V_h \subset V$.

Here Q and V are the spaces of actual pressures and displacements or velocities, respectively, and are defined as in (3.16a) and (3.16b) but dropping the subscript *h,* see K.J. Bathe [A].

For the solutions considered here, the spaces in (3.16a) and (3.16b) are defined by the overlapping finite element interpolations with the degrees of freedom in Table 3.5.

In essence, the condition requires that as a mesh is refined, the value of the inf-sup expression will not continue to decrease but be bounded from below by a constant, which we denote here by β. If the inf-sup condition is satisfied, the element is stable and indeed optimal in its approximation properties.

Since we do not have an analytical proof, we select an analysis problem and establish numerically whether for a selected mesh and its refinements the values β_h in (3.15) do not decrease as the mesh is refined but are bounded by the value β. For each mesh case, the inf-sup expression giving β_h is evaluated through an eigenvalue solution. The test has been applied abundantly to evaluate element formulations. We again refer for the details to perform an inf-sup test to D. Chapelle and K.J. Bathe [D] and K.J. Bathe [A, B].

To demonstrate the test, we consider the fluid flow problem in Figure 3.21, see W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R.C. Mesquita [D] where more details are given.

The fluid is incompressible and the solution is obtained by solving the Navier-Stokes equations but omitting all nonlinear terms. Figure 3.22 shows the predicted velocity and pressure fields using the reliable Q_2 / Q_1 element, see K.J. Bathe [A]. The element is defined by the traditional biquadratic interpolation for the velocity components and the bilinear interpolation for the pressure and satisfies the inf-sup condition.

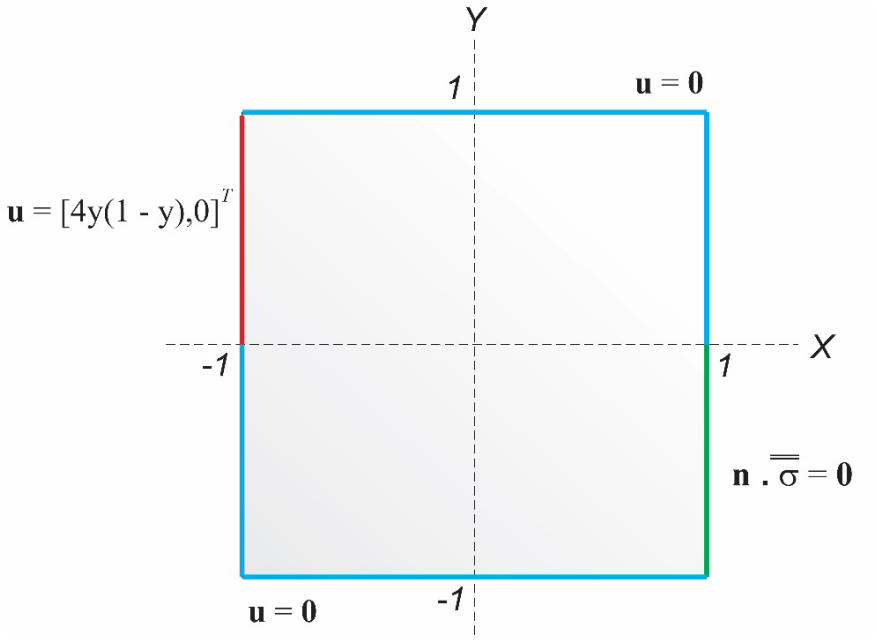
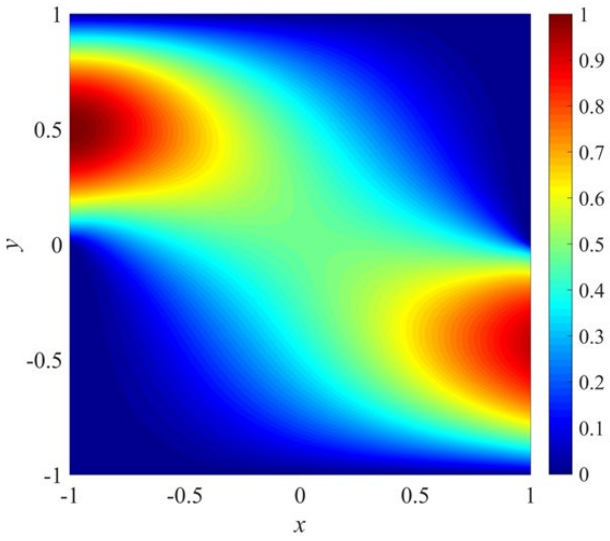
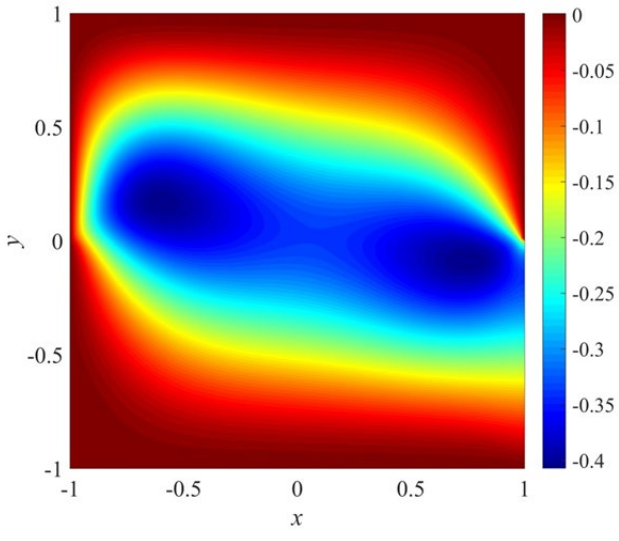


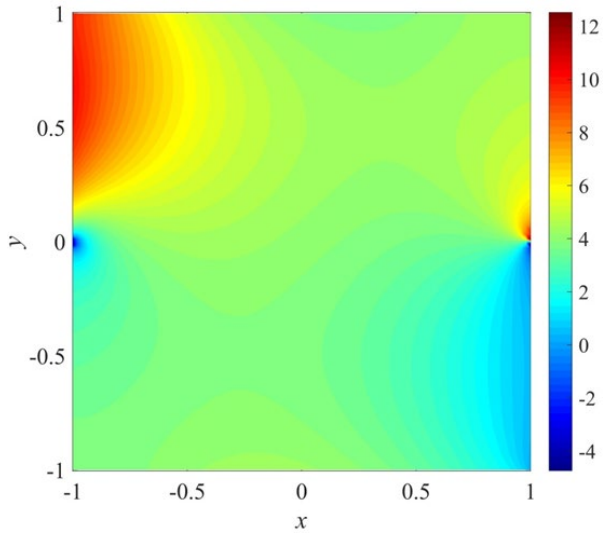
Figure 3.21 Fluid flow problem with in-flow condition on the boundary marked red, out-flow condition on boundary marked green and the no-slip condition on the boundary marked blue. Regarding Figs. 3.21 to 3.32 and a further discussion, see W.L. Nicomedes, K.J. Bathe, FJS Moreira and R.C. Mesquita [D].



Horizontal velocity



Vertical velocity



(c) Pressure

Figure 3.22 Finite element solution using the traditional Q_2 / Q_1 element, with a uniform mesh of 128×128 square elements.

As well known, the traditional Q_1 / Q_0 element (with the bilinear interpolation of the velocity components and the assumption of constant pressure in the element) does not satisfy the inf-sup condition and is hence unstable. Figure 3.23 shows some results of the pressure predictions. Performing the inf-sup test we obtain the results shown in Fig. 3.24.

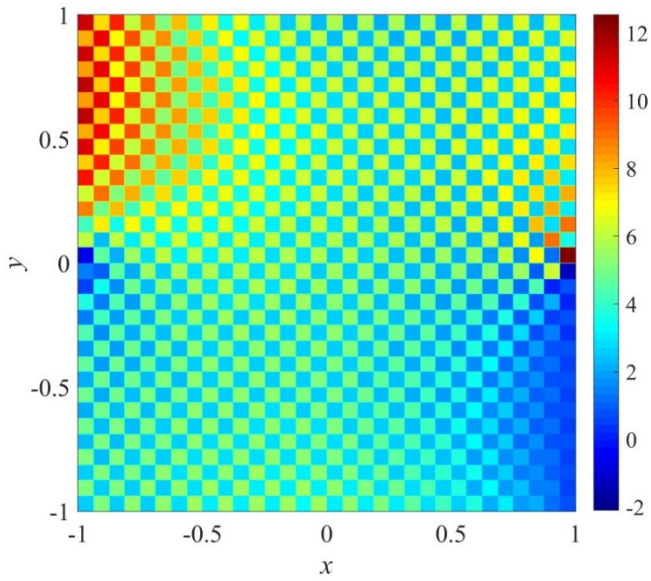


Figure 3.23 Predicted pressure using the traditional Q_1/Q_0 element with a 32x32 uniform mesh of elements.

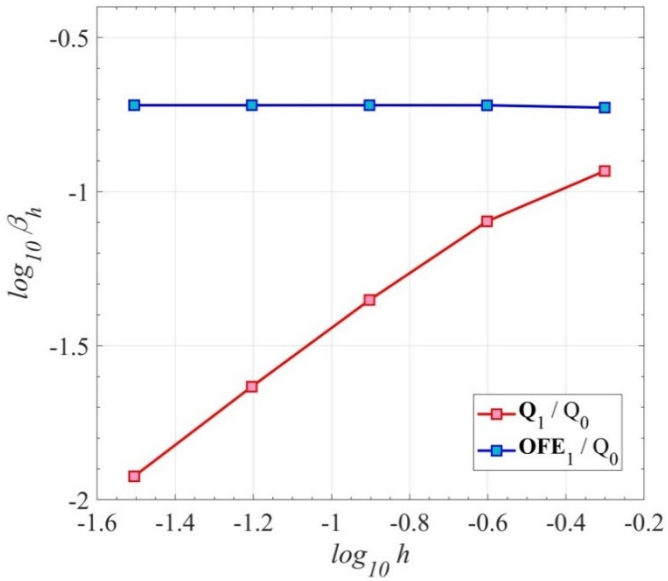


Figure 3.24 Inf-sup test results for the traditional Q_1/Q_0 element and the OFE_1/Q_0 element when the meshes are undistorted.

Figure 3.24 shows that the inf-sup values of the Q_1/Q_0 element meshes decrease, as expected, and also shows that the meshes using the OFE_1/Q_0 overlapping finite element, with the constant pressure assumption (as for the traditional Q_1/Q_0 finite element) and the linear basis as nodal degrees of freedom (see Table 3.5), give inf-sup values that do not decrease as the mesh is refined. These results indicate that this overlapping finite element is stable, but the testing needs to be continued using additional mesh configurations and, conservatively, also another problem solution.

Figure 3.25 shows the inf-sup test results of the OFE_1/OFE_0 and OFE_2/OFE_0 elements and

indicates that these elements are also stable. These elements use the nodal degrees of freedom of Table 3.5 for the velocity (or displacement) and pressure, and hence have a pressure degree of freedom at each node. Hence with these assumptions, the pressure is continuous over the element mesh.

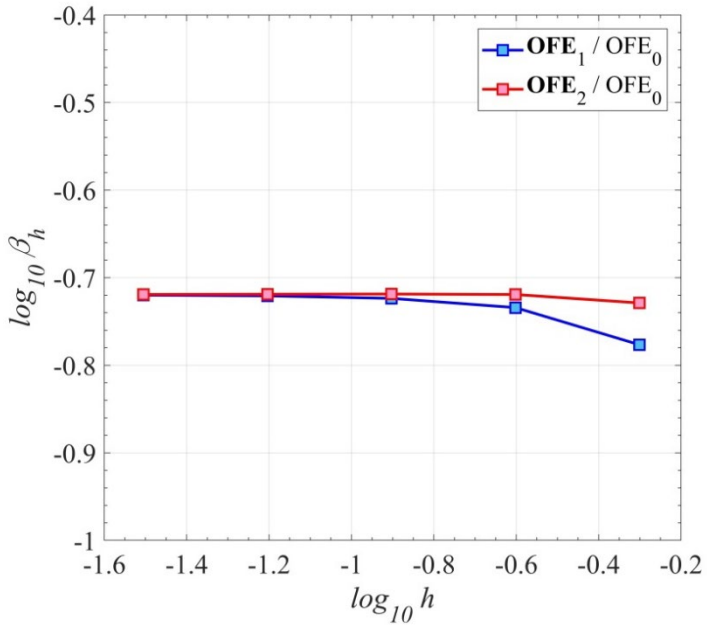


Figure 3.25 Inf-sup test results for the OFE_1/OFE_0 and OFE_2/OFE_0 elements , when the meshes are undistorted.

However, we also need to assure that the elements are effective when the mesh is distorted. For this reason, we also consider the distorted meshes described in Fig. 3.26 and calculate the inf-sup values for some elements that we recommend for use. Figure 3.27 shows the values for the OFE_1/Q_0 and OFE_1/OFE_0 elements when the meshes are distorted. In each case the results are compared with those using the Q_1/Q_0 element when undistorted, which is not stable

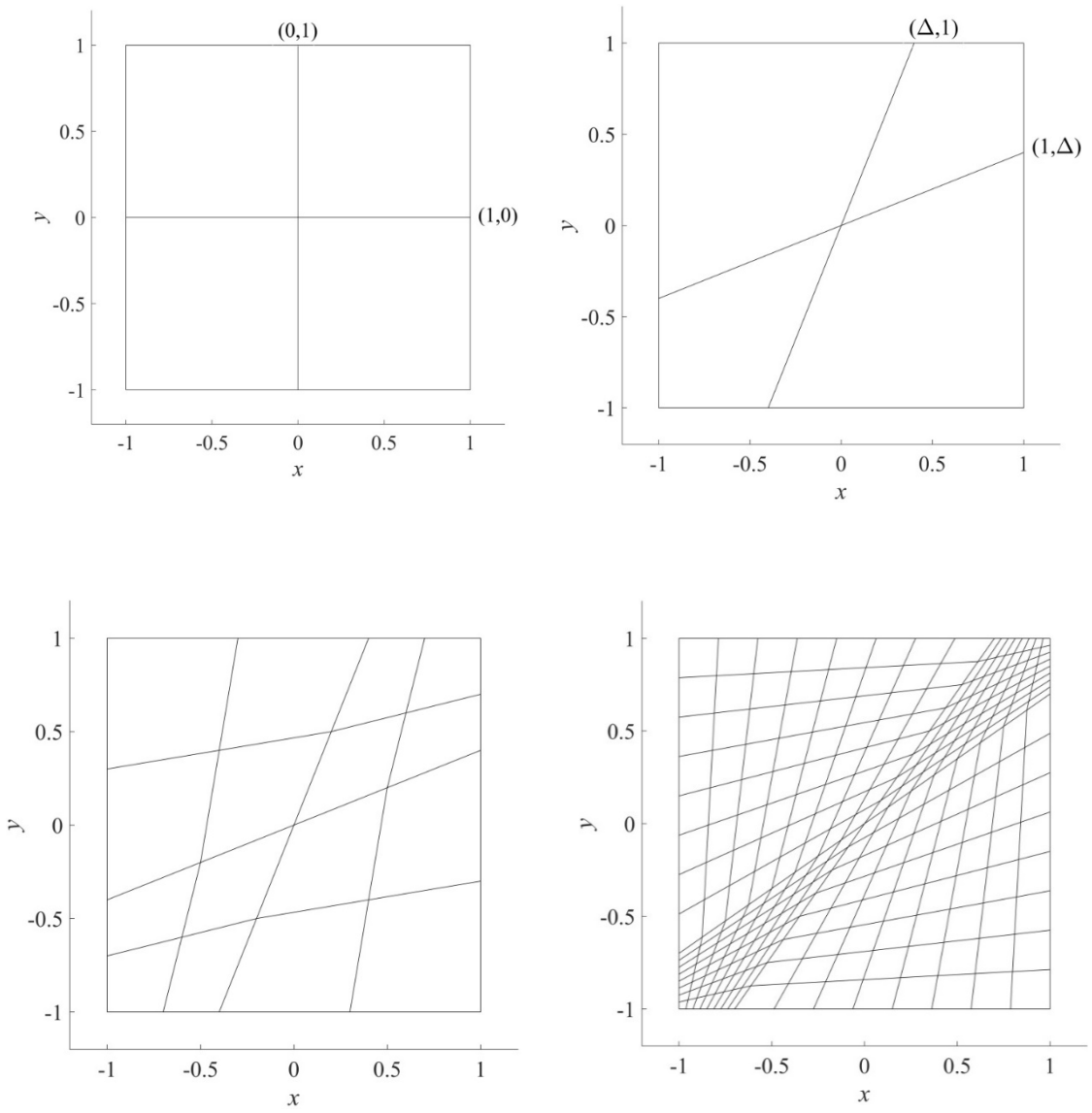


Fig 3.26 The original undistorted mesh, N=2, and distorted N = 2 and N = 4 element meshes for $\Delta = 0.4$, and distorted N = 16 element mesh for $\Delta = 0.7$. To reach the 4 x 4 element mesh, that is, $N = 4$, the elements in the distorted 2 x 2 element mesh are each subdivided into 4 elements. The finer meshes are similarly reached.

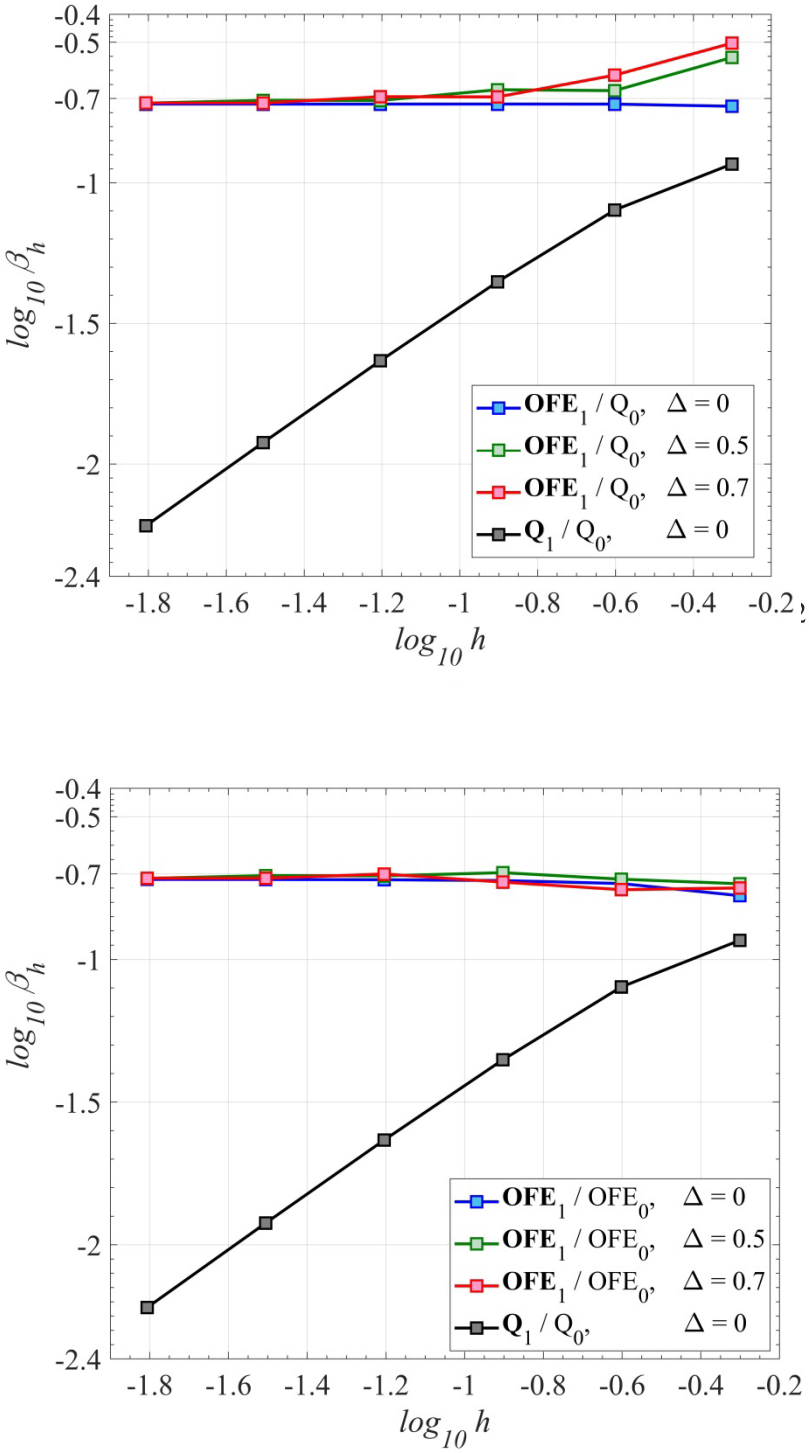


Figure 3.27 Inf-sup test using the distorted meshes shown in Fig. 3.26. The curves for the

overlapping elements OFE_1 / Q_0 and OFE_1 / OFE_0 level off, that is, do not decrease as the meshes are refined.

Considering Fig. 3.27, we see that each of the new elements considered passes the inf-sup test also when the meshes are distorted. While not an analytical proof that the inf-sup condition is satisfied, the inf-sup test has been passed and the OFE_1 / Q_0 and OFE_1 / OFE_0 elements very likely satisfy the inf-sup condition. Hence these elements are reliable and effective and can be recommended for use.

3.4.3 Illustrative solutions

We consider the solution of a low-Reynolds number fluid flow in a channel, see Fig. 3.28. The fluid is incompressible and we consider the cases Re = 50 and Re = 250. A regular mesh of 5, 579 undistorted OFE_1 / OFE_0 overlapping finite elements is used to solve the problem.

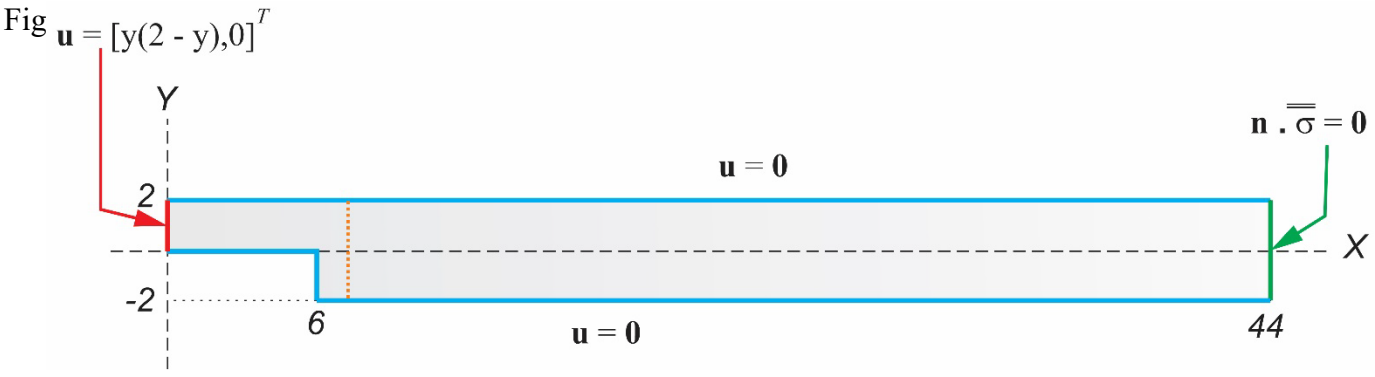
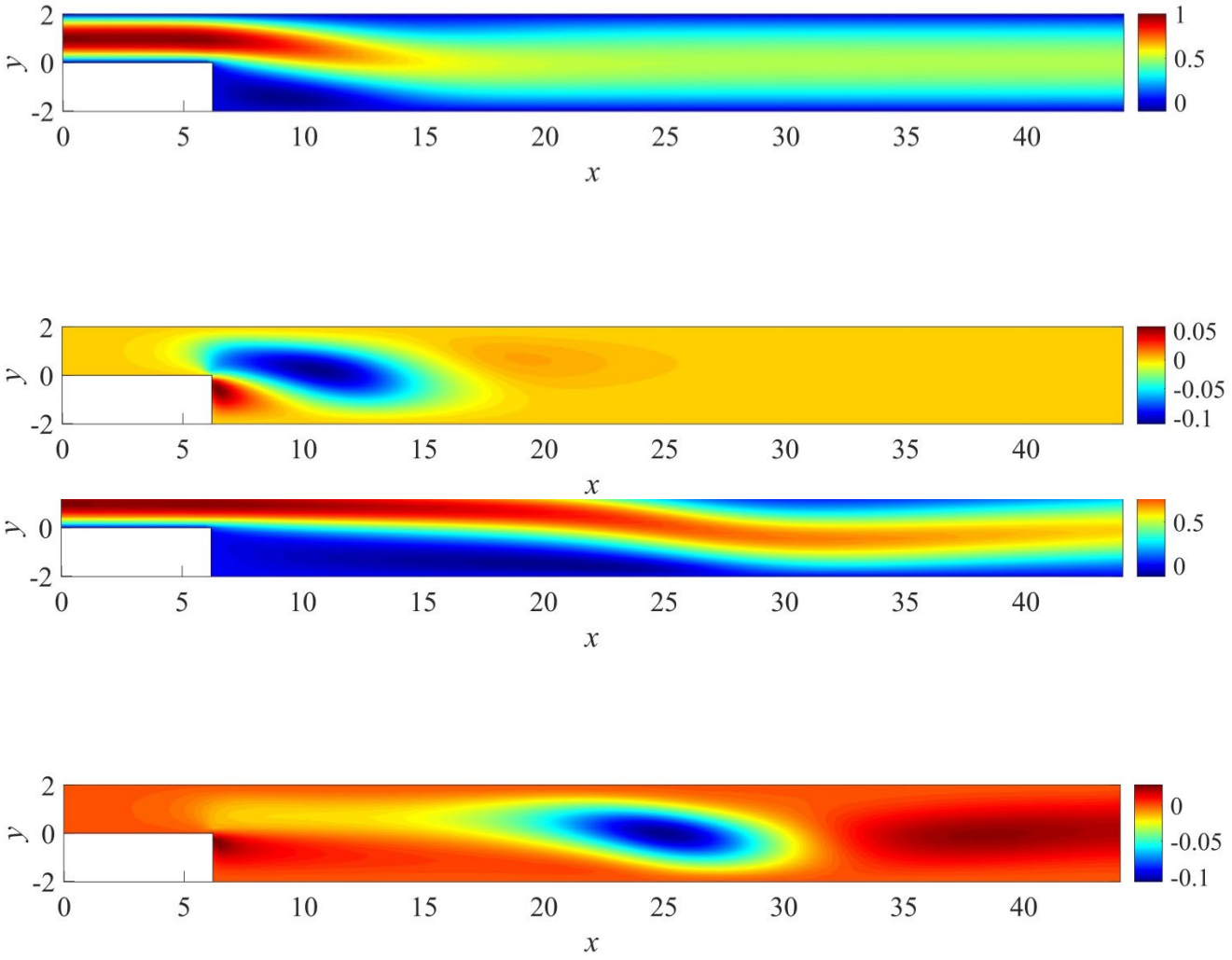


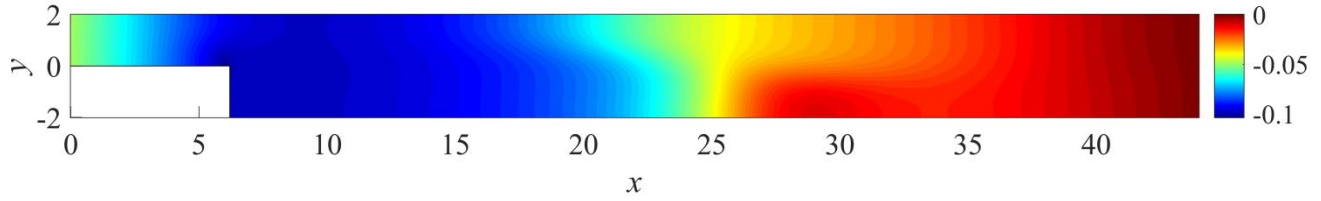
Figure 3.28. Navier-Stokes fluid flow in a stepped channel. The left boundary is subjected to the inlet velocity shown, the right side is subjected to the outlet boundary condition, and the

top and bottom boundaries are subjected to the no-slip condition. The dotted line at x = 7 is used in Fig. 3. 31. Fig 3.29 shows the calculated velocities.



(b) Horizontal velocity (top) and vertical velocity (bottom) for Re =250

Figure 3.29 Predicted fluid velocity components for the two flow conditions.



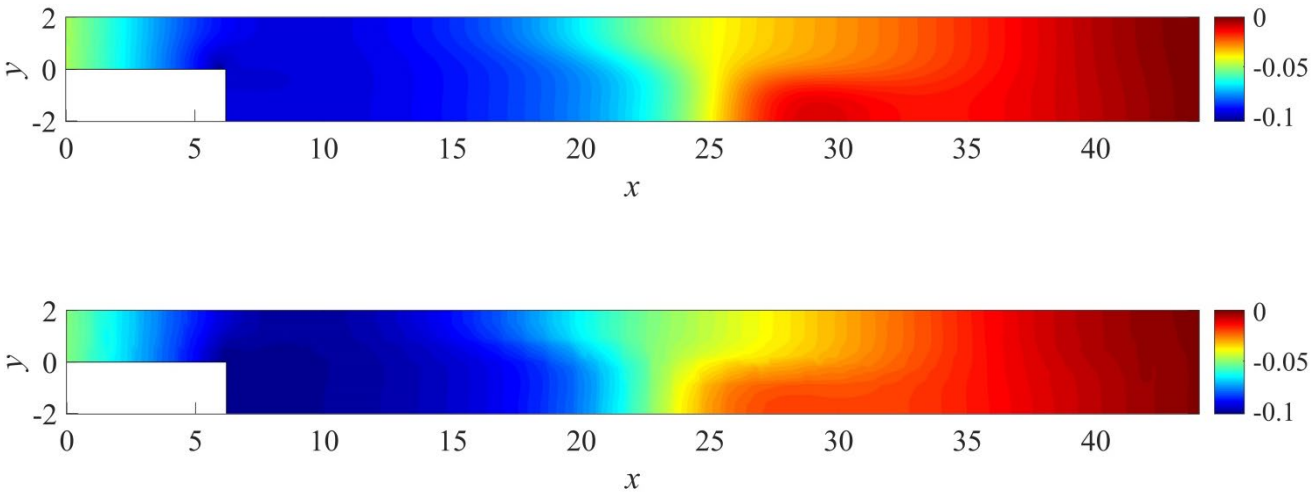


Figure 3.30 Predicted pressure at Re = 250; top: reference solution using the Q_2 / Q_1 element; middle: solution using the OFE_1 / OFE_0 element; bottom: solution using the OFE_1 / Q_0 element.

Figure 3.30 focuses on the predicted pressure. In each case the same mesh of 5,579 elements is used but with the Q_2 / Q_1, OFE_1 / Q_0 and OFE_1 / OFE_0 element assumptions.

The above figures show that the calculated velocities and pressures are very close to each other. In more detail, Fig. 3.31 gives a comparison of the predicted horizontal velocity along the vertical line x = 7 shown in Fig. 3. 28, and Fig. 3.32 shows the predicted pressure along the

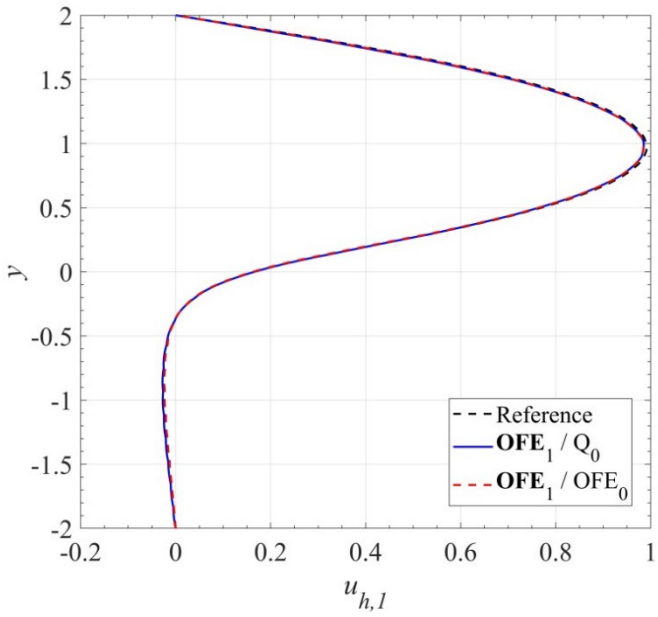


Figure 3.31 Predicted horizontal velocity along the line x = 7, see Fig. 3.28.

top wall of the channel. We see the close correspondence between the solutions but also see that, as expected, using the OFE_1 / OFE_0 element gives a more accurate prediction than using the OFE_1 / Q_0 element. Fig. 3.32 shows that the constant pressure assumed over each OFE_1 / Q_0 element results into a pressure staircase approximating a continuous line.

Further analysis results regarding this solution and more solutions using the overlapping elements are given in W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R.C. Mesquita [D].

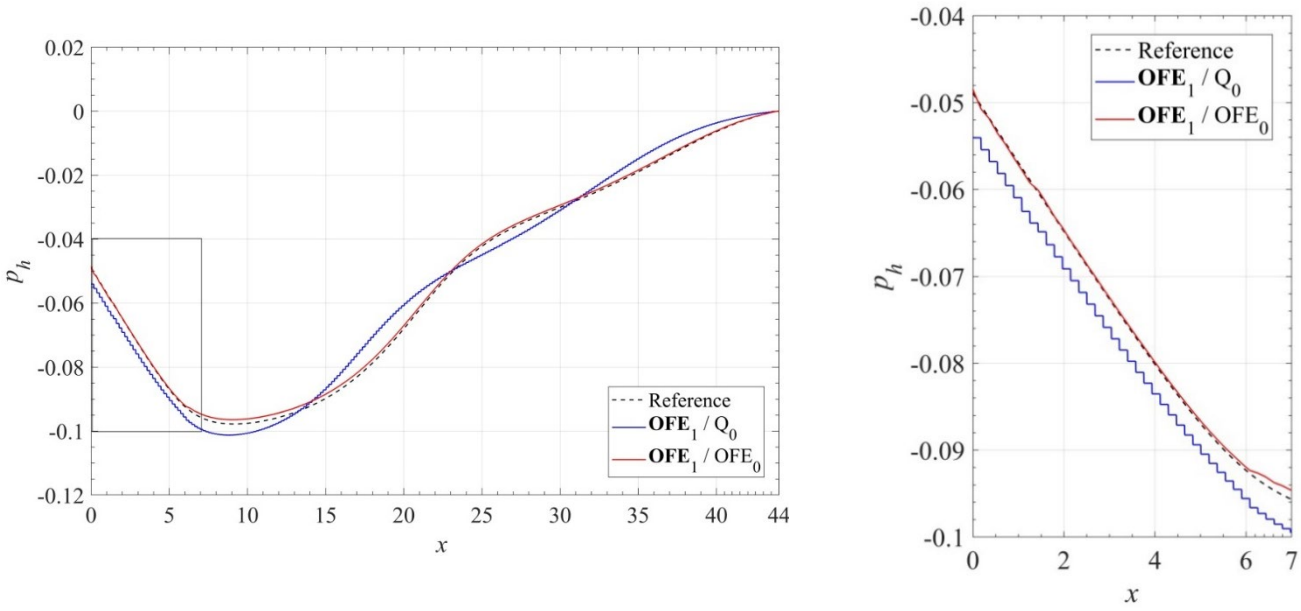


Fig. 3.32 Predicted pressure along the top wall of the channel; on the right, the solutions in the box indicated in the left figure are enlarged.

3.5 Overlapping meshes

In addition to using “overlapping finite elements”, we can also use “overlapping finite element meshes”. The objective is always, in essence, to achieve a better overall mesh quality with less computational effort.

3.5.1 The general use of overlapping meshes

In fluid mechanics, Chimera grids, also called composite overlapping grids or overset grids, have been used for some time with finite difference methods and finite volume methods, see for example, G.

Chesshire and W.D. Henshaw [A], J.A. Benek, P.G. Buning and J.L. Steger [A], and J.L. Steger and J.A. Benek [A]. This approach makes it possible to discretize local regions, like complicated boundary regions, with a fine mesh and use a coarser discretization in other regions. Similar procedures in finite element analysis require compatibility of finite element approximations and hence present additional difficulties, see J. Huang and K.J. Bathe [B, C].

Figure 3.33 illustrates the use of two overlapping finite element meshes, an interior mesh and a boundary mesh, each of the meshes consisting of 4-node elements.

The “interior mesh” consists of good quality square two-dimensional elements and the “boundary mesh” consists of somewhat distorted 4-node elements, but the distortions are not so large that the predictive capabilities of these elements are significantly affected. The green area shows where elements overlap, that is, if any node of an element of one mesh lies in the other mesh, we have an overlap of the meshes. Of course, the individual finite elements in these meshes can be traditional elements or overlapping elements.

The objective in the use of overlapping meshes is, as with the use of overlapping elements, to avoid the difficulties of meshing and only use geometrically non-distorted elements or close thereto. These elements perform best in displacement and stress predictions. By using individual meshes that together, with geometric overlapping, span over the complete analysis domain, the aim to only obtain non-distorted elements is much easier to fulfil.

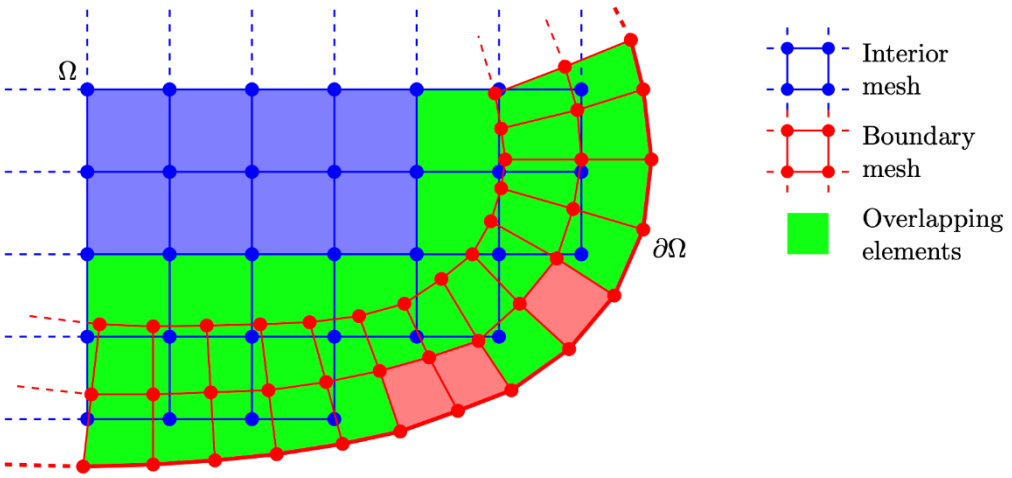


Figure 3.33 Illustration of two overlapping meshes. Regarding Figs. 3.33 to 3.38 and a further discussion, see J. Huang and K.J. Bathe [B].

A very attractive feature is that the individual meshes can be generated separately. Hence the complete process lends itself to parallel processing and to generating separately meshes in different engineering groups.

There are some key questions that need to be addressed in the development of a scheme to establish overlapping meshes, these are:

- How is compatibility between the meshes achieved ?
- How are the material properties of the structure assigned to the different meshes?
- How are the loads and the boundary conditions applied?
- And finally, what convergence and what rate of convergence are achieved when different elements are used in the different meshes?

In the following we consider a specific scheme to achieve and illustrate a solution procedure.

3.5.2 A scheme of overlapping finite element meshes

Let us assume that we want to solve for a continuous scalar approximation $u(\boldsymbol{x})$ of some exact solutionn, like temperature or displacement. One way to proceed in the analysis using overlapping meshes is to use for the approximation, see J Huang and KJ Bathe [B],

$$u = \sum_i w_i(\boldsymbol{x}) u_i(\boldsymbol{x}) \qquad (3.17)$$

where $w_i(\boldsymbol{x})$ is a continuous, non-negative weight function for mesh i and $u_i(\boldsymbol{x})$ is the variable to be solved for using that mesh. To obtain a physically acceptable solution we must have for all points on the analysis domain

$$\sum_i w_i(\boldsymbol{x}) = 1 \qquad (3.18)$$

where we have a choice in establishing the w_i as long as (3.18) is satisfied. Of course, we also want each element outside the regions of mesh overlays to be unaffected by the mesh overlays since that element does not overlap with any other mesh. Hence Eq. (3.17) is not used for elements that do not overlap.

With Eq. (3.17) given, we next interpolate each $u_i(\boldsymbol{x})$ as usual, with its element interpolation functions and nodal degrees of freedom, and assemble these functions with the multiplier w_i for each mesh i in the displacement interpolation matrix **H**. Hence the columns in that matrix list the element interpolations for mesh i weighted with w_i . These columns are multiplied by the vectors of nodal displacements pertaining to the respective meshes. The standard differentiations yield the strain-displacement matrices which are assembled in the matrix **B**. The calculation of the stiffness and mass matrices then proceeds as usual, with the constitutive relation and mass density. But we need to note that these matrices correspond to all nodal degrees of freedom of the meshes used in the overlapping of the meshes.

These steps are summarized in the following equations where we sum the contributions from m overlapping meshes on an area (or volume)

$$u(\boldsymbol{x}) = \sum_{i=1}^{m} w_i(\mathbf{x})\, u_i(\mathbf{x}) = \sum_{i=1}^{m} w_i(\mathbf{x})\mathbf{H}_i(\mathbf{x})\mathbf{q}_i \qquad (3.19)$$

and the $\mathbf{H}_i$ and $\mathbf{q}_i$ are the displacement interpolation matrix and the nodal displacements of mesh *i*. To obtain the approximations of the strains $\boldsymbol{\varepsilon}(\mathbf{x})$, we use

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \sum_{i=1}^{m} \mathbf{B}_i(\mathbf{x})\mathbf{q}_i = [\mathbf{B}_1 \quad \dots \quad \mathbf{B}_m] \begin{Bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_m \end{Bmatrix} \qquad (3.20)$$

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} [w_i(\mathbf{x})\mathbf{H}_i(\mathbf{x})] \qquad (3.21)$$

where $\mathbf{B}_i$ is the strain-displacement matrix of mesh *i.* We note that the weight function is included in the differentiation. The stiffness and mass matrices are then calculated in the usual way, for example

$$\mathbf{K} = \int_{\Omega} [\mathbf{B}_1 \quad \dots \quad \mathbf{B}_m]^{\mathrm{T}} \mathbf{C} [\mathbf{B}_1 \quad \dots \quad \mathbf{B}_m] d\Omega \qquad (3.22)$$

where $\mathbf{K}$ is the stiffness matrix of the *m* overlapping meshes on the area (or volume) Ω.

Since the displacement interpolations of the elements that overlap in the meshes are affected by the weight functions and so are the derivatives to obtain the strain-displacement interpolations, it is preferable to use simple weight functions. A reasonable choice are piecewise linear functions, then the numerical integration effort to calculate the element stiffness matrices is not large.

We can now address the questions posed above. Compatibility between meshes is satisfied due to the interpolation used, the interpolation process does not involve constitutive relations and hence is applicable

for all situations of material properties, and the loads and boundary conditions are simply applied as usual using the new interpolations established in Eq. (3.19).

More details of the above scheme, including a discussion of how to construct the weight functions, issues regarding the solvability, stability, convergence and the implementation of the method are presented in J. Huang and K.J. Bathe [C].

The rate of convergence is naturally affected by the interpolations used in each mesh

$$E - E_h \leq C \sum_i h_i^{2p_i} \tag{3.23}$$

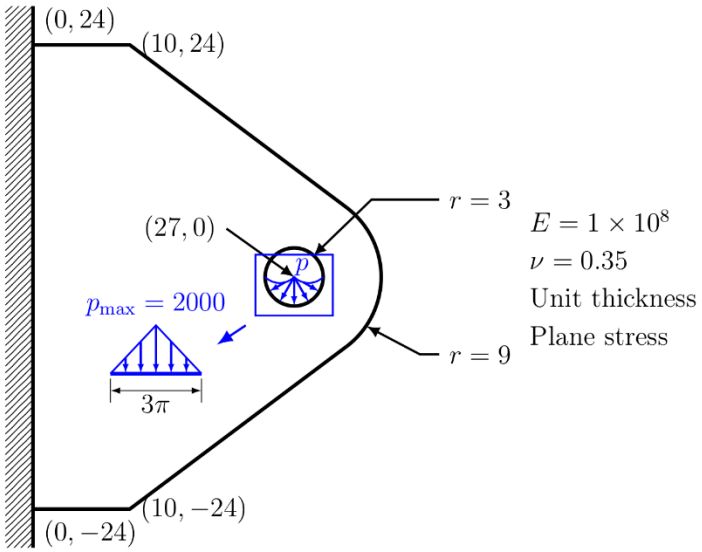
where E and E_h are the strain energies of the exact and numerical solutions, C is a constant independent of the element sizes, but dependent on the material properties, h_i is the typical element size of mesh i, p_i is the order of the approximation used in the elements of mesh i, and we sum over all meshes that are overlapping. This result is very similar and reduces to the result obtained for a single mesh used, see KJ Bathe [A] and the relation (3.6), but of course now holds for multiple overlapping meshes.

3.5.3 Illustrative solutions using overlapping meshes

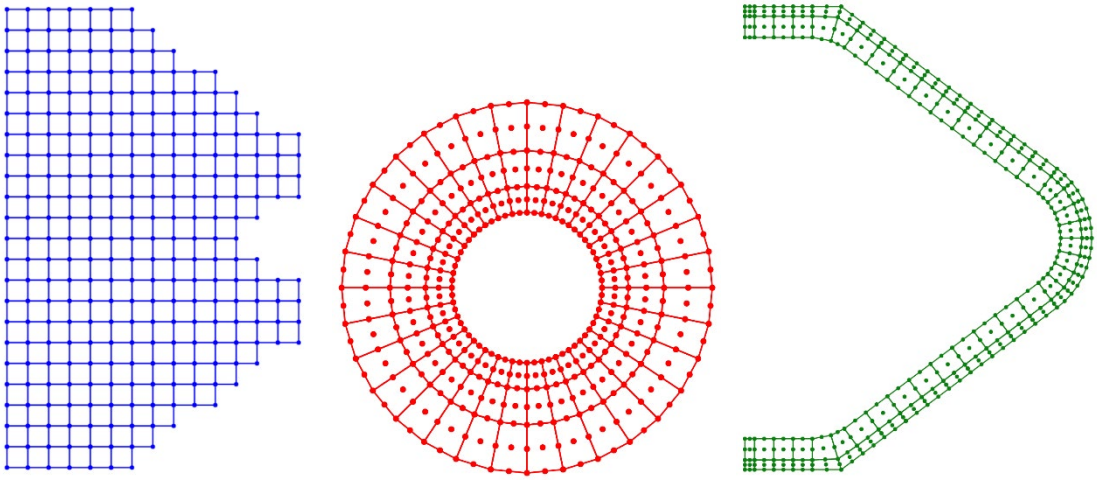
We give in this section two solutions in which overlapping meshes are used with the scheme described above, for more details see J. Huang and K.J. Bathe [B, C].

A bracket with a hole

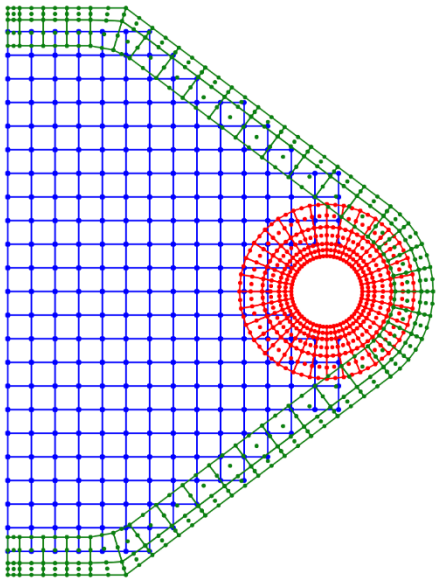
A bracket is fixed on the left boundary and loaded on its hole, as shown in Fig. 3.34. Three overlapping meshes are used to discretize the analysis domain. For the interior domain 4-node elements with incompatible modes are used, and for the boundary domains 9-node finite elements are employed.



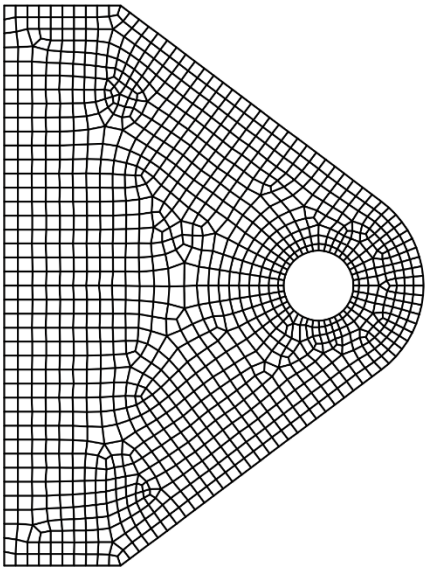
(a) Problem description



(b) Three meshes used (not to scale); the mesh shown on the left uses 4-node elements with incompatible modes



(c) Overlapping finite element meshes



(d) The mesh of 4-node elements with incompatible modes

Figure 3.34 A bracket with a hole

A mesh of 4-node elements with incompatible modes is also used to obtain a traditional solution, see Fig. 3.34 (d). The reference solution has been obtained using a very fine mesh of 9-node elements. The numerical results of this problem are given in Table 3.6 and Figure 3.35. The use of overlapping meshes gives in this analysis good displacement and stress predictions.

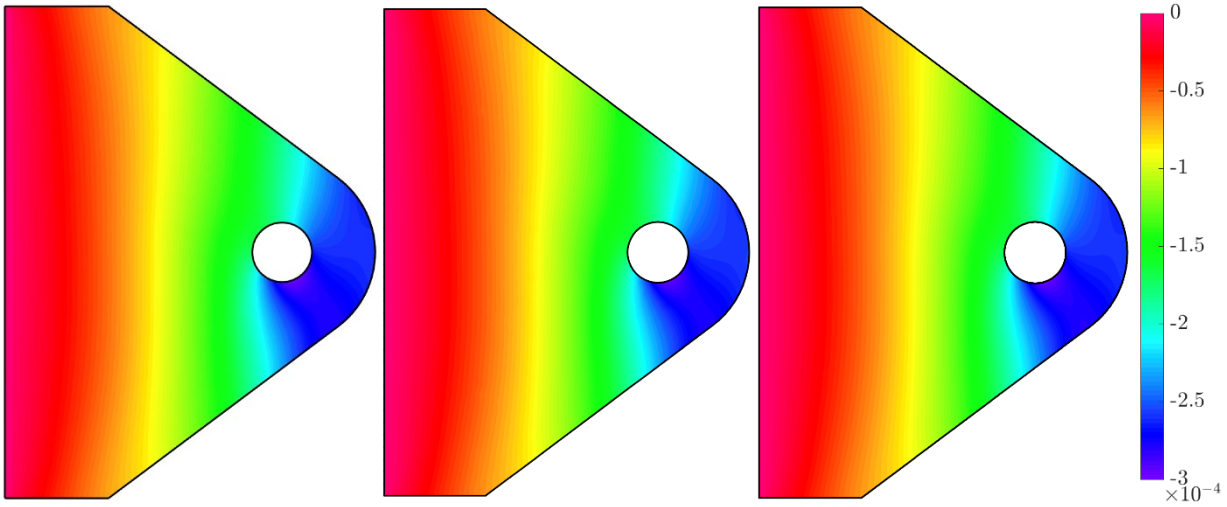
We notice that in this case, the number of non-zero sparse matrix entries is much larger using the overlapping meshes than in the traditional finite element analysis, although the number of degrees of freedom is less. However, while there is the increase in solution time, the effort expended on meshing is much less.

Table 3.6 Numerical solutions for the bracket with a hole

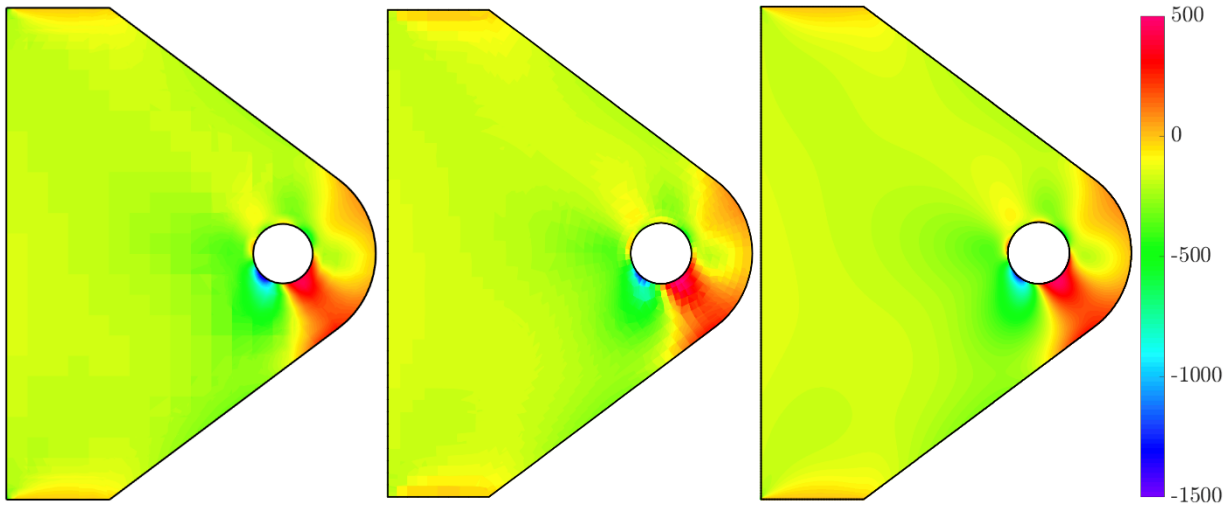
	Overlapping meshes	Traditional solution	Reference
Energy	1.0617	1.0551	1.0630
$v_{\min}$	-0.29732E-3	-0.29597E-3	-0.29777E-3
$\tau_{xy\min}$	-1.4079E3	-1.3531E3	-1.4560E3
$\bar{\tau}_{\max}$	2.5129E3	2.5560E3	2.5732E3
Number of dofs	2,156	2,480	>166,000
NNZ*	94,750	22,516	>2,722,000

*NNZ: Number of non-zero sparse matrix entries

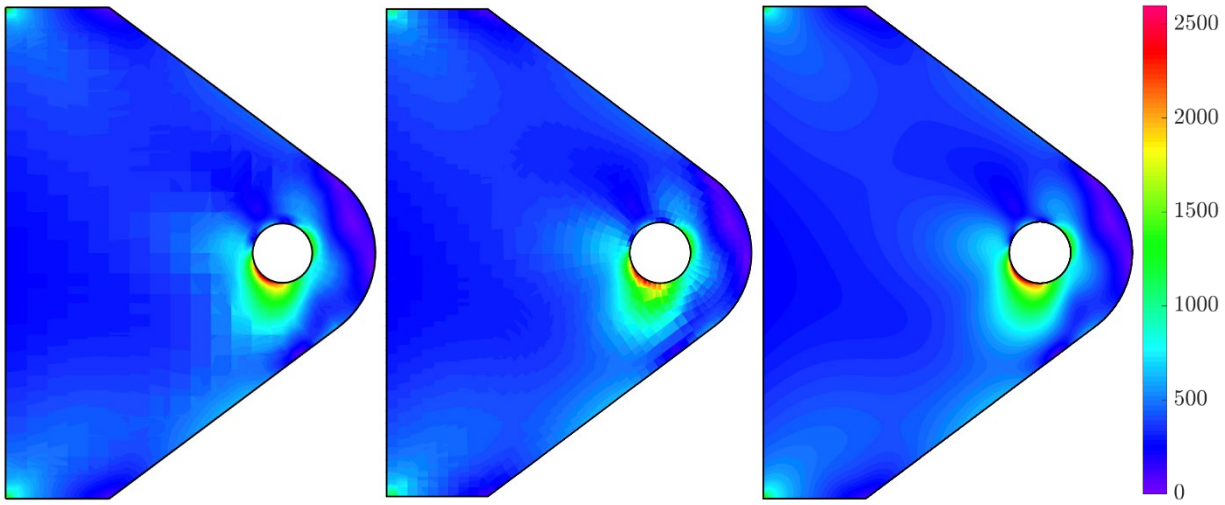
Number of dofs: Number of degrees of freedom



(a) Solutions of vertical displacement



(b) Solutions of shear stress τ_{xy}

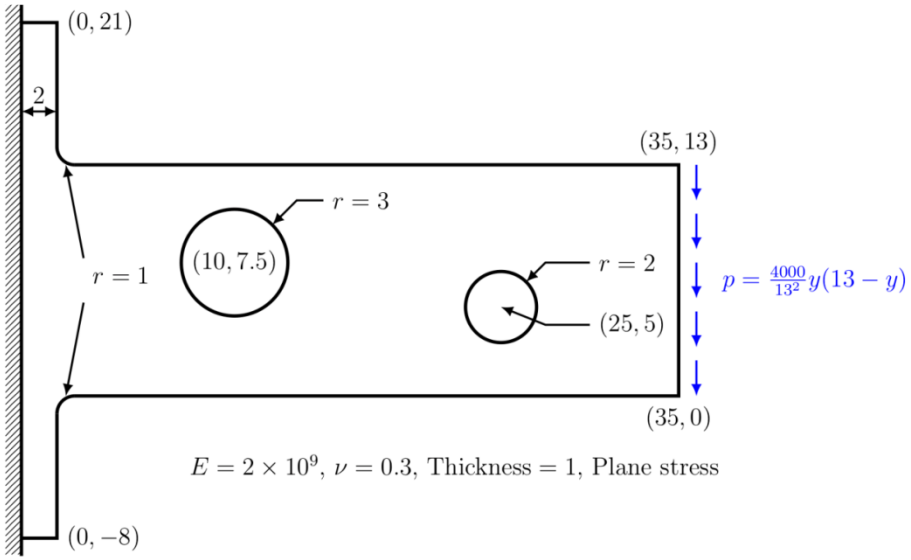


(c) Solutions of effective stress $\bar{\tau}$

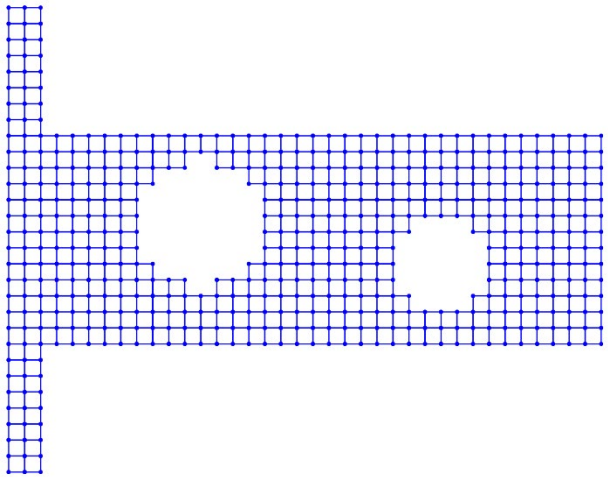
Figure 3.35 Numerical solutions for the bracket with a hole; using overlapping meshes (left), using the mesh of Fig. 3.34 (d) (middle), with the reference solution on the right

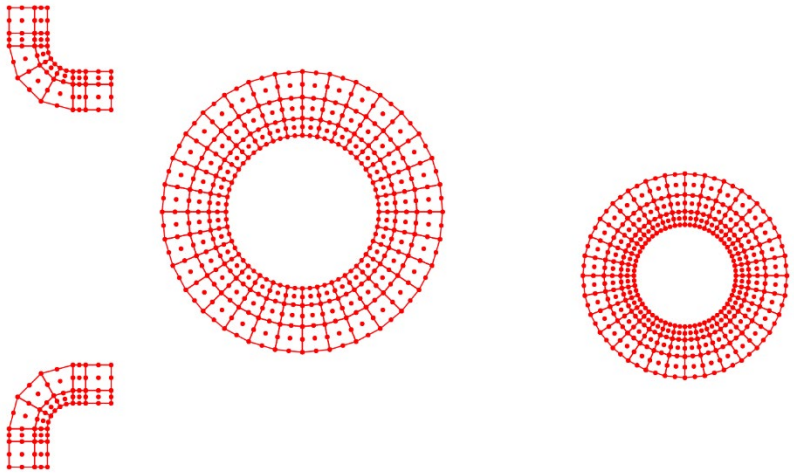
Analysis of a plate with two holes

In this analysis, we consider the solution of the plate problem with two holes given in Fig. 3.36.

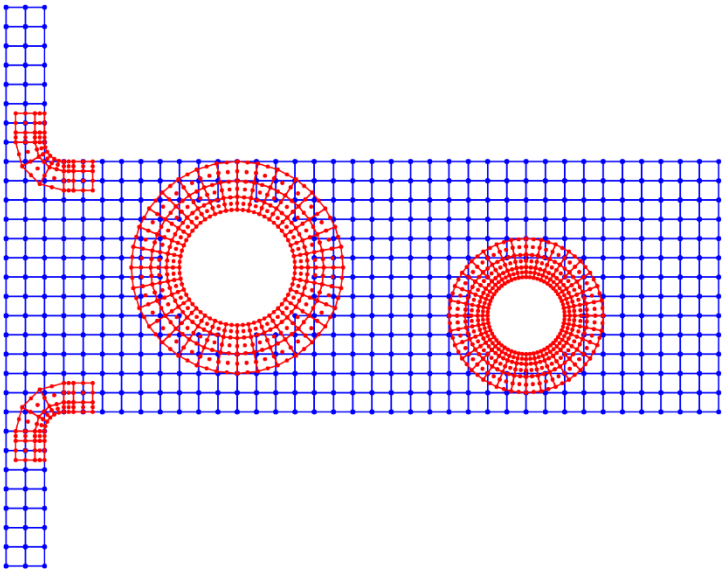


(a) Problem description





(b) The individual meshes used (not to scale)



(c) Assembled meshes

Figure 3.36 A plane stress plate problem and the overlapping meshes

A 4-node element mesh is spanned over most of the domain and 9-node element meshes are established in the domains near the curved boundaries.

We compare the numerical solutions in Table 3.7 and Fig. 3.38 with the result using the traditional 9-node finite element mesh shown in Fig. 3.37 and the reference solution obtained using a very fine mesh of 9-node elements. We see that good results have been obtained using the overlapping meshes, and, comparing with the traditional finite element analysis, in this case the number of degrees of freedom of the overlapping solution is smaller and the number of nonzero entries in the stiffness matrix is about the same.

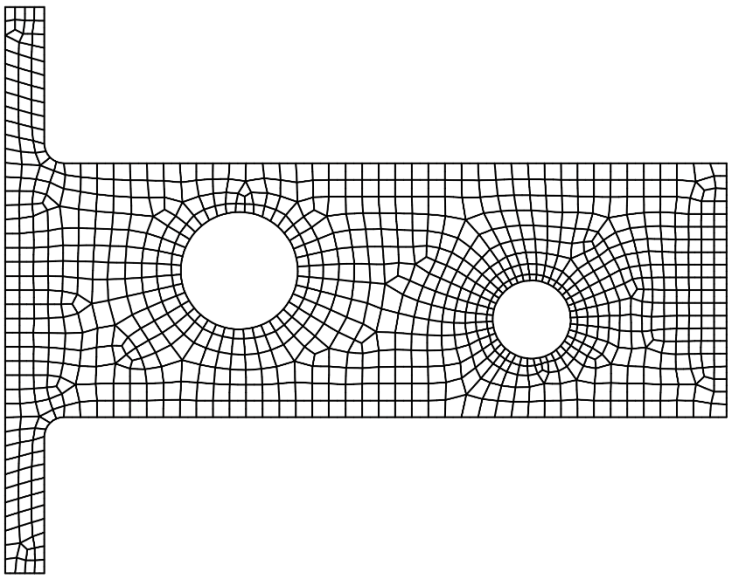


Figure 3.37 A traditional 9-node finite element mesh

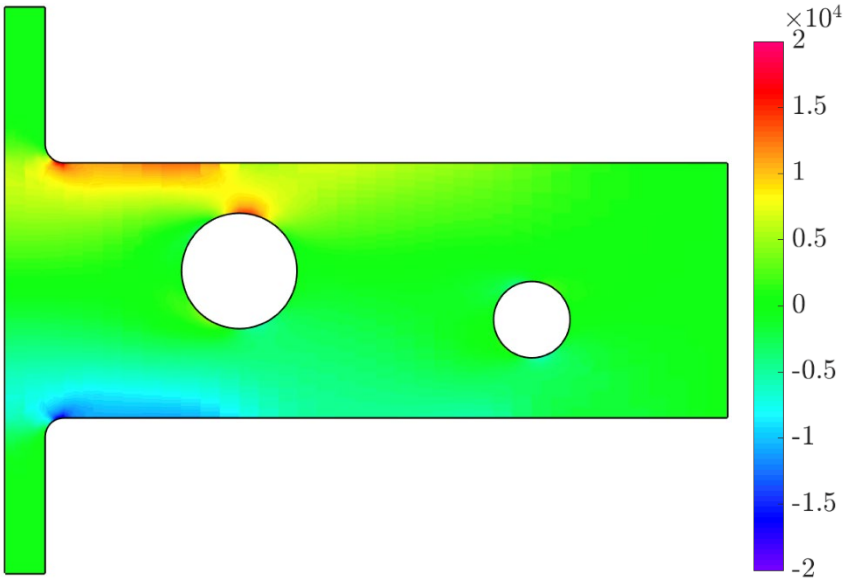
Table 3.7 Numerical solutions for the plate problem with two holes

	Overlapping meshes	Traditional finite element solution	Reference
Energy	1.9922	1.9870	1.9962
$u_{\max}$	0.11332E-3	0.11325E-3	0.11352E-3
$u_{\min}$	-0.10364E-3	-0.10353E-3	-0.10377E-3
$\tau_{xx\max}$	18.205E3	18.017E3	19.135E3

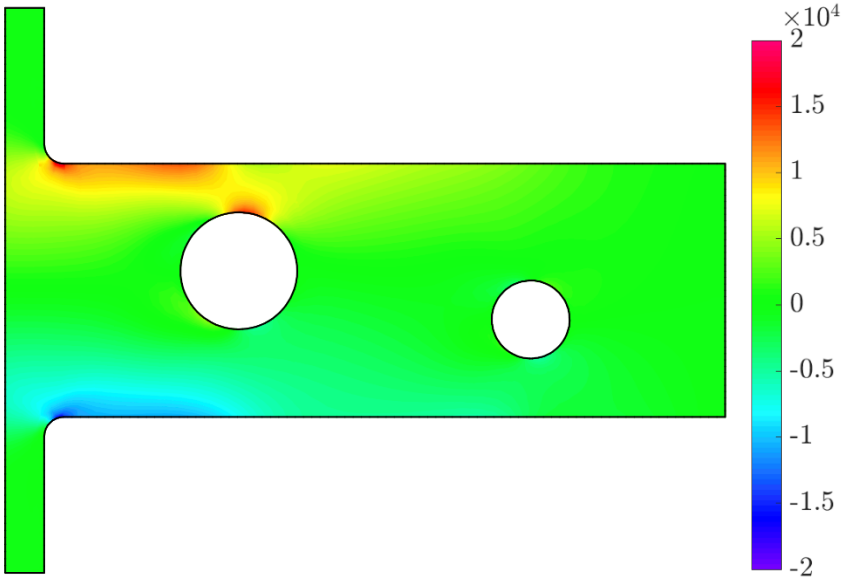
$\tau_{xx\min}$	-18.295E3	-18.013E3	-19.181E3
$\bar{\tau}_{\max}$	19.135E3	18.935E3	19.673E3
Number of dofs	3,070	7,724	>98,000
NNZ*	120,769	120,690	>1,602,000

*NNZ: Number of non-zero sparse matrix entries

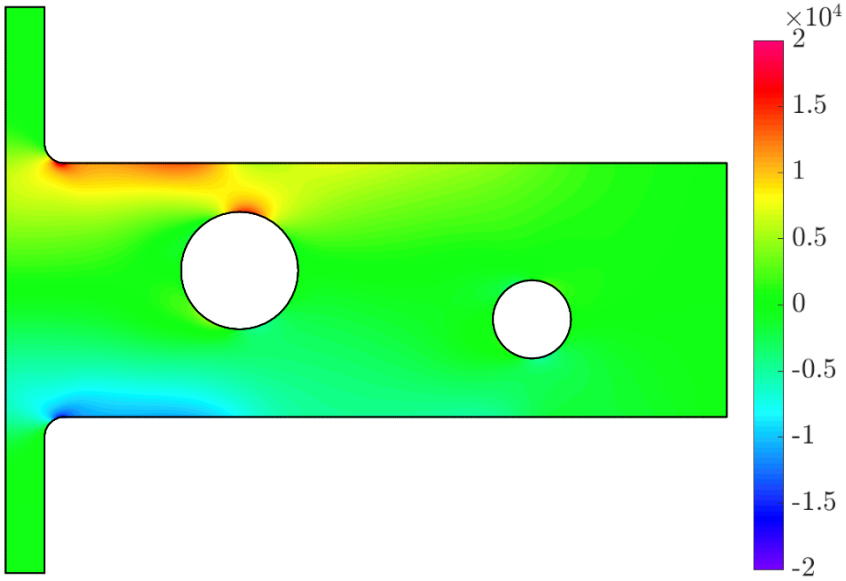
Number of dofs: Number of degrees of freedom



Solution using overlapping meshes

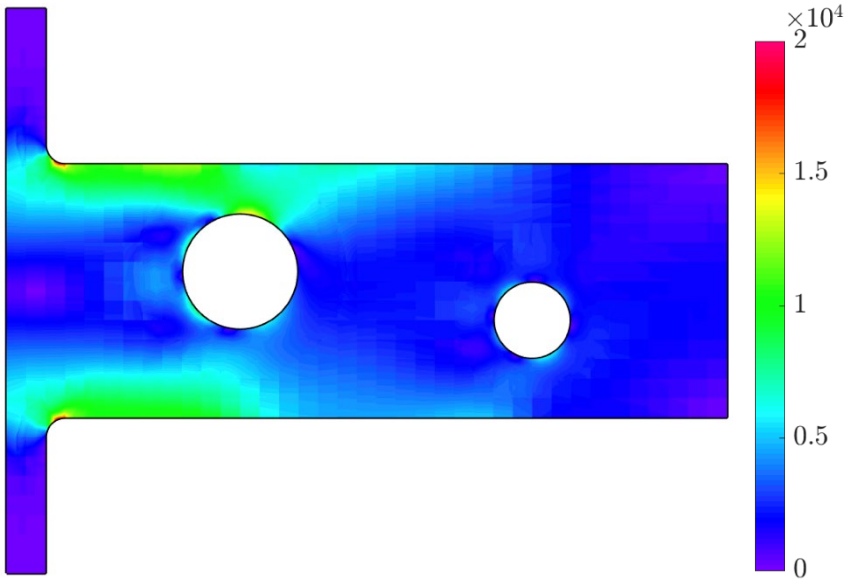


Traditional finite element solution

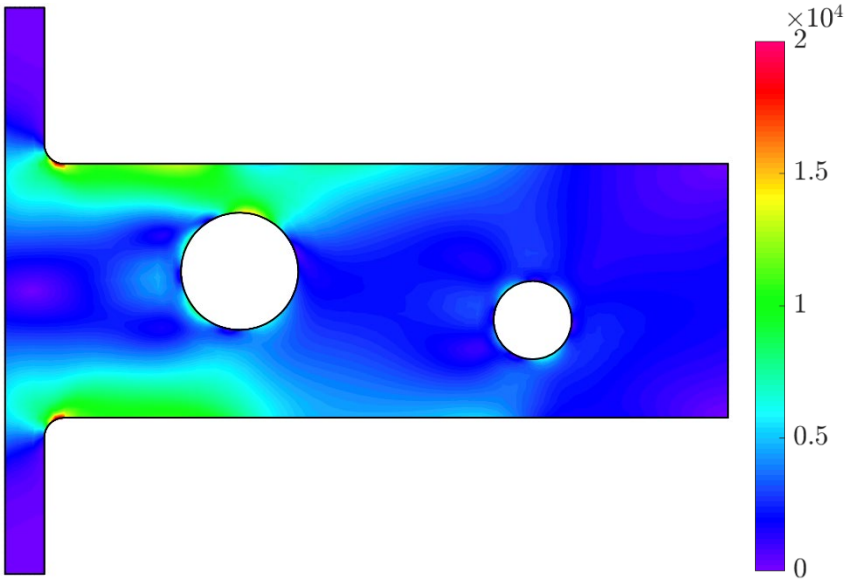


Reference solution

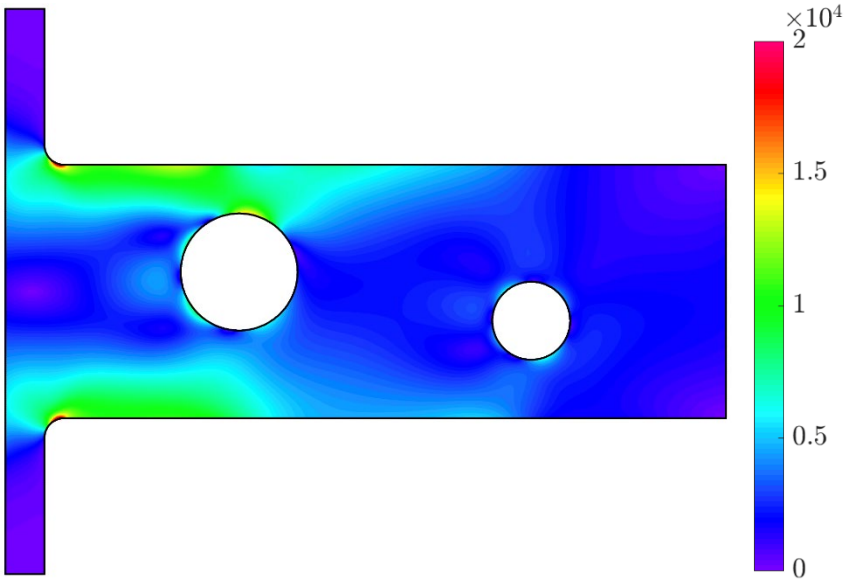
(a) Solutions of the normal stress τ_{xx}



Solution using overlapping meshes



Traditional finite element solution



Reference solution

(b) Solutions of the effective stress $\bar{\tau}$

Figure 3.38 Numerical solutions for the plate problem with two holes

3.6 The AMORE(FN2) procedure

As discussed above, the overlapping finite elements have been developed to establish elements that are quite insensitive to geometric distortions. Then these elements can be used in areas and volumes of a domain where meshing with (almost) undistorted elements is difficult or even impossible to achieve.

A computationally attractive procedure based on the use of overlapping elements to reduce the effort of meshing and obtain good solution results is given by the AMORE scheme, with “AMORE” an acronym for “Automatic Meshing with Overlapping and Regular Elements”. That is, in the scheme we use regular, traditional, finite elements and overlapping finite elements, always drawing on their best computational capabilities, see K.J. Bathe [C, D].

(FN2) While we use AMORE as an acronym, in Italian, “AMORE” means “LOVE”

3.6.1 The analysis steps using AMORE

The objective is to use the traditional *geometrically undistorted* finite elements as much as possible, because then these regular elements have the best predictive capability, and use overlapping finite elements when the elements must be distorted in the meshing to conform to the given geometry. The AMORE scheme is given by the following algorithmic steps, K.J. Bathe [D]:

Step 1: A Cartesian mesh of cells is spanned over the geometry to be analyzed; hence the geometry is immersed in a Cartesian mesh of regular (undistorted) cells. The geometry could be the output from any CAD program or been obtained by any other means.

Step 2: The boundary of the geometry is meshed. In two-dimensional analyses, the boundary lines are meshed using two-node straight lines, and in three-dimensional analyses, the boundary surfaces are meshed using 3-node triangular flat facets. Since these facets can be quite distorted, this meshing is relatively easy to achieve.

Step 3: All cells of the Cartesian mesh spanned in step 1 that lie outside the discretized boundary of the geometry or cut that boundary are removed. This process results into "empty regions". The remaining cells are converted into low-order regular finite elements, that is, traditional finite elements that are geometrically undistorted.

Step 4: The empty regions obtained in Step 3 are filled in with overlapping finite elements, which can be geometrically quite distorted because their predictive capabilities are not sensitive to element geometric distortions.

In engineering practice, the incompatible modes elements can be effective when they are geometrically non-distorted, that is, in two-dimensional analysis it is best to use square elements, rectangular elements

can be used too but not of long rectangular geometry, see T. Sussman and K.J. Bathe [A]. Hence these elements may be employed for the interior mesh. The regions that are empty in the geometry to be analyzed can then be meshed with any of the overlapping finite elements.

Considering the total computational effort for meshing, we can expect that the spanning of the Cartesian grid requires little effort, the establishment of the interior grid is not requiring much expense, and the meshing of the empty regions with overlapping elements can also be performed with a reasonable computational expense because these elements can be geometrically quite distorted. Therefore, we can expect that the total effort on meshing is much reduced from the way meshing is performed traditionally.

While not specifically mentioned in the above steps, we may also regard the "scheme of overlapping meshes" as an AMORE solution, see J. Huang and K.J. Bathe [B]. Here the interior mesh is obtained as described above in *Steps 1 to 3*, and instead of using overlapping elements to fill in the empty space, as described in *Step 4*, overlapping meshes are used, see Section 3.5.

A particular attractive procedure would be to use AMORE in an adaptive solution scheme. Assuming that a mesh has been established, and the analysis results have been calculated, we can visualize that based on an error measure, the solution is deemed not sufficiently accurate in some areas of the mesh where overlapping elements are used. Then simply the nodal basis could be increased at certain nodes of the overlapping elements –- without remeshing – and better accuracy could be reached. If the solution is not accurate in areas represented by traditional elements, these would first be changed to overlapping elements. Of course, this approach would require an effective error measure and an effective scheme for assembling and solving the governing equations as these equations change in the adaptive procedure. The aim in this process is to solve the underlying mathematical model more accurately, see K.J. Bathe [A]. Hence the objective is different from the aim in *hierarchical modeling* in which we proceed from a lower-order mathematical model to a higher-order mathematical model and we solve each of these models accurately, see M.L. Bucalem and K.J. Bathe [A].

3.6.2 Illustrative solutions

To exemplify the use of AMORE, we present in this section two solutions. The AMORE scheme is also available in ADINA, see Bentley Systems, Inc. [A], and K.J. Bathe [N[.

The analysis of a bracket

We consider the solution of the problem shown in Fig. 3.39. Such a bracket was also considered in Fig. 3.34.

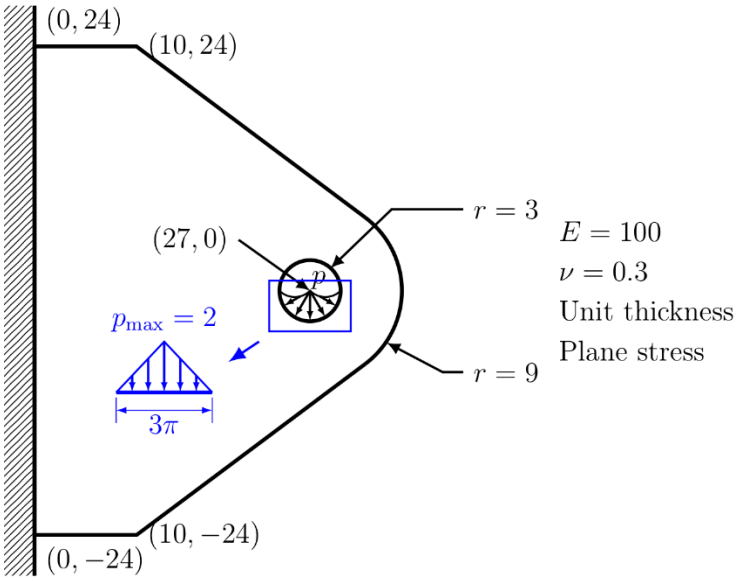
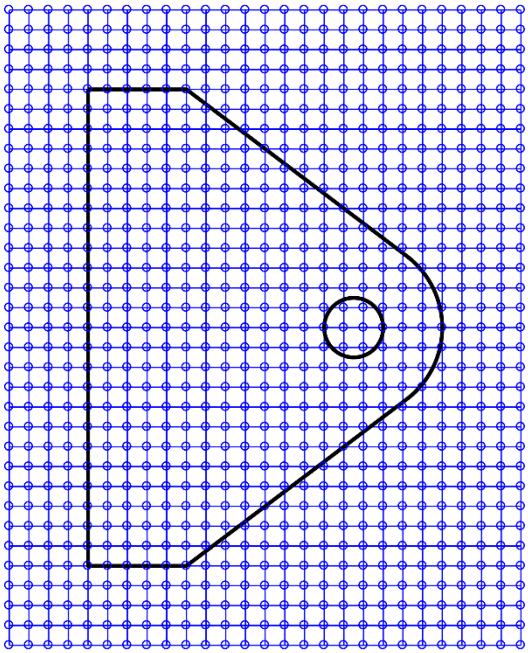
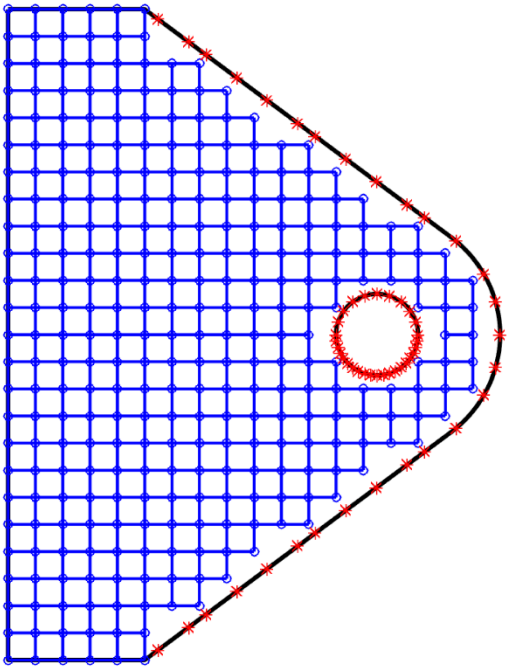


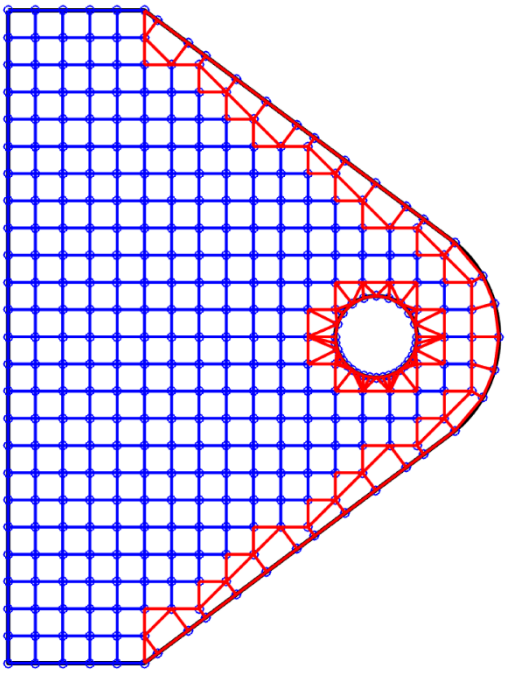
Figure 3.39 The bracket problem, K.J. Bathe [D]



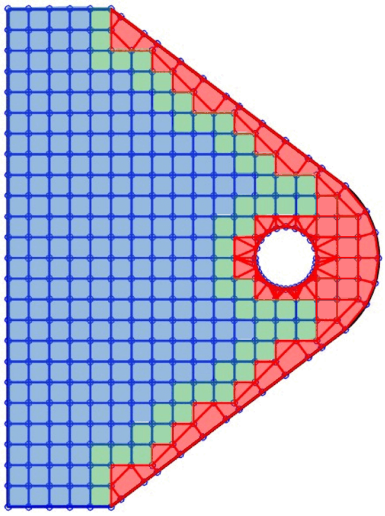
a) Step 1: The structure is immersed in a Cartesian grid



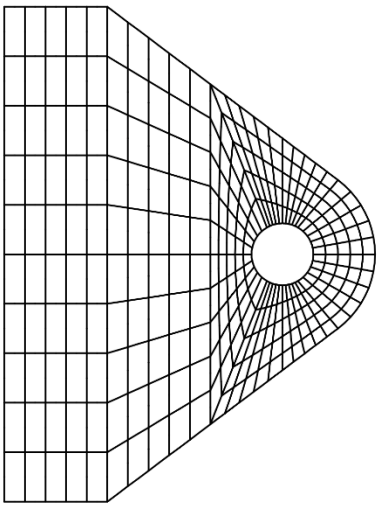
b) Step 2: The boundary is discretized; Cells outside or cutting the analysis domain are removed



c) Step 3: The empty space is meshed with overlapping elements

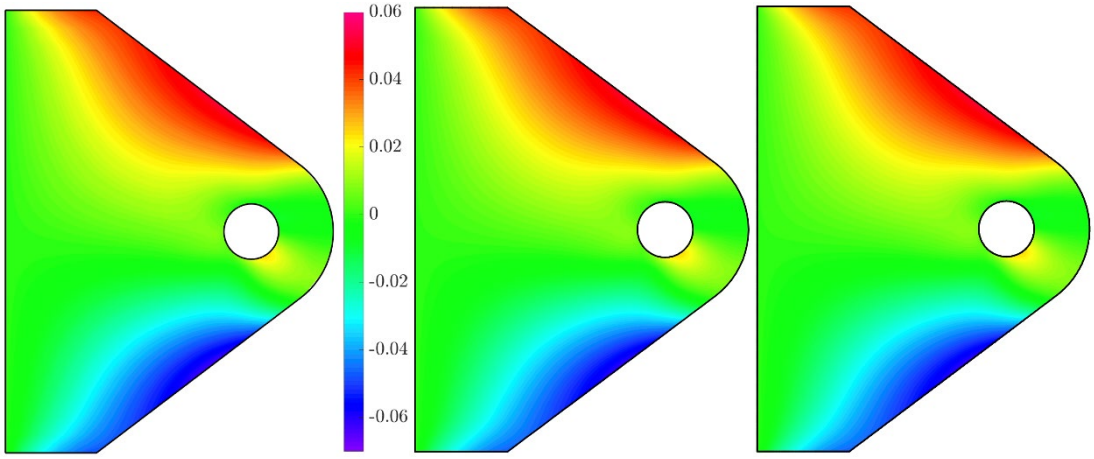


d) AMORE mesh (1,936 degrees of freedom)



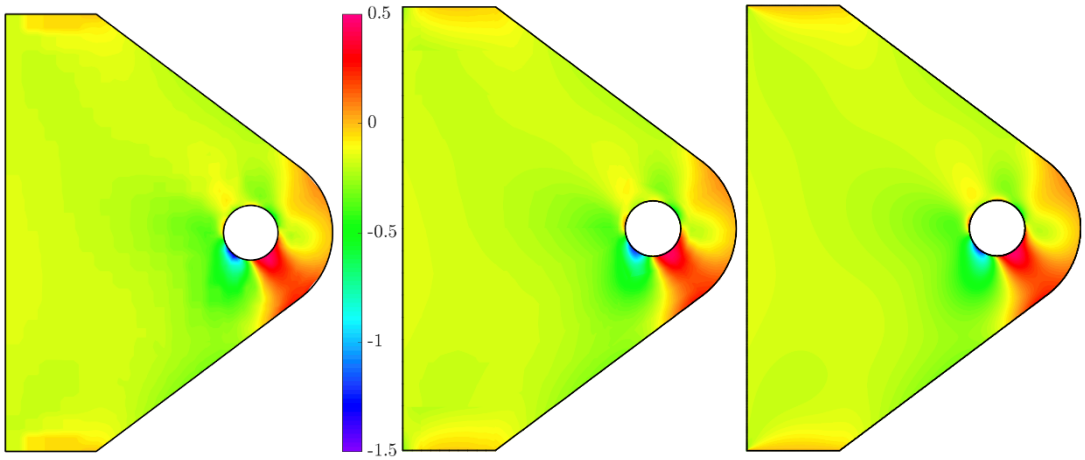
e) Traditional mesh (9-node elements; 2,558 degrees of freedom)

Figure 3.40 Meshing in the AMORE paradigm and for the traditional mesh.



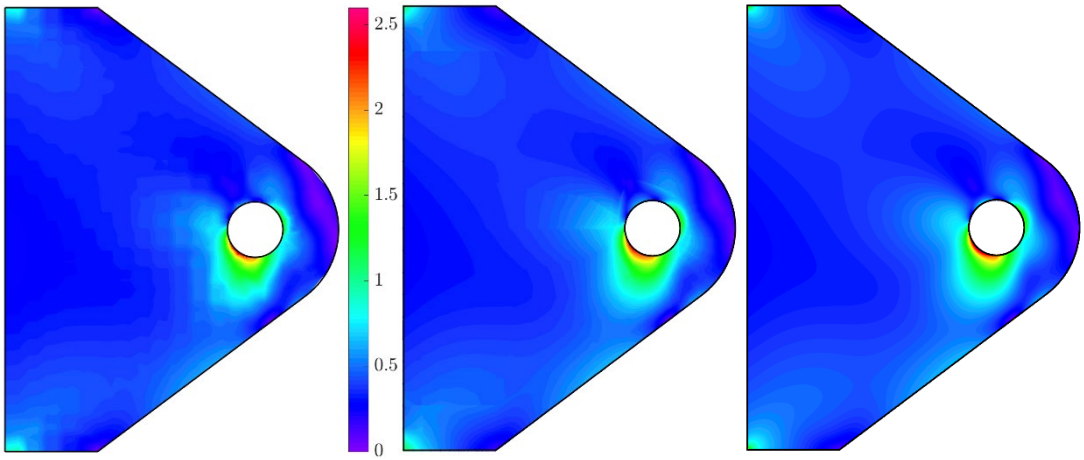
Left: $u_{\max} = 0.05224, u_{\min} = -0.06284$; middle: $u_{\max} = 0.05224, u_{\min} = -0.06298$;

right: $u_{\max} = 0.05231, u_{\min} = -0.06312$



$\sigma_{xy\max} = 0.4728, \sigma_{xy\min} = -1.4523$; $\sigma_{xy\max} = 0.4625, \sigma_{xy\min} = -1.3187$;

$$\sigma_{xy\max} = 0.4680,\ \sigma_{xy\min} = -1.4604$$



$\bar{\sigma}_{\max} = 2.6682$ $\bar{\sigma}_{\max} = 2.3358$ $\bar{\sigma}_{\max} = 2.5800$

Figure 3.41 Numerical solutions of the bracket problem, left: AMORE solution, middle: Traditional mesh solution, and right: Reference solution. The displacements, shear stresses and effective stresses are considered.

Figure 3.40 shows the steps of meshing using the AMORE scheme and Fig. 3.41 gives the solutions obtained.

The reference solutions have been established using a very fine mesh of traditional 9-node elements. We see that the results using AMORE are quite accurate although a fairly coarse mesh has been used.

Dynamic analysis of a three-dimensional tool jig

In this analysis we consider the mode superposition solution of the bending problem given in Fig. 3.42, S. Lee and K.J. Bathe [C]. A mesh of 8-node incompatible mode elements with 106,866 degrees of freedom

and an AMORE mesh with 66,120 degrees of freedom are used to solve the problem, see Fig. 3.43. Using both meshes, the relative error of the smallest natural frequency is 0.3%. The reference solution has been obtained using a fine 27-node element mesh of 867,258 degrees of freedom.

For the analysis, the ten smallest frequencies of the models are calculated of which the six smallest frequencies and associated mode shapes are employed for the mode superposition solution. The sixth frequency is about four times the excitation frequency and hence no higher modes need to be included in the solution, see K.J. Bathe [A].

In the superpositions of the modal responses, the time step used is $30\,\mu s$ which is about one twentieth of the period of the sixth mode and the static correction is applied, see KJ Bathe [A].

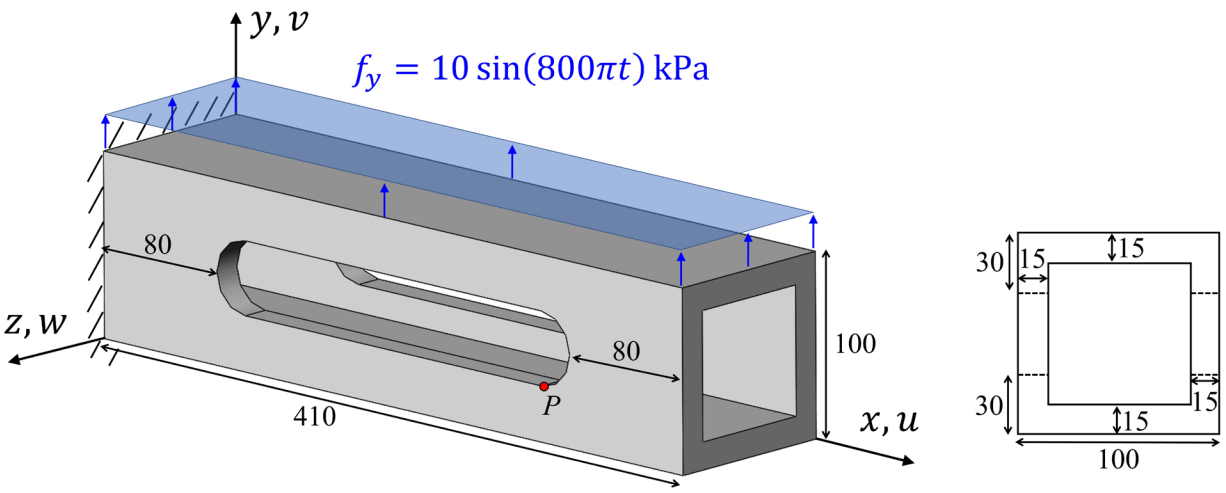


Fig. 3.42 Mode superposition analysis of a tool jig; Young's modulus = 200 GPa and Poisson's ratio = 0.3; the tool jig is subject to a sinusoidal *y*-direction surface traction; *t* denotes the time elapsed in seconds; lengths in mm, S. Lee and K.J. Bathe [C].

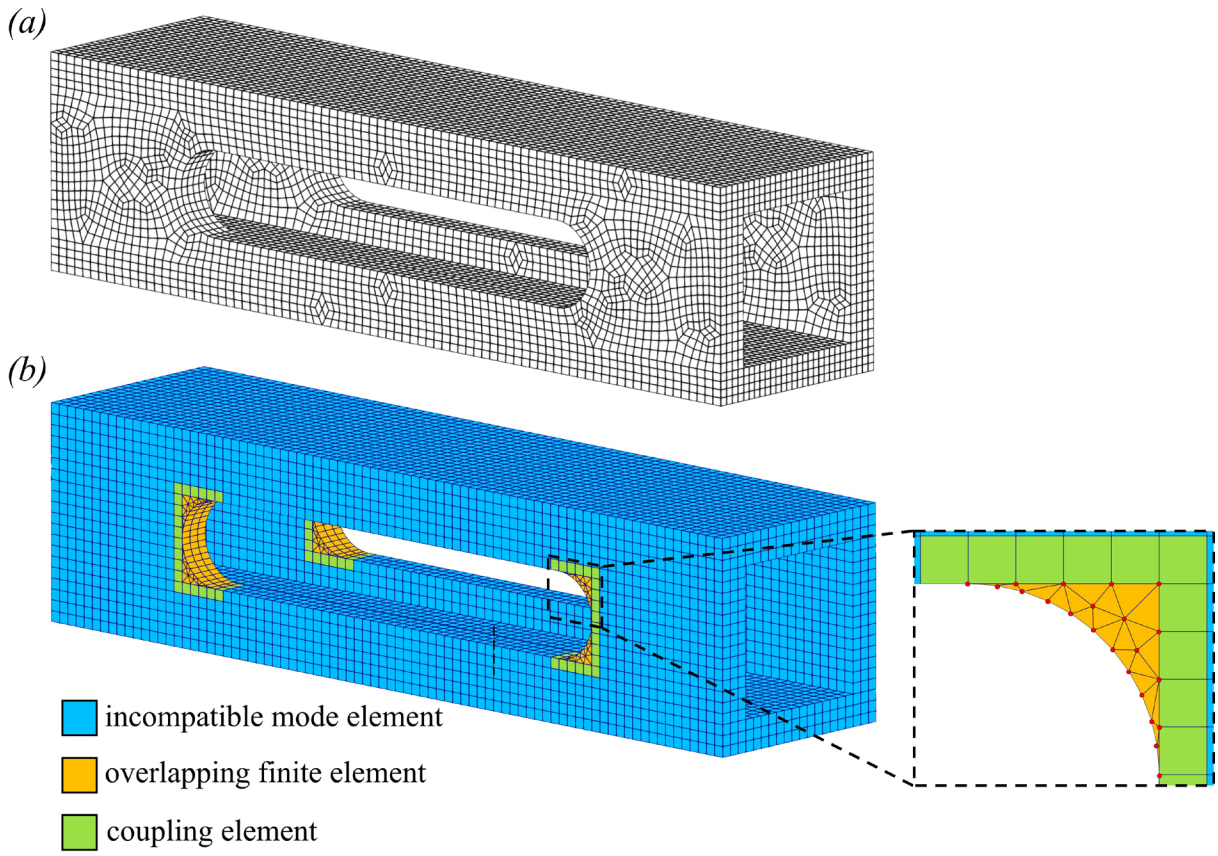


Fig. 3.43 For the tool jig; (a) The traditional mesh using the incompatible mode element; (b) The AMORE mesh using the 6-node prism overlapping element with the linear basis and $\beta = 0.01$; the 8-node coupling element is used.

We see that the use of the AMORE mesh and the traditional mesh results in similar solution accuracies for numerical predictions at point *P* (see Figs. 3.44) while the CPU time is less when using the AMORE mesh (see Table 3.8).

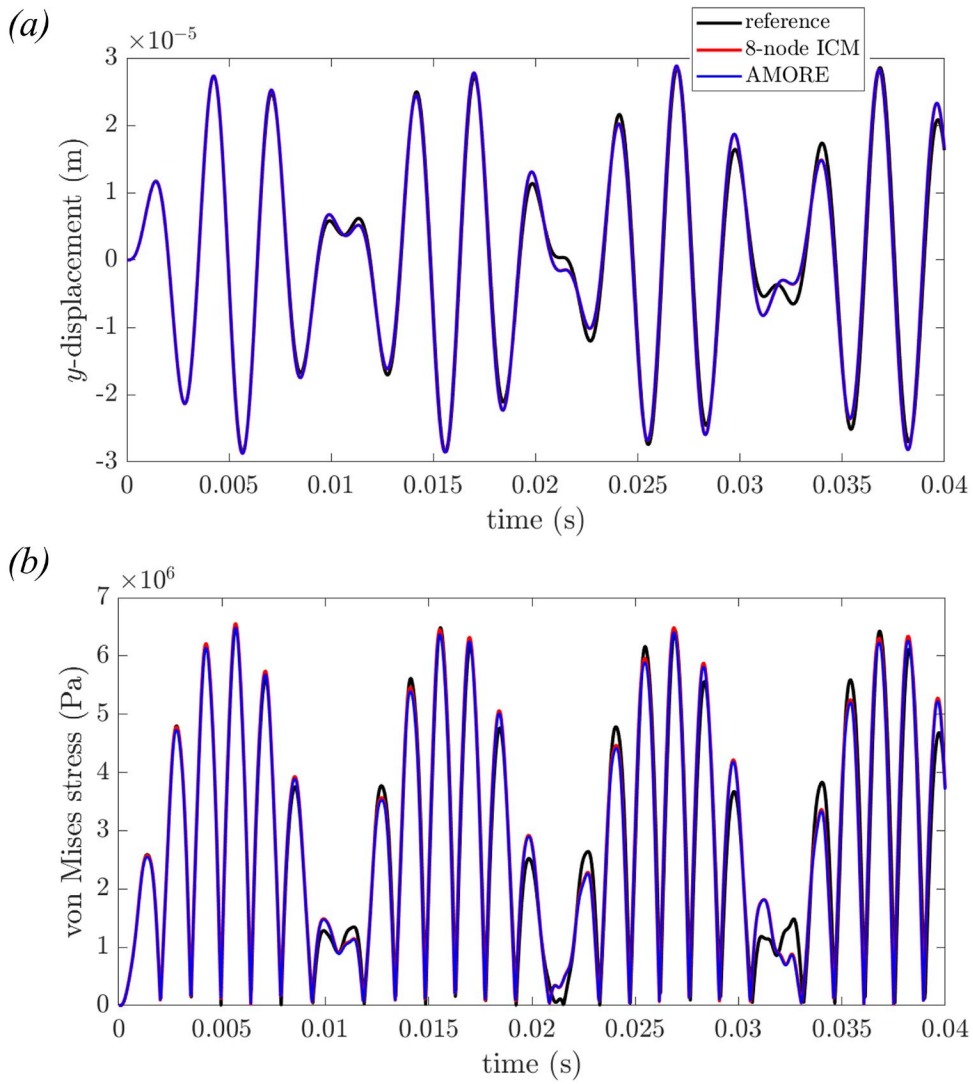


Fig. 3.44 Numerical predictions at point *P*, see Fig. 3.42; (a) the *y*-displacement; (b) the von Mises stress.

Table 3.8 Solution of tool jig for smallest 10 frequencies; all times are normalized by the actual total CPU time used for the 8-node incompatible mode element (ICM) solution; note that both meshes give the relative error of 0.3% for the first frequency.

Mesh	Numerical integration time	Time for time integration	Total CPU time	Half-bandwidth of **K**	Total degrees of freedom
AMORE	1.6×10^{-2}	4.2×10^{-3}	3.5×10^{-1}	1,995	66,120

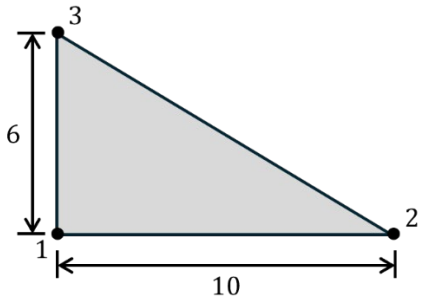
8-node ICM	2.0×10^{-2}	5.4×10^{-3}	1.0×10^{0}	2,577	106,866

3.7 Exercises

We suggest these Exercises to obtain further insight into the solution schemes discussed in this chapter. To complete an Exercise, the material given in KJ Bathe [A] may need to be known.

3.7.1

Consider the 3-node overlapping element shown below. Establish the strain-displacement matrix of the element.



Nodal basis for the degrees of freedom is $(1,\ \hat{x},\ \hat{y})$

3.7.2

Show that the coupling elements defined by Eqs. (3.1) and (3.7) provide displacement compatibility with the traditional finite elements and the overlapping elements. Also show that the equations correspond to the traditional element interpolations when there are no overlapping nodes and to the overlapping element interpolations when there are no traditional nodes.

3.7.3

Prove that the coupling elements defined by Eqs. (3.1) and (3.7) when used between overlapping elements and traditional elements satisfy the patch test.

3.7.4

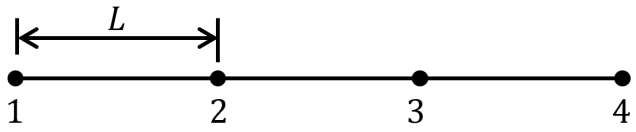
Consider Eq. (3.11) and specialize the assumption to the case of analysis of the wave propagations in a one-dimensional medium, that is, a bar.

3.7.5

Consider the 4-node overlapping element OFE_1 / Q_0 for the analysis of an incompressible medium in plane strain conditions. The element carries the 1, x, y degrees of freedom at each node and assumes a constant pressure. Evaluate the displacement interpolation functions for a square 4-node element with side lengths 2.

3.7.6

Consider the one-dimensional bar shown below. The structure is modeled using three equal-length overlapping 2-node elements with the linear polynomial as degrees of freedom. Calculate the **K** and **M** matrices of the structural model and set up the generalized eigenvalue problem.



Young's modulus = E
Cross-sectional area = A

4. Direct Time Integration Schemes

Having considered in the previous chapters improved spatial discretization schemes, we turn now to advances in procedures to solve the governing finite element equations, also including the degrees of freedom of overlapping elements.

In static analyses, these advances pertain to efficiency increases of the Gauss elimination schemes through sparse solvers and parallel processing, and efficiency increases in iterative schemes. The solution procedures now used in practice are much more efficient than, for example, COLSOL in KJ Bathe [A]. Solution strategies based on graph theories, optimization and parallel processing are used, but

fundamentally, the basic theories of elimination of equations during the solution and of the iterative schemes have not changed. Hence also the physical process associated with Gauss elimination and static condensation, as described in KJ Bathe [A], remains important. Since we do not discuss graph theories and schemes of optimization and parallel processing, we do not focus here on solution methods in static analysis.

In dynamic analyses, frequency solutions, mode superpositions and direct time integrations are performed, as described in KJ Bathe [A]. For frequency and mode shape solutions, significant advances have been achieved in the Bathe subspace iteration method as described in the next chapter. If any additional degrees of freedom are used to those traditionally employed, like when using overlapping elements, these need to be included in the frequency and mode superposition solutions, see Section 3.3.3.

We next focus on advances achieved in direct time integration schemes.

The "criteria" to have an effective, indeed ideal, time integration scheme for the solution of the governing equations in dynamics -- for applications in engineering and scientific practices -- can be summarized as follows:

The time integration method should be *uniformly effective when used to solve structural vibration and wave propagation problems, and in linear and nonlinear analyses*. Hence the method should give stable and accurate response predictions in all problem solutions, and be reliable, for the parameters used and time step sizes selected.

For these purposes, we shall see that it can be of advantage if the method can be used, depending on the application, as a (globally) first-order accurate or a (globally) second-order accurate scheme. The time stepping scheme may use any “internal” procedure within a time step to have an effective solution

method using large time steps (recognizing that the size of the time steps is also governed by the time variations of the externally applied forces and displacements).

With these criteria in mind, we can state that the *effectiveness of a time integration scheme is measured by –*

***Firstly**, that the scheme can be used to solve the problems in linear and nonlinear analyses, and*

***Secondly**, that the scheme is reliable in that the solutions are always obtained with an expected level of computational effort, and*

***Thirdly**, the computational effort needed for a specific level of accuracy is reasonable, and ideally smallest when considering all time integration schemes.*

.

Here the first and second points for effectiveness focus on the reliability of a scheme, which is very important in engineering practice and frequently more important than the computational cost focused on in the third point.

We already mentioned above the first and second orders of accuracy of methods. When we refer to the order of accuracy of a time integration scheme, we shall always refer to the *global* order of accuracy. This measure gives the accumulation of solution errors during the time stepping up to the desired solution time (in contrast to the *local* solution error which is the error obtained in one time step Δt).

Significant research efforts have been directed to achieve advances in direct time integration methods. However, these efforts have largely focused on single-step methods, in which the solution for the next time, that is, time $t + \Delta t$, only uses the already known solution at time *t* and the unknown solution at time

$t + \Delta t$. The accomplishments we discuss below are based on the idea that it is not necessary to advance the time integration using only solutions Δt apart, like discussed e.g. in K.J. Bathe [A], G. Chung and M. Hulbert [A], X. Zhou and K.K. Tamma [A], but that each step Δt may contain sub-steps, with internal procedures simply performed to advance the solution by Δt. In the simplest case two sub-steps are used, see K.J. Bathe and M.M.I. Baig [A] and K.J. Bathe [E], where this idea was introduced for the solution of structural vibrations and wave propagations and then analyzed and further developed as discussed below.

After the introduction of the approach to use sub-steps in the implicit direct time integration schemes for the response solution in structural dynamics, many researchers focused on that approach for implicit integration see e.g. W. Wen, Y. Tao, S. Duan, J. Yan, K. Wei and D. Fang [A], J. Zhang, Y. Liu and D. Liu [A], H.M. Zhang and Y.F. Xing [A], W. Kim and S.Y. Choi [A], J.Z. Li and K.P. Yu [A] . The basic idea of using sub-steps for a full time step was then also pursued for explicit time integration in G. Noh and K.J. Bathe [A] and, for example, in D. Soares [A], W. Kim and J.H. Lee [A], M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [A].

The objective is here not to survey this field of new developments but only to introduce the basic idea of using sub-steps in the time integration of structural dynamics and present some effective solution schemes We should note that the basic approach used has been around for a long time, see L Collatz [A] for many time stepping procedures. Two sub-steps, like employed in the Bathe method were, for example, used for the solution of first-order differential equations in time governing the response of silicon devices and circuits by R.E. Bank, W.M. Coughran, Jr., W. Fichtner, E.H. Grosse, D.J. Rose and R.K. Smith [A] and to solve for the stochastic-in-time current pulses of an electro-diffusion model by C.L. Gardner, J.W. Jerome and R.S. Eisenberg [A], see K.J. Bathe [E] for more references.

It is also valuable and useful to look at some historical developments of time integration schemes, and here foremost the contributions of the mathematicians Carl Runge and Martin Wilhelm Kutta need to be mentioned. Many Runge-Kutta methods also use multiple solutions to advance a time step and are extremely valuable in many applications, see L Collatz [A]. However, mathematicians are primarily focusing on new strategies and overall analyses, whereas in structural engineering applications, focus is on the reliability and efficiency in solving the dynamic equations of equilibrium and then *mathematical analyses and details* on the reliability and efficiency of a method as given below are important.

Throughout this chapter we denote the full time step as " Δt " and the natural period of a single degree of freedom dynamic system without damping as "T". Recall that for the stability and accuracy analyses, we consider a single degree of freedom system with which we can study all properties of a time integration method, see K.J. Bathe [A].

4.1 The "patch test" for time integration schemes

There is a well-known patch test for the evaluation of spatial finite element discretizations, see K.J. Bathe [A], which is widely used to evaluate new finite element schemes (we refer to and used the test in Chapters 2 and 3). The test evaluates whether the basic required properties are satisfied by the spatial finite element formulation and implementation.

An analogous test was introduced by G. Noh and K.J. Bathe [B] for implicit time integration schemes. In this case, the test measures the various important properties that an implicit time integration scheme should display. The test is also used for explicit time integration schemes but with some differences.

An explicit scheme is by its nature only conditionally stable, that is, the integration time step size must be smaller than a critical value, where the use of physical damping needs to be considered to establish the critical time step. Then using a time step smaller than that value, the patch test seeks to identify whether some basic additional requirements are satisfied.

For an explicit integration scheme, we test for the following two properties:

Property 1e: *Numerical damping*

The scheme should contain numerical damping (irrespective of whether physical damping is present). The numerical damping should be negligible or very small for time steps much smaller than the critical time step value and rapidly increase as the time step size approaches that value. The reason for this desired numerical damping is that high spurious frequencies are then automatically suppressed. These frequencies are not accurate when compared with the exact values (analytical or very accurately established numerical values) of the mathematical model and hence are best suppressed in the calculation of the dynamic response (as in effective implicit time integration, see Property 2i discussed below).

With the conditions of stability and numerical damping satisfied, the time integration scheme should also not show an overshoot in the response predictions (see Property 3i discussed below).

Property 2e*: Accurate response prediction when displacements are imposed*

Since the numerical integration scheme should calculate an accurate response prediction for any loading applied, it should also do so when nodal displacements and not just forces are imposed. The accuracy in the response prediction for imposed initial conditions and externally applied forces is of course required

and assumed to be given when Property 1e is satisfied. Hence this test focuses on whether the time integration scheme also performs well when nodal displacements or velocities are imposed.

It is valuable to note that the commonly used central difference method does not pass Property 1e, that is, the method does not contain numerical damping. However, the methods we discuss below in Sections 4.2.1 and 4.2.2 pass the patch test.

***For an implicit integration scheme, we test for the following four properties*:**

Property 1i: *Unconditional stability*

The method should be unconditionally stable, which means that the calculated solution, hence also any solution errors like due to round-off, will not "blow up" for any time step size used.

Mostly, stability analyses have been performed assuming that physical damping is not present, and that if present, the physical damping will further stabilize the solution. However, as shown in C. Lee, K.J. Bathe and G. Noh [A], physical damping may in fact impair the stability of a time integration scheme. Hence it can be important to perform an analysis to identify how physical damping will affect the performance of a solution scheme.

Property 2i*: Numerical damping*

The method should contain numerical damping, which means that for a large time step Δt to period ratio, $\Delta t/T$, where T denotes the natural period, the numerical solution will correspond to a very damped numerical response. This will automatically suppress any spurious high frequencies in the response which are typically due to modeling and should not be included in the response predictions.

Property 3i: *No overshoot*

The method should not give any undue "overshoot" in the solution, that is, for example, when a single degree of freedom problem is solved with no external force applied and is subject only to an initial displacement with zero initial velocity, the solution should not give, for any time step size used, results significantly larger than the initial displacement.

Property 4i: *Accurate response prediction when displacements are imposed*

As in an explicit time integration solution, the accuracy in the response prediction for imposed initial conditions and externally applied forces is of course required and assumed to be given. Hence this test focuses on whether the time integration scheme also performs well when nodal displacements or velocities are imposed.

Property 1i is relatively easy to satisfy. Many implicit direct time integration schemes exist that are unconditionally stable, but as already mentioned above, it can be important to also identify the effect of physical damping on the stability of a solution scheme.

Property 2i is also not difficult to satisfy, except when we look for the "best" accuracy achieved in the solution using the time integration. That is, numerical damping should not (or only at a very small amount) be present for small time steps Δt, that is, when $\Delta t / T \leq 0.3$, where T is the natural period of the highest excited frequency in the finite element model. In addition, the period elongation error should also be small for such time steps. Then for larger time steps, the response should be numerically damped out of the solution. This is achieved if the time integration scheme introduces numerical damping when $\Delta t > 0.3T$ and this damping increases as the time step size increases. Furthermore, the numerical damping should strongly increase as the time step value gets increasingly larger, that is, when we have $\Delta t >> 0.3T$. With these characteristics about 3 time steps are needed for an accurate response prediction for a period of value T, which is reasonable, and "a scheme is in place for the response in shorter periods to be eliminated from the numerical solution". The response in shorter periods is merely due to "the mesh used", see the discussion below, and modeling features, like, possibly including elements with small and long edges (e.g. sliver elements), or the use of stiff beams or springs to impose constraints, and is not sought in a dynamic solution. If present, we call this a "spurious" response.

"The mesh used", even without special modeling features, can cause some spurious response because in engineering practice the number of degrees of freedom (n) needs to be much larger than the number of frequencies and mode shapes actually excited (p). This is necessary in order to have these p frequencies and associated mode shapes accurately represented (compared to the "exact" or numerically very accurate values). While n needs to be much larger than p to obtain an accurate solution, any calculated response in the $(n - p)$ higher frequencies and associated mode shapes is not accurate (since the time step Δt is too large) and hence is ideally not present.

To obtain further insight, let us consider that the direct integration of the *coupled* governing finite element equations is equivalent to integrating these equations in their modal basis. For the case of no damping (or only modal damping) this basis is given by the n mode shapes that are **M**- and **K**-orthogonal to each other, see KJ Bathe [A] and Chapter 5. Hence the direct integration of the *coupled* finite element equations with the time step Δt is in essence equivalent to integrating all n *decoupled* modal equations with the same time step Δt.

We can thus infer that the scheme of numerical damping to eliminate high frequency response is similar to the use of a mode superposition in which only the p modes are included that are needed for an accurate response prediction, and the static correction is applied. These p mode shapes are excited by the loading (which includes imposed displacements) and, with the corresponding frequencies, are accurate when using an appropriately constructed finite element model, as mentioned above, see also K.J. Bathe [A]. The higher frequency modes are simply not included in the superposition of the p modal responses. Hence, the essential difference between a direct time integration solution and a mode superposition solution is in how the $(n - p)$ highest frequencies and corresponding mode shapes are included: in the mode superposition solution, the effect of these frequencies and corresponding mode shapes is only included through the static correction whereas in the direct time integration, the static response is automatically included and the numerical damping needs to eliminate any additional dynamic response in the $(n - p)$ highest frequencies. However, a difficulty is that the magnitude of the numerical damping increases smoothly as the time step

size increases and not as a step function, which would correspond to a mode superposition solution. For this reason, the number of frequencies and modes to include in a direct time integration solution, that is *p*, by a fine enough mesh and a small enough time step, may need to be larger than in a mode superposition solution.

In the above discussion, we seem to have assumed that the dynamic response is accurately represented by all lowest *p* frequencies and mode shapes and *all* are excited. However, the same reasoning is also applicable if only certain *intermediate* frequencies and associated mode shapes are excited since then the highest such frequency would simply correspond to the *p*th frequency in our discussion. In the direct time integration, the mesh and time step need still to be selected as given above, while in a mode superposition solution the mesh needs to be chosen as discussed above, but the response in those frequencies and modes that are not excited might not be included.

Property 3i should clearly hold because the dynamic response should not be (significantly) larger than the expected physical response of the system, see also J. M. Benítez and F. J. Montáns [A]. An appropriate test problem to use is the one described above: the solution of a single degree of freedom system subjected only to an imposed initial displacement and zero initial velocity. Then, for any time step size used, the numerically measured displacement over time should not significantly exceed the initial displacement.

Property 4i is important because the numerical integration scheme should also give an accurate response prediction when displacements are imposed, and not only when forces are applied. Displacements may be imposed when analyzing the dynamic response of many structures, like machinery, bridges and buildings. This important property was highlighted by G. Noh and K.J. Bathe [B].

We shall show that the methods discussed in Sections 4.3.1 and 4.3.2 pass the patch test.

4.2 Explicit schemes

An explicit time integration scheme can be very effective when used to analyze dynamic events of short duration, like seen in impact analyses. However, explicit schemes are also used in more general analyses, and with heavy damping, even to achieve static solutions.

An important feature of an explicit solution is that, in general, the solution time is advanced with steps that are small compared to the time step used in an (unconditionally stable) implicit time integration. However, frequently many time steps are required for an explicit solution.

The basic ideas of performing an explicit time integration have been discussed in KJ Bathe [A]. We now focus on advances that have been achieved using sub-steps in a solution step from time t to time $t + \Delta t$. It is through the use of sub-steps within Δt -- no longer only using the known solution at time t and the unknown solution at time $t + \Delta t$ in the time integration -- that significant advances have been accomplished.

4.2.1 The Noh-Bathe method

The governing finite element equations to be solved are

$$\mathbf{M\ddot{U} + C\dot{U} + KU = R} \tag{4.1}$$

with given initial conditions, where **M, C,** and **K** are the mass, damping and stiffness matrices, and the vectors **U** and **R** list, respectively, the nodal displacements and externally applied nodal forces. An overdot denotes a time derivative. If the time step size Δt is set and all primary solution variables are known up to time *t,* then the time integration scheme is to calculate the solution at time $t + \Delta t$.

The approach used in the proposed explicit scheme is to calculate the unknown displacements, velocities,

and accelerations by considering the time step Δt to consist of two sub-steps, as in the Bathe implicit time

integration method, see K.J. Bathe [A], K.J. Bathe and G. Noh [A], and G. Noh and K.J. Bathe [C, D, E]. The time step sizes are $\gamma\Delta t$ and $(1-\gamma)\Delta t$ for the first and the second sub-step, respectively, where $\gamma \in (0,1)$, and we develop the solution scheme to have second-order accuracy, see G. Noh and K.J. Bathe [A].

In the Noh-Bathe scheme we consider for the first sub-step,

$$\mathbf{M}\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\gamma\Delta t}\tilde{\dot{\mathbf{U}}} + \mathbf{K}\,{}^{t+\gamma\Delta t}\mathbf{U} = {}^{t+\gamma\Delta t}\tilde{\mathbf{R}} \tag{4.2}$$

$${}^{t+\gamma\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + (\gamma\Delta t)\,{}^{t}\dot{\mathbf{U}} + \frac{1}{2}(\gamma\Delta t)^2\,{}^{t}\ddot{\mathbf{U}} \tag{4.3}$$

$${}^{t+\gamma\Delta t}\tilde{\dot{\mathbf{U}}} = {}^{t}\dot{\mathbf{U}} + \frac{1}{2}(\gamma\Delta t)\,{}^{t}\ddot{\mathbf{U}} \tag{4.4}$$

$${}^{t+\gamma\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + \frac{1}{2}(\gamma\Delta t)[{}^{t}\ddot{\mathbf{U}} + {}^{t+\gamma\Delta t}\ddot{\mathbf{U}}] \tag{4.5}$$

where ${}^{t+\gamma\Delta t}\tilde{\mathbf{R}}$ is chosen as mentioned below, see Eq. (4.12). Then using Eqs. (4.2) to (4.4) we can solve for ${}^{t+\gamma\Delta t}\ddot{\mathbf{U}}$ and with Eq. (4.5) we solve for the final velocities at time $t+\gamma\Delta t$. Note that we use in essence a simple truncated Taylor series in Eqs. (4.3) and (4.4) for the displacements and velocities at time $t+\gamma\Delta t$ but the "1/2" in Eq. (4.4) would usually not be present. We use this "1/2" because it directly leads to the trapezoidal rule in Eq. (4.5); namely, we calculate the unknown accelerations at time $t+\gamma\Delta t$ and we then update in Eq. (4.5) the velocities at time $t+\gamma\Delta t$ for a more accurate solution using the trapezoidal rule. If **M** is a diagonal matrix, this solution is computationally quite inexpensive, and the same holds for the second sub-step (and when using the β_1/β_2 -Bathe method described in the next section).

We should note that the solution can be started directly using the above equations for the first time step, without the need of a special starting procedure, as used for the central difference method, see KJ Bathe [A].

We proceed similarly in the second sub-step,

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\Delta t}\widetilde{\dot{\mathbf{U}}} + \mathbf{K}\,{}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} \tag{4.6}$$

$${}^{t+\Delta t}\mathbf{U} = {}^{t+\gamma\Delta t}\mathbf{U} + [(1-\gamma)\Delta t]\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}} + \tfrac{1}{2}[(1-\gamma)\Delta t]^2\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} \tag{4.7}$$

$${}^{t+\Delta t}\widetilde{\dot{\mathbf{U}}} = {}^{t+\gamma\Delta t}\dot{\mathbf{U}} + \tfrac{1}{2}[(1-\gamma)\Delta t]\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} \tag{4.8}$$

$${}^{t+\Delta t}\dot{\mathbf{U}} = {}^{t+\gamma\Delta t}\dot{\mathbf{U}} + \tfrac{1}{2}[(1-\gamma)\Delta t]\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + [(1-\gamma)\Delta t](q_0\,{}^{t}\ddot{\mathbf{U}} + q_1\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + q_2\,{}^{t+\Delta t}\ddot{\mathbf{U}}) \tag{4.9}$$

where γ, q_0, q_1 and q_2 are parameters to be determined. These parameters, of course, affect the stability and accuracy of the method.

In this sub-step we also use, like in the first sub-step, first a simple truncated Taylor series for the displacements and velocities in Eqs. (4.7) and (4.8) to then calculate the accelerations by use of Eq. (4.6). Thereafter, we update the velocities, like in the first sub-step but now using in Eq. (4.9) the term $(q_0\,{}^{t}\ddot{\mathbf{U}} + q_1\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + q_2\,{}^{t+\Delta t}\ddot{\mathbf{U}})$ instead of $\frac{1}{2}\,{}^{t+\Delta t}\ddot{\mathbf{U}}$. Of course, if $q_0 = q_1 = 0$ and q_2=1/2 we would use the traditional trapezoidal rule like in Eq. (4.5). Therefore, the only difference to the calculations in the first sub-step is this specific updating of the velocities, in which a weighted acceleration involving the three accelerations of the step is used.

We note that the Noh-Bathe scheme is inherently explicit with a lumped mass matrix and any damping matrix. Also, the $\mathbf{K}^{t+\gamma\Delta t}\mathbf{U}$ and $\mathbf{K}^{t+\Delta t}\mathbf{U}$ terms could be evaluated without calculating stiffness matrices but summing over element force vectors, see K.J. Bathe [A], and the same holds for the damping matrix terms.

It is important to recognize that in the Noh-Bathe scheme the dynamic equilibrium equations are considered at time $t+\gamma\Delta t$ (and $t+\Delta t$) to advance the solution, whereas using the central difference method, the equilibrium is considered at time t to advance the solution to time $t+\Delta t$, see K.J Bathe [A].

To analyse the time integration scheme, we consider, as is usual, a representative single degree of freedom system with natural period *T*, frequency ω, hence $\omega = 2\pi / T$, damping ratio ξ and load *r,* see K.J. Bathe [A] and K.J. Bathe and E.L. Wilson [B]. Hence the values given in the figures below for the spectral radii, amplitude decays and period elongations are measured as a function of $\Delta t/T$ (considering the single degree of freedom system) although *T* is of course, in practice, the smallest period of the finite element model (referred to as T_n).

The analysis shows that to have second-order accuracy, we need to set

$$q_0 + q_1 + q_2 = \frac{1}{2}; \quad q_2 = \frac{1}{2} - \gamma q_1 \tag{4.10}$$

and to maximize the stability limit we use

$$q_1 = \frac{1-2\gamma}{2\gamma(1-\gamma)} \tag{4.11}$$

Now the values of q_0, q_1 and q_2 are all given by γ, we see that with the value of γ we can prescribe the amount of numerical dissipation in the high frequency response. Notice that when $\gamma = 1/2$ we have the special case mentioned above already, $q_0 = q_1 = 0$ and q_2=1/2 and hence we use the traditional

trapezoidal rule like in Eq. (4.5).

As mentioned, the presented scheme shows second-order accuracy. We should note that the procedure could be generalized to also offer first-order accuracy.

Figures 4.1 and 4.2 show the spectral radius of the scheme and the percentage amplitude decays and period elongations for different values of γ, calculated as, for example, discussed in K.J. Bathe [A]. The critical time step is given by the value at which the curve of the spectral radius $\rho(\mathbf{A})$ crosses the magnitude 1.0. In these figures, the case of no physical damping is considered.

We note that with $\gamma = 0.5$, the procedure performs like the central difference method, but of course with two sub-steps per time step and gives no numerical dissipation. The critical time step is then twice the value pertaining to the central difference method, that is, we then have $\Delta t_{cr} = 2\, T_n / \pi$, where T_n is the smallest period of the finite element mesh, see KJ Bathe [A]. Figure 4.1 also shows that the critical time step decreases as the value of γ increases since the curves with increasing γ are crossing the value $\rho(\mathbf{A}) = 1.0$ at a decreasing value of $\Delta t/T$.

However, the spectral radius is always smaller than 1.0 for all values of γ that we consider, if $\Delta t/T \leq 0.5$. Hence, we show the amplitude decays and period elongations for that range.

Considering Fig. 4.2, we see that if γ is varied, we reach at $\gamma = 2-\sqrt{2}$ the maximum numerical dissipation. This result is like in the Bathe implicit time integration method, for which the splitting ratio γ also provides the maximum dissipation at $\gamma = 2 - \sqrt{2}$, see Sections 4.3.1 and 4.3.2.

The figure also shows that for a large time step in the range considered, a very large amplitude decay and considerable period elongation are measured when $\gamma = 2 - \sqrt{2}$ or when γ is close to $(2 - \sqrt{2})$. In

practice, the time step used in an explicit time integration can be small, like $\Delta t \leq 0.2T_n$ as dictated by the loading, model details possibly including contact conditions, accuracy sought, etc. However, in our illustrative example solutions, we can use larger time steps to show some capabilities of the method, see Section 4.2.3.

Let us recall that the above observations pertain to satisfying ***Property 1e***.

Property 2e is also satisfied by using the given nodal prescribed displacements in Eqs. (4.2) and (4.6) and corresponding sufficiently accurate velocities and accelerations at these nodes. In special cases, the nodal velocities and accelerations can be evaluated from analytical expressions of the given imposed displacements as a function of time. However, these expressions of the displacements may not be given, and then the equations of the implicit Bathe method, see Section 4.3, or higher-order numerical differentiation may be employed to evaluate the velocities and accelerations at the nodes with prescribed displacements.

In the solution, the availability of numerical damping is important because then possible spurious oscillations of the nodes next to those with prescribed displacements are damped out. With all these properties, the patch test is satisfied.

To obtain further insight, additional analyses are given in P. Zakian and K.J. Bathe [A]. In this contribution, comparisons of the Noh-Bathe method and the central difference method are presented when employed with the spectral element method for the solution of wave propagations. The spatial and spatial-temporal dispersion errors are derived and presented together with illustrative solutions of wave propagation problems.

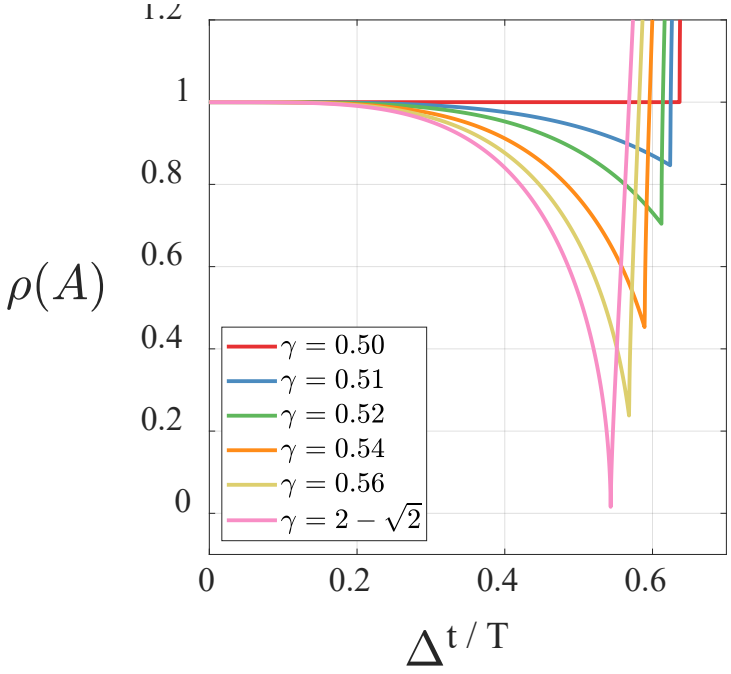


Figure 4.1 Spectral radius of Noh-Bathe explicit time integration scheme, with $\xi = 0.0$, G. Noh and K.J. Bathe [A].

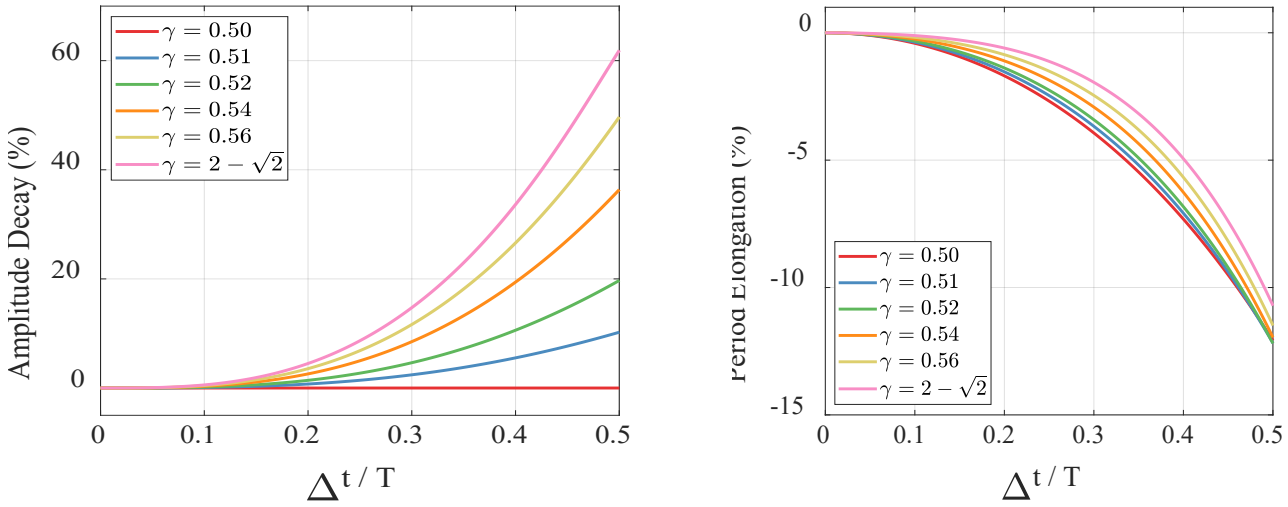


Figure 4.2 Percentage amplitude decay and period elongation of Noh-Bathe explicit time Integration.

We need to also address how to choose ${}^{t+\gamma\Delta t}\tilde{\mathbf{R}}$. In general, external loads are defined and sampled at the discrete time points t and $t+\Delta t$ only, and these values are used for the external forces at the beginning and the end of the time step. With these values, the best value in the sub-step to use is

$$ {}^{t+\gamma\Delta t}\tilde{\mathbf{R}} = (1-\gamma)\,{}^{t}\mathbf{R} + \gamma\,{}^{t+\Delta t}\mathbf{R} \tag{4.12}$$

which corresponds to a "mean" value over the time step. Of course, for smooth loads also the actual value of the load at time $t+\gamma\Delta t$ can be used.

While Figs. 4.1 and 4.2 give the spectral radius, amplitude decay and period elongation for the case of no physical damping, Fig. 4.3 shows the effect of physical damping on the spectral radius when γ = 0.5. Similar results are obtained when γ changes from $\gamma = 0.5$ to $\gamma = 2-\sqrt{2}$. The important observation is that the region of stability is reduced as ξ increases, hence the critical time step is smaller with larger physical damping. We see the same phenomenon in the β_1/β_2 -Bathe method described next.

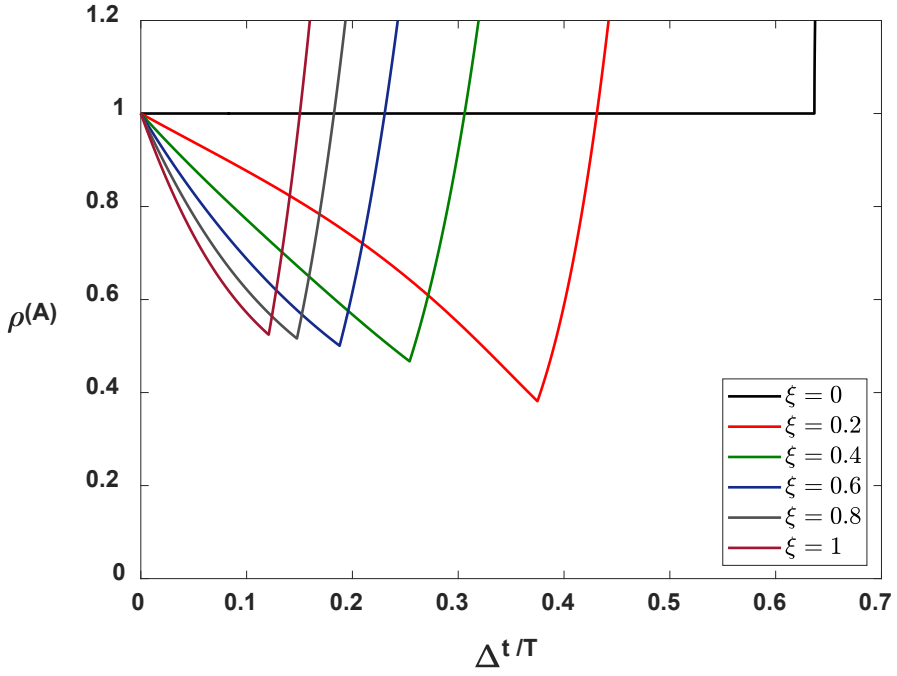


Figure 4.3 The effect of physical damping ($\xi \geq 0.0$) on the spectral radius when $\gamma = 0.5$.

4.2.2 The explicit β_1/β_2-Bathe method

Another new effective explicit time integration scheme is the explicit β_1/β_2 -Bathe method, in which the parameters β_1 and β_2 are used, see M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A] and M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe [A]. The scheme is also based on employing two sub-steps for each time step with the objective of introducing numerical dissipation. This method can be thought of as an important further development of the Noh-

Bathe scheme. The main attributes of the explicit β_1/β_2-Bathe method are that the method is a one parameter scheme, β_1 or β_2 is chosen, and reasonable default values for the parameter used are available. The method can be employed to accurately solve analysis problems of structural vibrations and wave propagations.

In addition, as we describe below, the explicit β_1/β_2-Bathe method can also be used as a "single step" time integration scheme giving first- and second-order accuracy with numerical damping which is not possible when using the Noh-Bathe method (since in this method we use γ to prescribe the numerical damping).

We consider again that the size of the first sub-step is $\gamma\Delta t$ and the size of the second sub-step is therefore $(1-\gamma)\Delta t$. For this scheme, we use, for each sub-step, standard truncated Taylor series on the displacements and the velocities to solve for the unknown accelerations, after which we obtain the final displacements and velocities by applying correction terms using the increment in accelerations. We note that in the Noh-Bathe scheme a correction is only used for the velocities and this correction does not involve a parameter (for a given value of γ).

In the β_1/β_2-Bathe method, the equations of motion at time $t+\gamma\Delta t$ are for the first sub-step

$$\mathbf{M}\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\gamma\Delta t}\mathbf{U} = {}^{t+\gamma\Delta t}\tilde{\mathbf{R}} \qquad (4.13)$$

where we use the usual notation. The load vector in this equation would be established as for the Noh-Bathe scheme, see Eq. (4.12).

The truncated Taylor series expansions give

$$ {}^{t+\gamma\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + (\gamma\Delta t)\,{}^{t}\ddot{\mathbf{U}} \qquad (4.14) $$

and

$$ {}^{t+\gamma\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + (\gamma\Delta t)\,{}^{t}\dot{\mathbf{U}} + (0.5)(\gamma\Delta t)^2\,{}^{t}\ddot{\mathbf{U}} \qquad (4.15) $$

Using Eqs. (4.13) to (4.15) we can calculate the unknown accelerations at time $t+\gamma\Delta t$. These calculations are like those in the first sub-step of the Noh-Bathe method but are not identical to the calculations in that method, see Eqs. (4.2) to (4.4).

We should note, here too, that the solution can be started directly using the above equations for the first time step, without the need of a special starting procedure, as used for the central difference method, see KJ Bathe [A].

Next, we employ Eqs. (4.14) and (4.15) and apply correction terms that involve the just calculated accelerations and the parameters β_1 for the velocities and β_2 for the displacements to obtain the final velocities and displacements at the end of the sub-step

$$ {}^{t+\gamma\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + (\gamma\Delta t)\,{}^{t}\ddot{\mathbf{U}} + \beta_1(\gamma\Delta t)({}^{t+\gamma\Delta t}\ddot{\mathbf{U}} - {}^{t}\ddot{\mathbf{U}}) \qquad (4.16) $$

$$ {}^{t+\gamma\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + (\gamma\Delta t)\,{}^{t}\dot{\mathbf{U}} + (0.5)(\gamma\Delta t)^2\,{}^{t}\ddot{\mathbf{U}} + \beta_2(\gamma\Delta t)^2({}^{t+\gamma\Delta t}\ddot{\mathbf{U}} - {}^{t}\ddot{\mathbf{U}}) \qquad (4.17) $$

Note that these correction terms use in each of the two equations, Eqs. (4.16) and (4.17), the change in accelerations from time t to time $t+\gamma\Delta t$.

We note that in the special case of $\beta_1=\beta_2=0$, the Eqs. (4.16) and (4.17) are simply Eqs. (4.14) and (4.15).

The solution for the second sub-step is obtained in an analogous procedure. The governing equations are for this sub-step

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}+\mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}+\mathbf{K}\,{}^{t+\Delta t}\mathbf{U}={}^{t+\Delta t}\mathbf{R} \tag{4.18}$$

$${}^{t+\Delta t}\dot{\mathbf{U}}={}^{t+\gamma\Delta t}\dot{\mathbf{U}}+(1-\gamma)(\Delta t)\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} \tag{4.19}$$

$${}^{t+\Delta t}\mathbf{U}={}^{t+\gamma\Delta t}\mathbf{U}+(1-\gamma)(\Delta t)\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}}+(0.5)(1-\gamma)^2(\Delta t)^2\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} \tag{4.20}$$

Using Eqs. (4.18) to (4.20) we calculate the accelerations ${}^{t+\Delta t}\ddot{\mathbf{U}}$ and then obtain the final velocities and displacements at time $t+\Delta t$ using

$${}^{t+\Delta t}\dot{\mathbf{U}}={}^{t+\gamma\Delta t}\dot{\mathbf{U}}+(1-\gamma)(\Delta t)\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}}+\beta_1(1-\gamma)(\Delta t)({}^{t+\Delta t}\ddot{\mathbf{U}}-{}^{t+\gamma\Delta t}\ddot{\mathbf{U}}) \tag{4.21}$$

$${}^{t+\Delta t}\mathbf{U}={}^{t+\gamma\Delta t}\mathbf{U}+(1-\gamma)(\Delta t)\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}}+(0.5)(1-\gamma)^2(\Delta t)^2\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}}+\beta_2(1-\gamma)^2(\Delta t)^2({}^{t+\Delta t}\ddot{\mathbf{U}}-{}^{t+\gamma\Delta t}\ddot{\mathbf{U}}) \tag{4.22}$$

The above equations are employed recursively to solve the equations of motion of the finite element system. Since the accelerations are not used in Eqs. (4.14) and (4.15), and (4.19) and (4.20), the scheme is an explicit integration method.

Considering the complete procedure, we see that the solution method is quite simple, even when a banded damping matrix is employed, which makes it an attractive scheme to use. However, for good solution efficiency we usually need to use a lumped mass matrix (as in all explicit solution schemes).

We should also note that in each sub-step the same procedures are employed but of course, for the sub-steps of sizes $\gamma\Delta t$ and $(1-\gamma)\Delta t$. Comparing the method with the explicit Noh-Bathe scheme, we see that the truncated Taylor series for the displacements and velocities are also employed, as in the Noh-Bathe method but not using the "1/2" mentioned earlier, after which the calculations to obtain the final displacements and velocities in the explicit β_1/β_2-Bathe method are in general different.

However, although the calculations are in general different, the equations show that for the special case of no physical damping (that is, $\mathbf{C}=\mathbf{0}$) and using $\beta_1 = 0.5$, $\beta_2 = 0.0$ and $\gamma = 0.5$, the scheme gives the same results as the Noh-Bathe method which is constructed to have second-order accuracy (and then performs like the central difference method, see Section 4.2.1). As we show below, the advantage of the explicit β_1/β_2-Bathe method is that the procedure can directly be used as a first- or second-order scheme with numerical damping, and it can yield very accurate solutions.

The equilibrium is in the above equations of the β_1/β_2 -Bathe method considered at time $t+\gamma\Delta t$ and $t+\Delta t$ in order to advance the solution for a time step (which is, as for the Noh-Bathe method and as pointed out above already, different from the use of the central difference method). When considering the effect of varying γ , we find that we can use $\gamma = 0.5$ and thus reduce the number of unknown parameters to just two values (although the changing of γ could be exploited). The keeping of the value of γ equal to 0.5 is quite different to the use of the Noh-Bathe method, in which the value of γ is used to prescribe the numerical damping and $\gamma = 0.5$ corresponds to second-order accuracy without numerical damping (see Figs. 4.1 and 4.2).

In our further presentation of the scheme, we shall use $\gamma = 0.5$. Since, for this case, the sub-steps are identical, we may regard the technique as a method without sub-steps -- each sub-step then taken as a full time step. The scheme might then be referred to as a "single-step explicit solution method", but of course the value of γ not necessarily being equal to 0.5, still means that the method is fundamentally a composite scheme. Hence, we shall continue to use the scheme as a composite method and see that for this reason the optimal CFL number in some problem solutions can be close to 2 (for reasonable values of β_1 and β_2) and hence for each sub-step close to 1. It is valuable to know that the critical time step of the proposed method without physical damping can be close to twice that of the central difference method, like for the Noh-Bathe method.

The analysis of the method, performed like given for example in KJ Bathe [A], shows that the scheme is consistent for any values of the parameters β_1 and β_2 and conditionally stable, like all explicit

methods, hence only stable if $\Delta t \le \Delta t_{cr}$, where Δt_{cr} is the critical time step size. This time step depends on the parameter values used and also on the physical damping used, that is, the matrix **C**, see M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe [A].

Considering the accuracy, using $\beta_1 \neq 0.5$ the method has first-order accuracy and using $\beta_1 = 0.5$, the method shows second-order accuracy, in each case, independent of the physical damping used. Further analysis, when considering no physical damping, also shows that these two regions are of interest:

Recommendation A

Use for first-order accuracy, the region $0.5 < \beta_1 \le 0.9,\ \beta_2 = 0$

Use for second-order accuracy, the region $\beta_1 = 0.5,\quad 0 \le \beta_2 \le 0.2$

Using values of parameters from these regions excellent results have been obtained, for wave propagation analyses using the first-order scheme, and for structural vibration solutions using the second-order scheme. However, to obtain the most accurate results, experience using the method and perhaps also some experimentation is needed on which precise values to choose and here the use of machine learning procedures may be effective, see Chapter 6.

On the other hand, to obtain reasonably accurate results (and perhaps even the best results), these specific values are recommended:

Recommendation B

For wave propagation solutions, use: $\beta_1 = 0.54; \beta_2 = 0$

For structural vibration solutions, use: $\beta_1 = 0.5\,;\, \beta_2 = 0.04$

Considering these recommendations, it is best to use *Recommendation B* and, if necessary, change the values only slightly according to *Recommendation A.* We demonstrate the use of the values in Section 4.2.3.

Figures 4.4 to 4.7 give the spectral radii, amplitude decays and period elongations of the scheme for various values of β_1 and β_2 when there is no physical damping, that is, $\xi = 0.0$.

In the analyses, amplitude decays and period elongations should only be measured for time step sizes small enough to not be in the region where the spectral radius is larger than 1.0 . This condition is satisfied by the values in Figs. 4.5 and 4.7. We mentioned already in Section 4.2.1 that in practice the time step in an explicit time integration might be quite small like $\Delta t \leq 0.2T_n$ and is also dictated by the loading, nonlinearities, contact conditions, and so on.

Figs. 4.5 and 4.7 show the amplitude decays and period elongations including for large values of β_1 and β_2. For this reason, only relatively small time steps are considered but the data indicates already very large amplitude decays for large values of β_1 in Fig. 4.5 (a) and for large values of β_2 in Fig.4.7 (a). We will note that in the illustrative numerical results given in Section 4.2.3 we use values of β_1 and β_2 close to those recommended in *Recommendation B* and that choice allows the use of large time steps which lead to good results.

These figures show that ***Property 1e*** is satisfied for this scheme.

Considering the case of prescribed displacements, that is, ***Property 2e,*** we need to proceed as in the Noh-Bathe scheme, that is, impose the given nodal displacements and also sufficiently accurate values of the velocities and accelerations at these nodes (the analytically given quantities or accurately evaluated quantities using the equations of the implicit Bathe scheme or higher-order numerical differentiation). Then the patch test is satisfied.

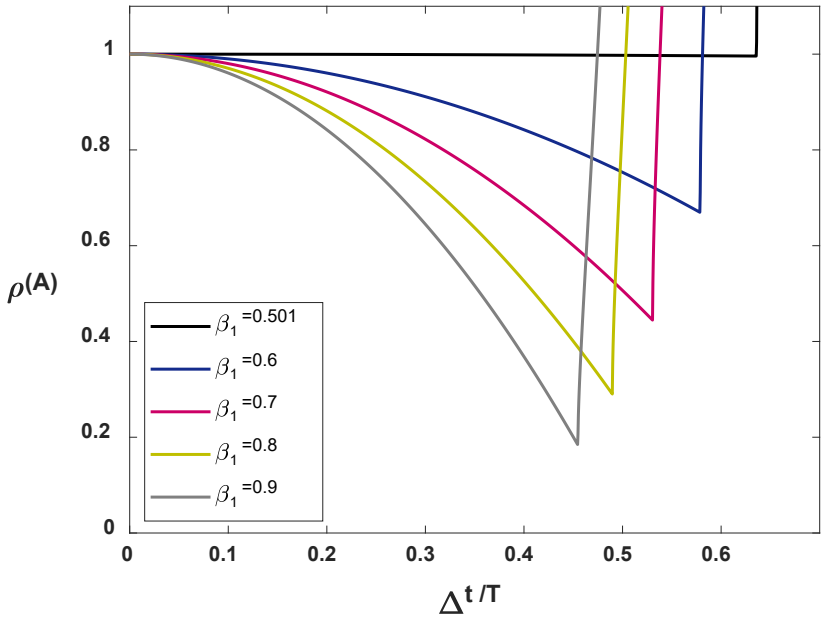
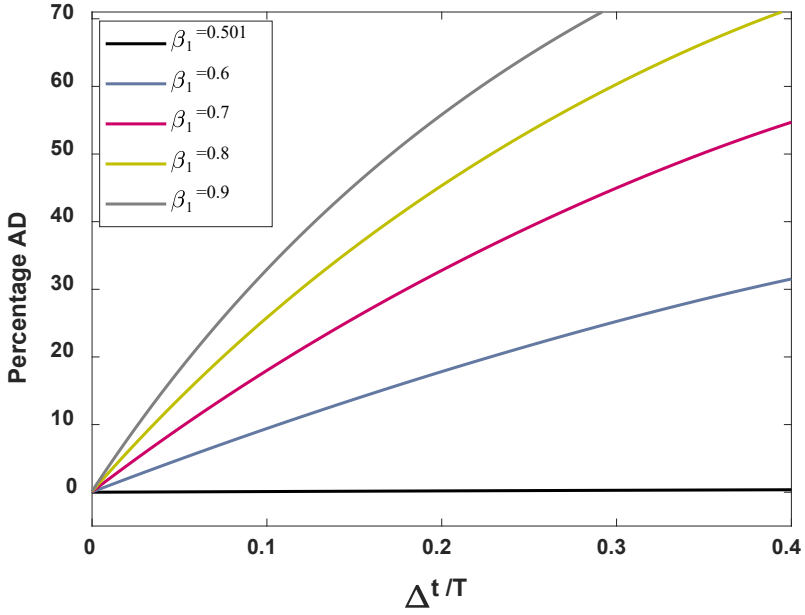
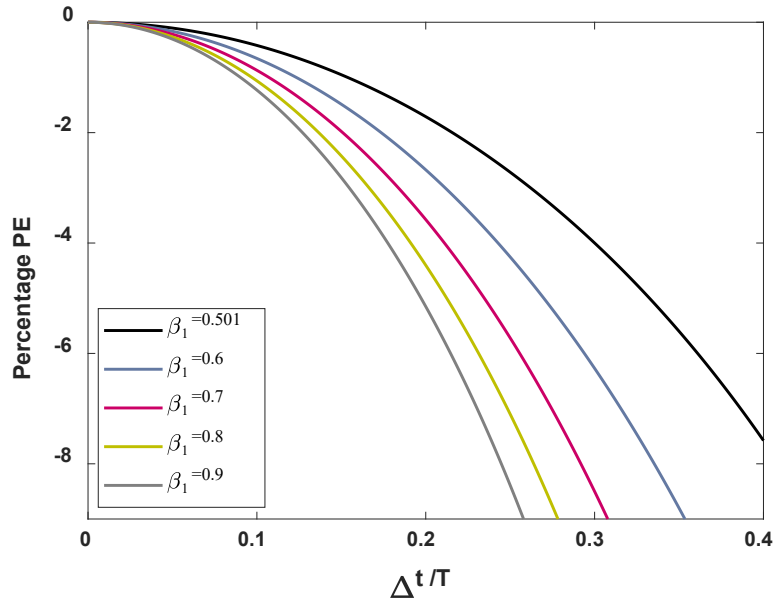


Fig. 4.4 Spectral radius of the explicit β_1/β_2 -Bathe method; first-order scheme, $\beta_2 = 0$, with $\xi = 0.0$.

Regarding Figs. 4.4 to 4.8 and more information, see M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A] and M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe [A].



(a) Amplitude decay of the explicit β_1/β_2 -Bathe method; first-order scheme, $\beta_2 = 0$, with $\xi = 0.0$



(b) Period elongation of the explicit β_1/β_2 -Bathe method; first-order scheme, $\beta_2 = 0$, with $\xi = 0.0$

Fig. 4.5 Amplitude decay and period elongation, first-order scheme

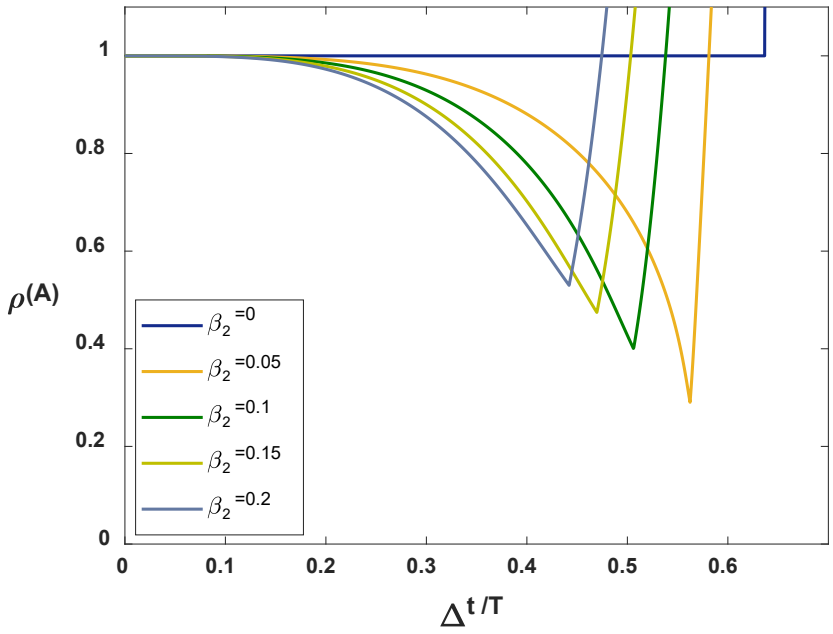
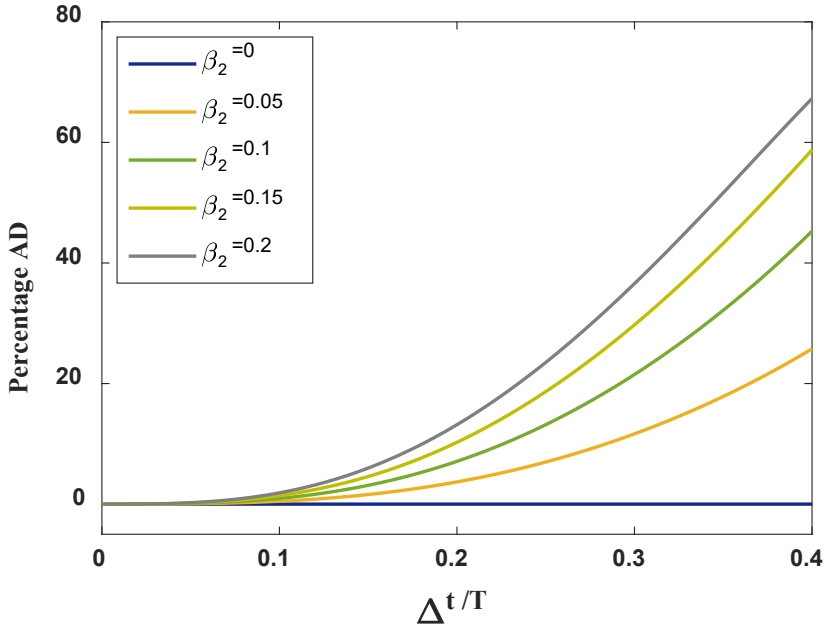
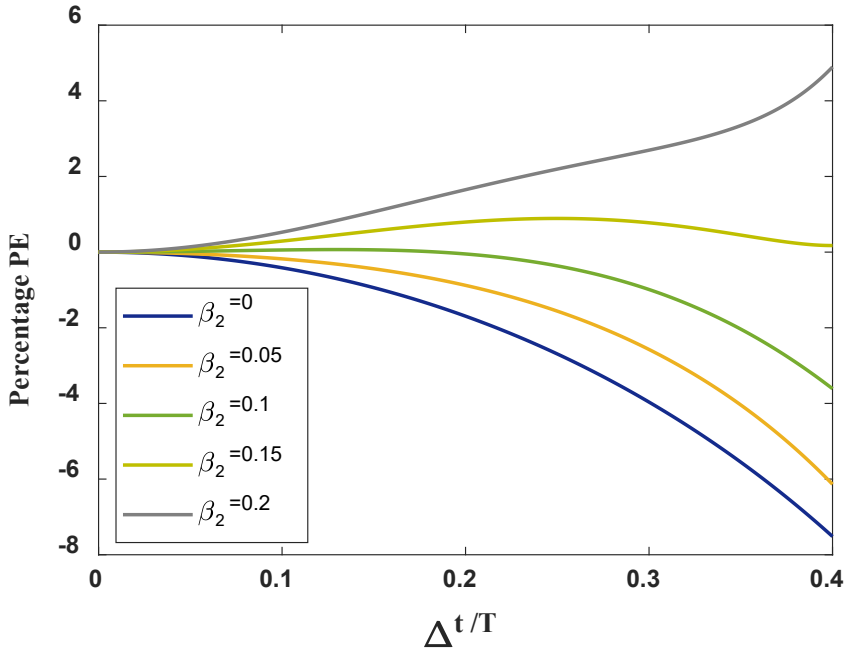


Fig. 4.6 Spectral radius of the explicit β_1/β_2 -Bathe method; second-order scheme, $\beta_1 = 0.5$, with $\xi = 0.0$



(a) Amplitude decay of the explicit β_1/β_2 -Bathe method; second-order accurate scheme $\beta_1 = 0.5$, with $\xi = 0.0$



(b)Period elongation of the explicit β_1/β_2 -Bathe method second-order accurate scheme, $\beta_1 = 0.5$, with $\xi = 0.0$

Fig. 4.7 Amplitude decay and period elongation, second-order scheme

It is interesting to note that the lines of amplitude decays shown for the second-order explicit β_1/β_2 -Bathe method in Fig. 4.7(a) curve in the desired way as those seen for the Noh-Bathe method, but the lines of the period elongations in Fig. 4.7 (b) curve differently, showing in the figure negative and positive values. We also see that considering the first-order method with $\beta_1 = 0.501$, the critical time step is almost twice the critical time step of the Central Difference method, that is, $\Delta t_{cr} = 2\,T_n / \pi$ and considering the second-order method with $\beta_2 = 0.0$, the critical time step is exactly equal to that value. Therefore, using the explicit β_1/β_2 -Bathe method with values of parameters close to those (with two sub-steps), the solution effort is almost the same as when employing the Central Difference method with a time step half the size used with the Bathe method. The major reason is that most of the solution effort in the time stepping

schemes is expended on the evaluation of the stiffness matrix times a solution vector (assuming a lumped mass matrix is used), although this calculation can be performed by summing over the individual element contributions, see K.J. Bathe [A].

In general, the maximum time step size to use can be determined from Figs. 4.4 and 4.6 for the values of the parameters considered in these figures. An analysis reveals that the critical time step for structural vibrations (using the second-order method) is

$$\Delta t_{cr} = \left(\frac{1}{\omega_n}\right)\frac{4}{\sqrt{4\beta_2+1}} \qquad \text{with } \beta_1 = 0.5$$

and for solutions of wave propagations (using the first-order method) is

$$\Delta t_{cr} = \left(\frac{1}{\omega_n}\right)\sqrt{\frac{8}{\beta_1}} \qquad \text{with} \quad \beta_2 = 0.0$$

and we may aim to use a time step very close to the critical values, that is, just below Δt_{cr}.

.

We need to also recognize that, like when using the Noh-Bathe scheme, the critical time step is affected by physical damping, if present. To see the effect of physical damping we consider in Fig. 4.8 the case when there is no numerical damping, that is, we use the parameters $\beta_1 = 0.5$ and $\beta_2 = 0.0$. The figure shows the effect of physical damping on the stability of the scheme, and that, as expected, the reduction in the critical time step value is like when using the Noh-Bathe method (see Fig. 4.3)

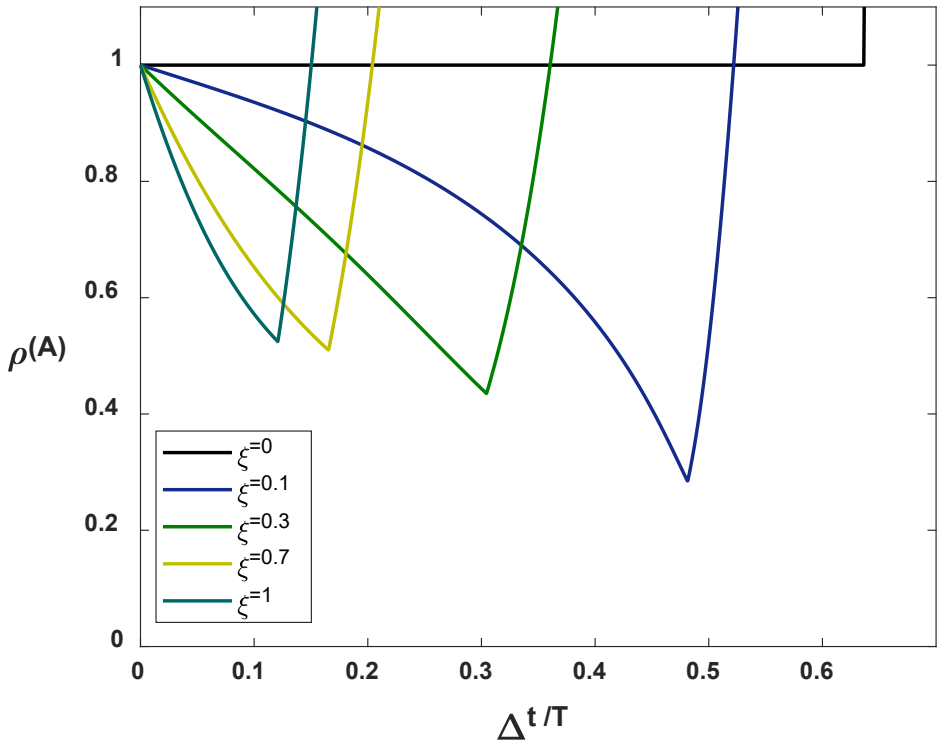


Fig. 4.8 Spectral radius of the explicit β_1/β_2 -Bathe method in the presence of physical damping $0 \le \xi < 1$ for $\beta_1 = 0.5$ and $\beta_2 = 0$

This result is in contrast to how physical damping affects the stability limit of the Central Difference method. The critical time step is unaffected by physical damping when using that method, see K.J. Bathe [A] and M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S. Hamzehei-Javaran, and K.J. Bathe [A], but of course, the accuracy of solution may decrease when physical damping is accounted for.

4.2.3 Illustrative solutions

We give here some illustrative solutions using the explicit Noh-Bathe and β_1/β_2 - Bathe methods. The results are taken from G Noh and KJ Bathe [A] and M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe [A], where also more solutions are discussed. We consider below wave propagation problems, because accurate solutions are difficult to obtain, and for comparison we also include some results obtained using the central difference method. For these solutions, we use the CFL numbers of the time integration schemes. The Noh-Bathe and Bathe explicit time integration schemes are also available as solution methods in the commercial finite element program ADINA [A].

Considering in an overall manner, the performance of the methods, when no physical damping is present, we show that the explicit Noh-Bathe and explicit β_1/β_2 - Bathe methods can be used with a time step size almost twice the size required when using the central difference method. Hence the computational effort using the three methods is almost the same. However, the central difference method gives considerably less accurate solutions.

A clamped uniform bar loaded by a step-load at its free end

We consider first the solution of the seemingly simple problem shown in Fig. 4.9. The material and geometric properties of the bar are $E = 30\times10^6\ psi$, $\rho = 0.00073\ lb/in^3$, $A = 1\ in^2$, and $L = 200\ in$. The bar is at rest when suddenly the step load F(t)=10,000 *lb* is applied. We idealize the bar using 1000 equal 2-node elements and the lumped mass matrix, which gives the smallest period to be $T_n = 3.099422668 \times 10^{-6}$. We solve the problem to illustrate how the explicit β_1/β_2 - Bathe method can be used.

With the recommended values $\beta_1 = 0.54$ and $\beta_2 = 0$, see *Recommendation B,* and using the critical time step formula given above with $c = \sqrt{\frac{E}{\rho}}$, the CFL measured on the usual critical time step (of the central difference method, that is, $\Delta t_{cr} = \frac{\Delta x}{c}$) is

$$CFL = (\frac{c}{\Delta x})(\frac{T_n}{2\pi})\frac{\sqrt{8}}{\sqrt{\beta_1}} = 1.925$$

which gives

$$\Delta t = \text{CFL}(\frac{\Delta x}{c}) = 1.925 \times (\frac{\frac{200}{1000}}{\sqrt{\frac{30\times10^6}{0.00073}}})$$

$$= 1.925 \times 0.986577 \times 10^{-6}$$

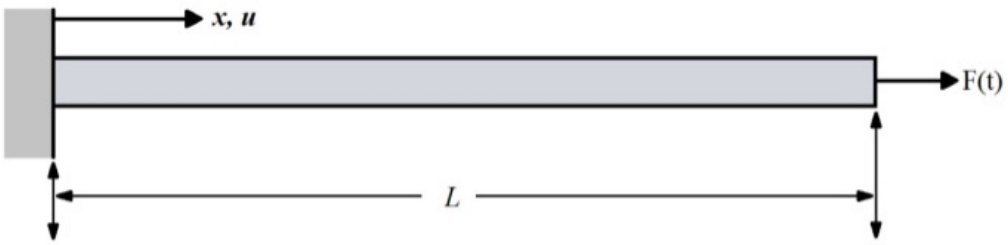
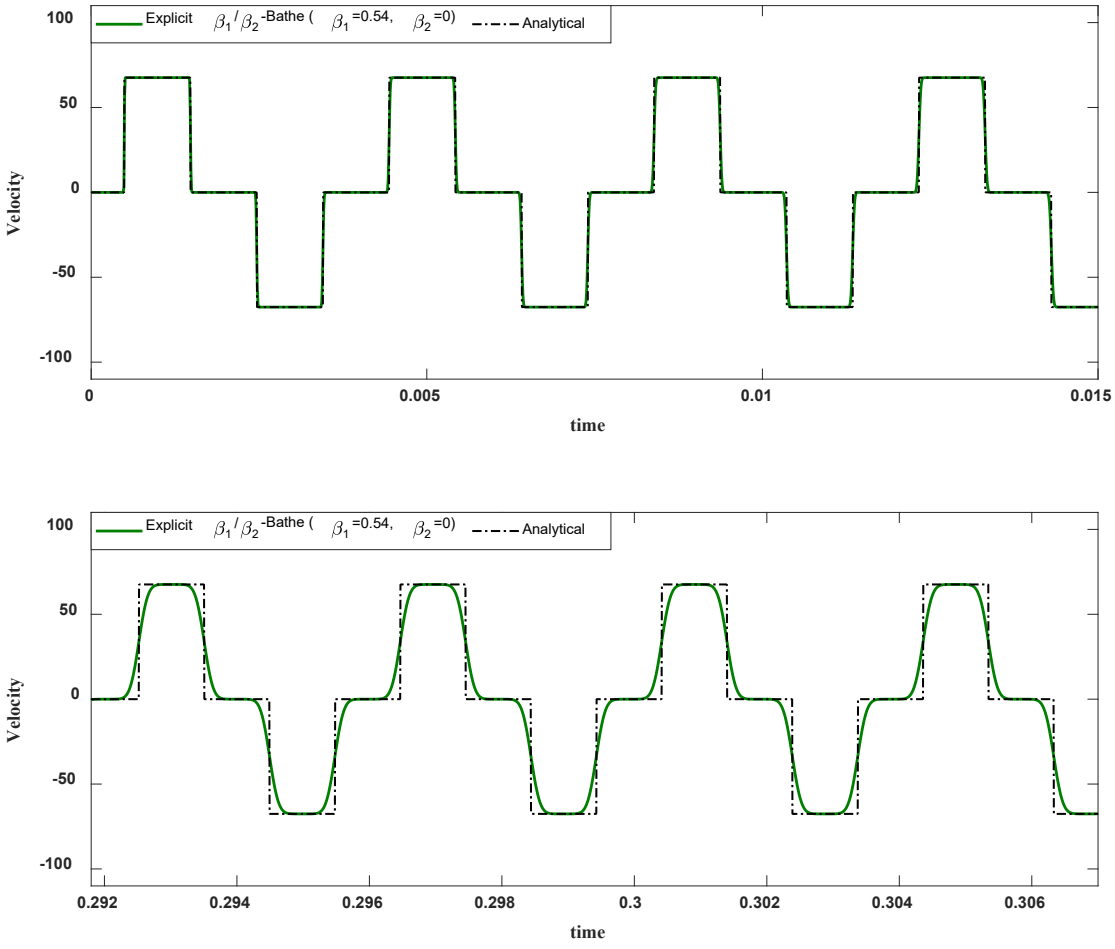


Fig. 4.9 A clamped–free bar subjected to a step end load. Regarding Figs. 4.9 to 4.18 and more discussion, see G. Noh and K.J. Bathe [A], M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A] and M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe [A].

With these data, we obtain the predicted response in Fig. 4.10. We see that an accurate velocity prediction is given for the early response time, but this accuracy deteriorates as time progresses.

To obtain a better accuracy throughout, we select $\beta_1 = 0.501$ and $\beta_2 = 0$ which give using the above calculations CFL = 1.998. Hence the time step we use is almost twice that of the critical value for the central difference method. Figure 4.11 shows the response obtained with this data and we see a remarkable accuracy throughout the time considered. However, to reach this "much better" selection of β_1, experience or some numerical experimentation is required, and, as already mentioned, here the use of machine learning procedures could be effective, see Chapter 6.



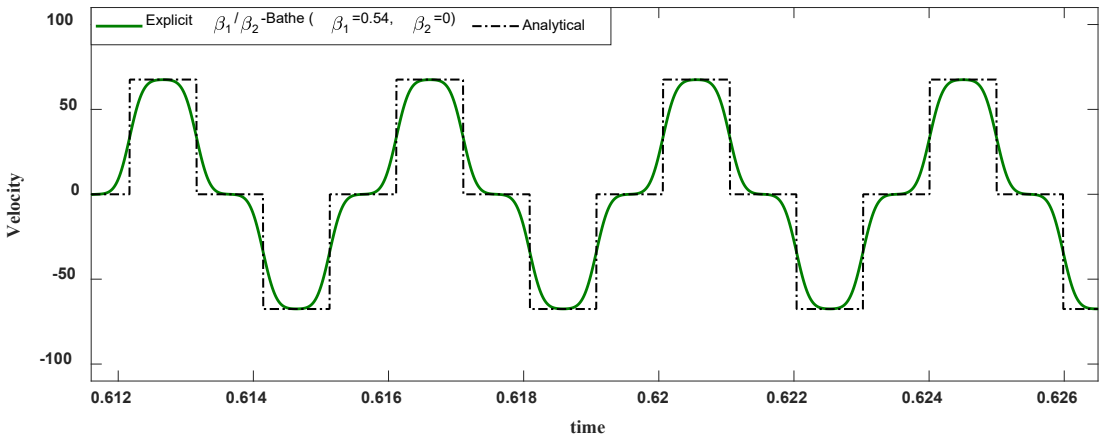
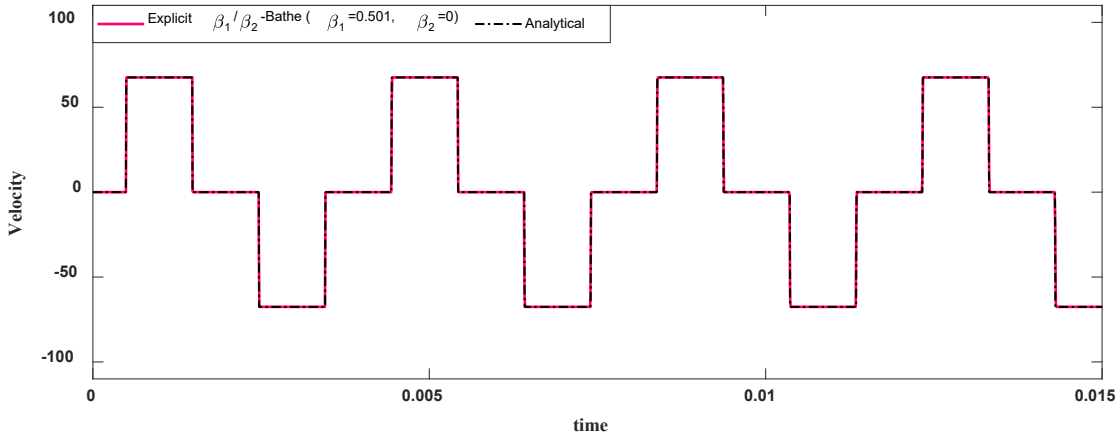


Fig.. 4.10 Predicted velocity at the mid-point of the bar, CFL = 1.925



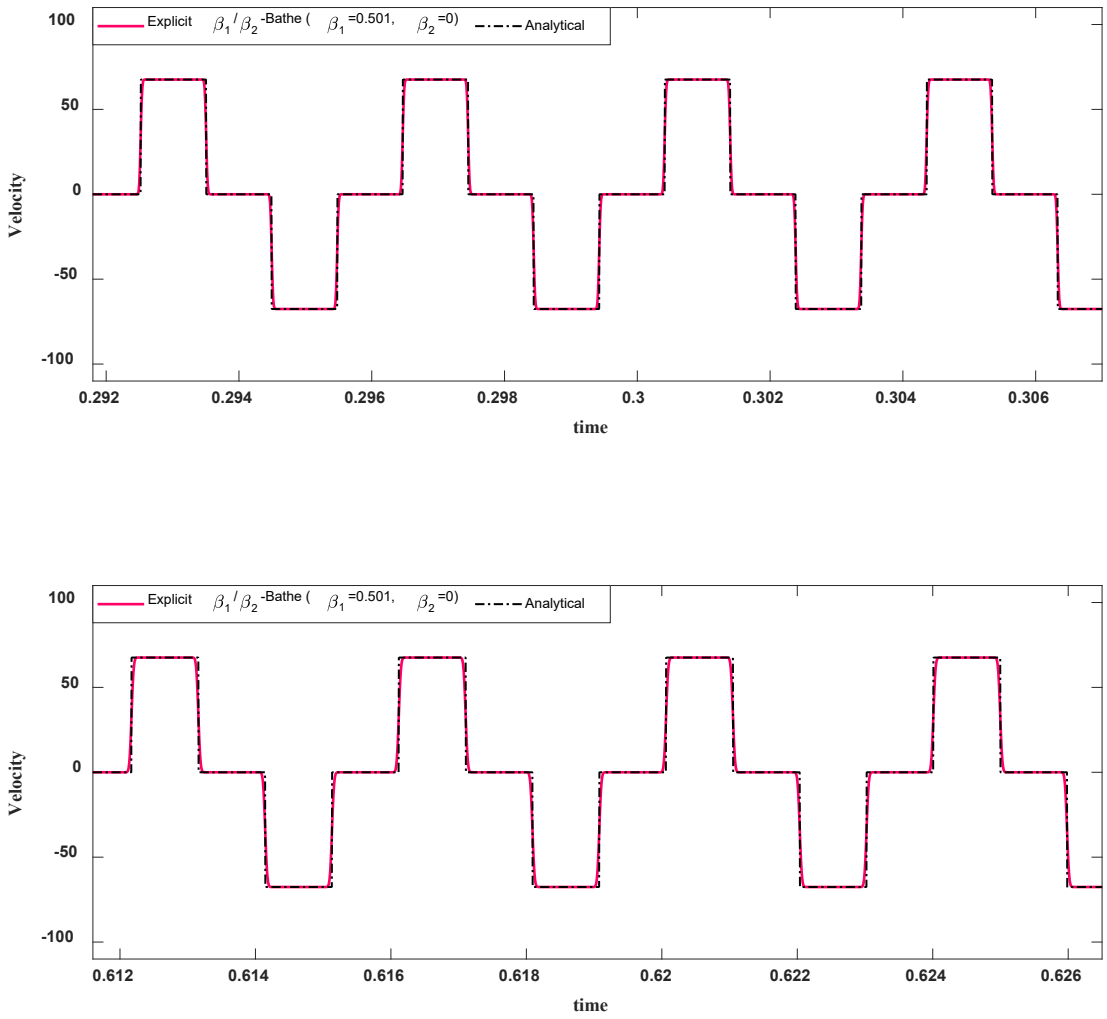


Fig.. 4.11 Predicted velocity at the mid-point of the bar, $\mathrm{CFL} = 1.998$

A bi-material rod subjected to a step-load at its free end

In this analysis, we consider a more difficult problem to solve, namely, a rod of two pieces with different material properties, see Fig. 4.12. The wave speeds in the sections of the rod are $c_1 = 40\sqrt{5}$ m/s and

$c_2 = 20\sqrt{2}$ m/s, respectively. The bar is at rest when suddenly the uniform constant step traction $T_x = 1.0$ is applied.

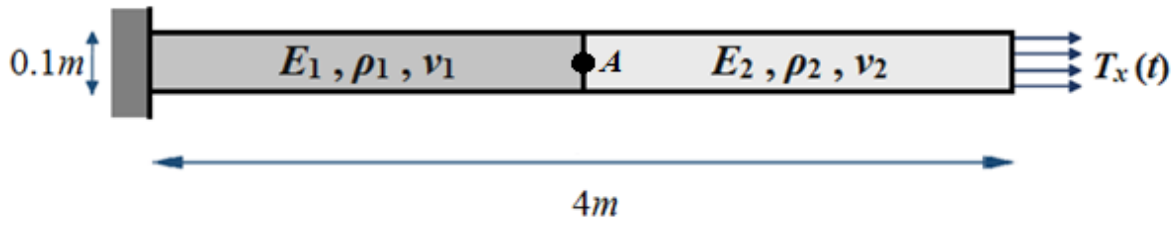


Fig. 4.12 A bi-material rod subjected to a step traction at its right edge

In the study, we solve for the velocity at point *A*. The number of equal size 4- node two-dimensional elements for discretizing the spatial domain is 1×800 and the time-steps used in the Noh-Bathe and explicit β_1/β_2 -Bathe method using the larger wave speed is

$$\Delta t = \text{CFL}(\frac{\Delta x}{c_1}) \qquad \text{(i)}$$

where $(\frac{\Delta x}{c_1})$ is the critical time step of the central difference method. We proceed as in the solution of the uniform bar above. For the explicit β_1/β_2 -Bathe method we use the above given critical time step size (dependent on β_1) and divide it by the critical time step of the central difference method to obtain

$$CFL = (\frac{c_1}{\Delta x})(\frac{T_n}{2\pi})\frac{\sqrt{8}}{\sqrt{\beta_1}} \qquad \text{(ii)}$$

We notice that for β_1 only slightly larger than 0.5, the time step is close to twice the critical time step of the Central Difference method (as in the previous example solution). The formula (i) above now gives the time step for the analysis.

Figs. 4.13 and 4.14 give some illustrative solutions. As expected, the response predictions using the central difference method show significant spurious oscillations, see Fig. 4.13. The Noh-Bathe scheme used with γ = 0.54 and CFL = 1/ γ , which is the value recommended in G. Noh and K.J. Bathe [A], also shows significant oscillations, see Fig. 4.14 (a). Using the explicit β_1/β_2 -Bathe method, with the recommended parameters for wave propagations more accurate results are obtained, see Fig. 4.14(b) and these can be further improved as illustrated in Fig. 4.14(c). The improvement is in this case achieved by using a smaller CFL.

Of course, other values of parameters and time steps could be employed with the Noh-Bathe and β_1/β_2 -Bathe methods with the aim to obtain better results, while the central difference method is optimal in performance when CFL = 1.

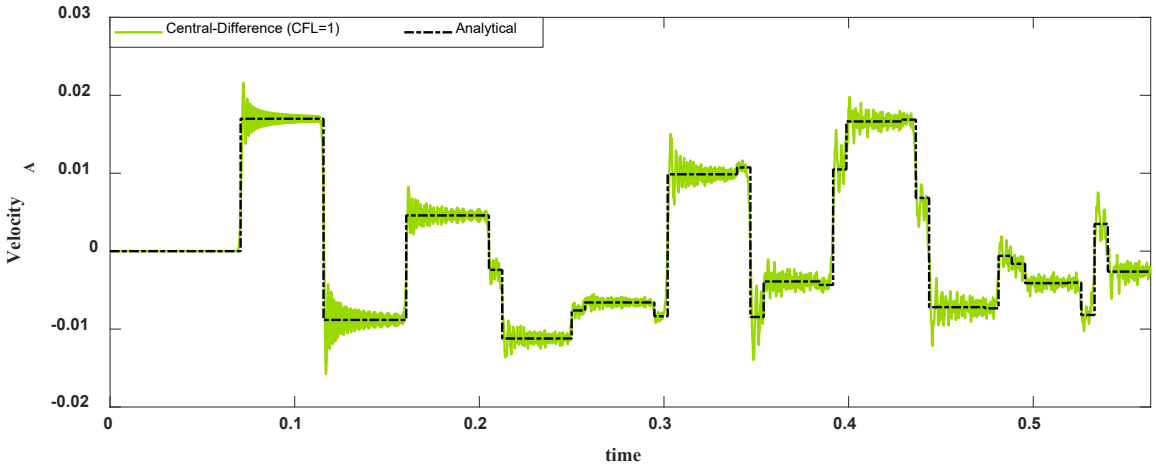


Fig.4.13 Solution using the central difference method with CFL = 1

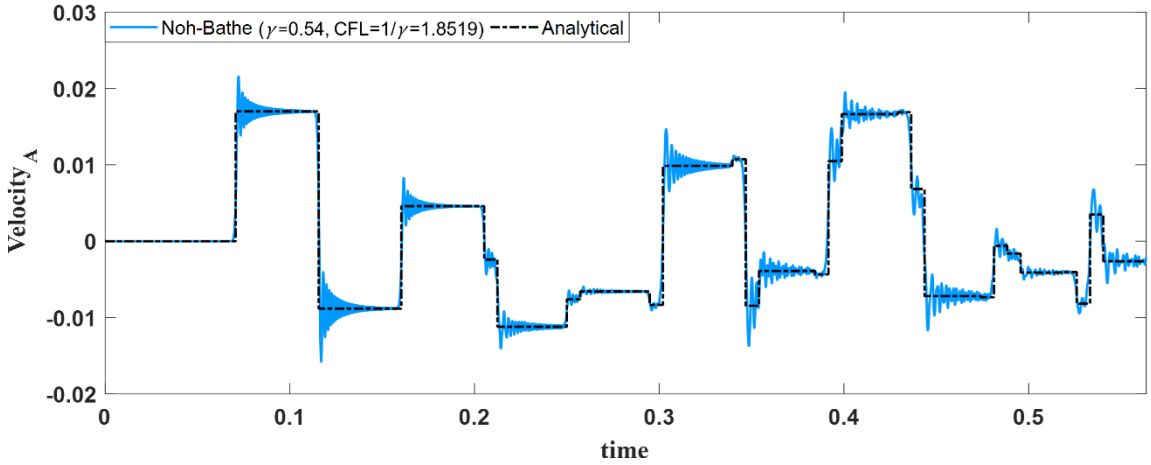


Fig. 4.14 (a) Solution using the Noh-Bathe method with CFL = 1.8519

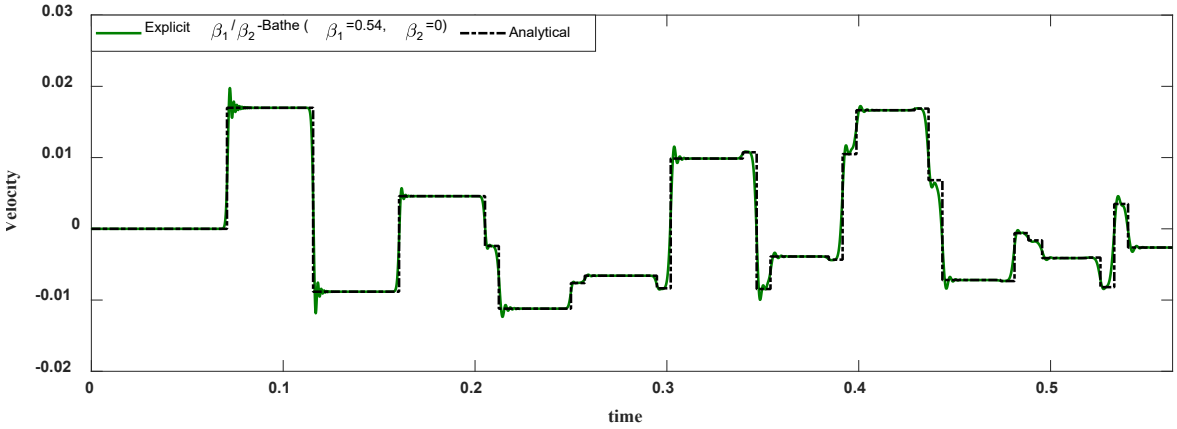


Fig. 4.14 (b) Solution using the β_1/β_2 -Bathe method with CFL = 1.925

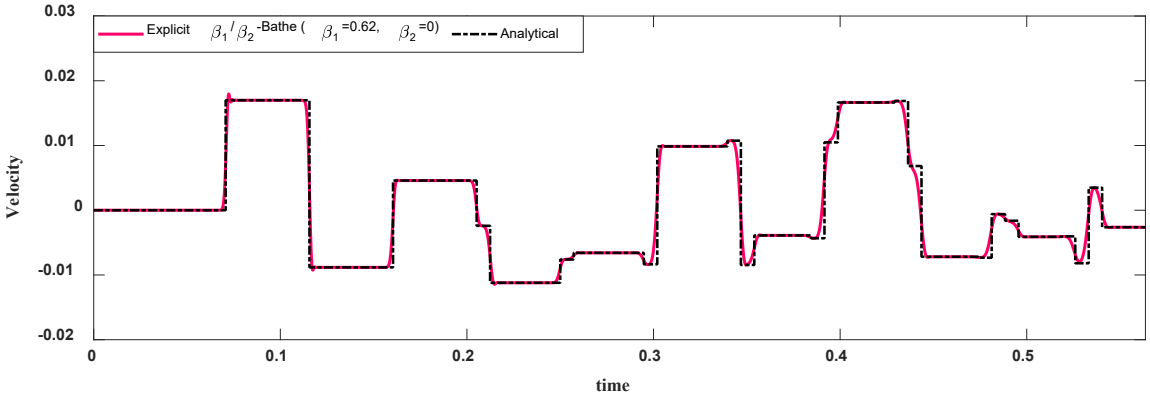


Fig. 4.14 (c) Solution using the β_1/β_2 -Bathe method with CFL = 1.80

A pre-stressed square membrane subjected to initial velocity over the central part

The pre-stressed square membrane with L=10 m shown in Fig. 4.15 is at rest when suddenly it is subjected to a constant unit initial velocity prescribed over its central square domain of side length l =7m. The wave velocity and mass density of the membrane are $c = 10\ m/s$ and $\rho = 1\ kg/m^3$, respectively. Due to symmetry, we only discretize a quarter of the membrane using 150×150 equal 4-node elements.

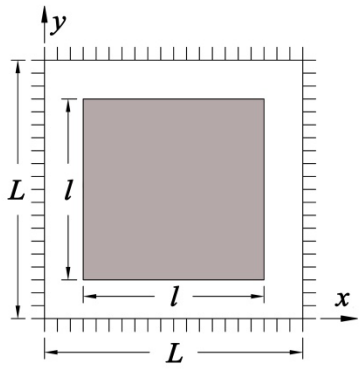
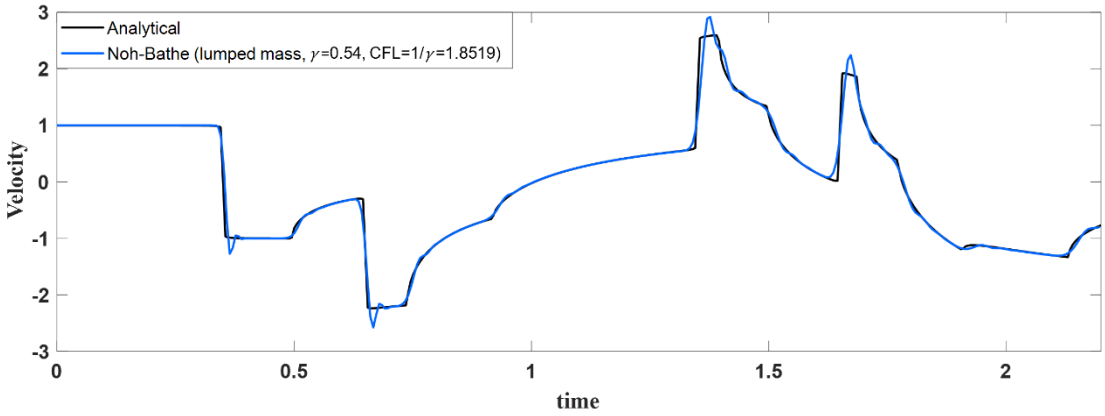
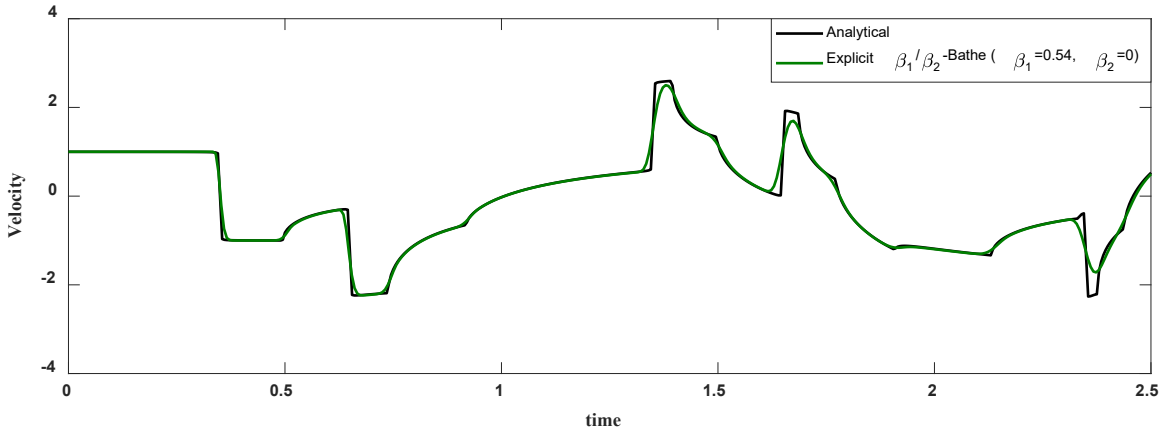


Fig. 4.15. The square membrane

We want to predict the velocity at the center point and since this is a wave propagation problem, we use again the above given critical time step size and $\Delta t = \text{CFL} \times \Delta t_{cr}$



(a) using the Noh-Bathe method, γ = 0.54, CFL = 1.8519



(b) using the explicit β_1/β_2 -Bathe method, using the recommended parameters, CFL = 1.925

Fig. 4.16 Predicted velocity at center point of membrane

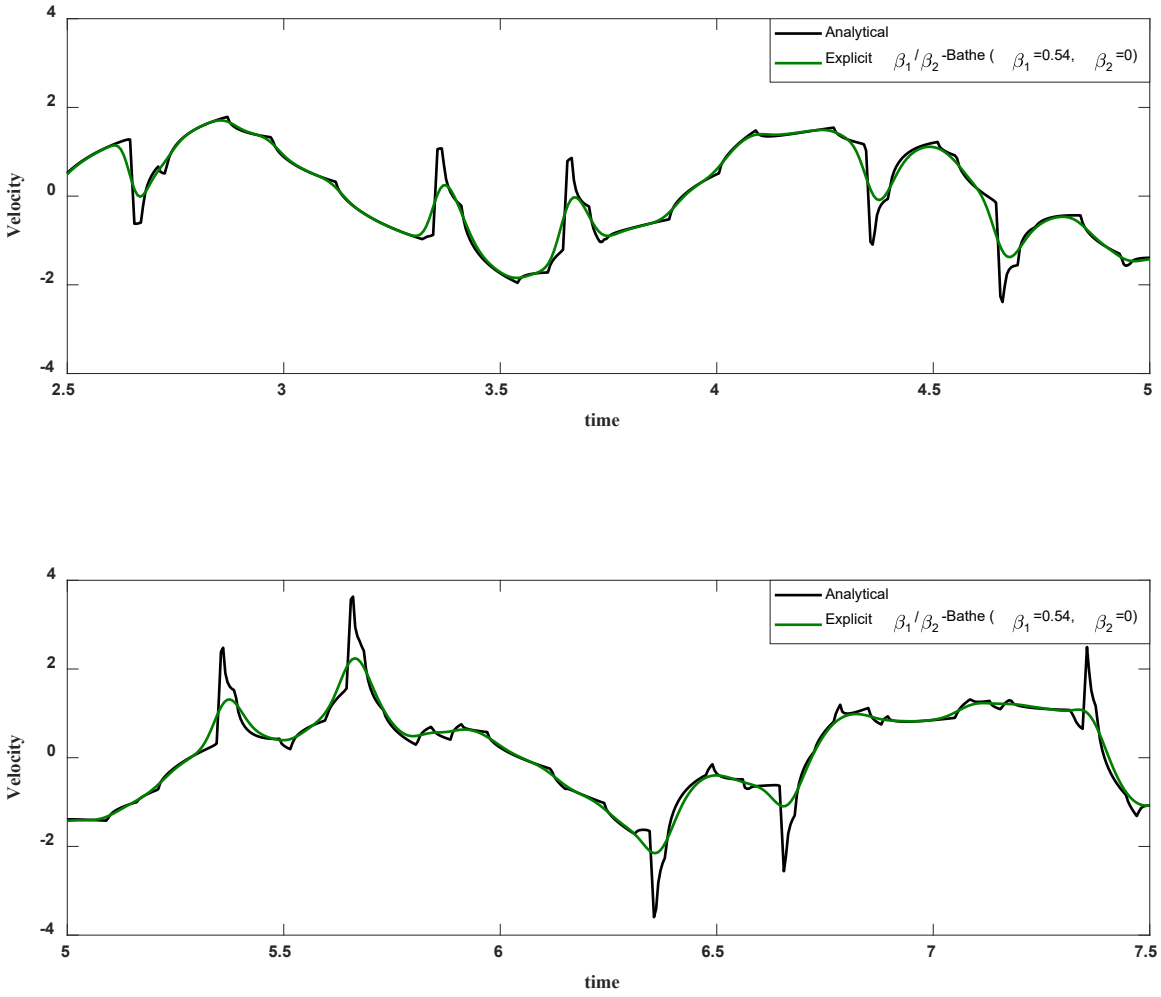


Fig. 4.17 Predicted longtime velocity using the explicit β_1/β_2 -Bathe method with CFL = 1.925.

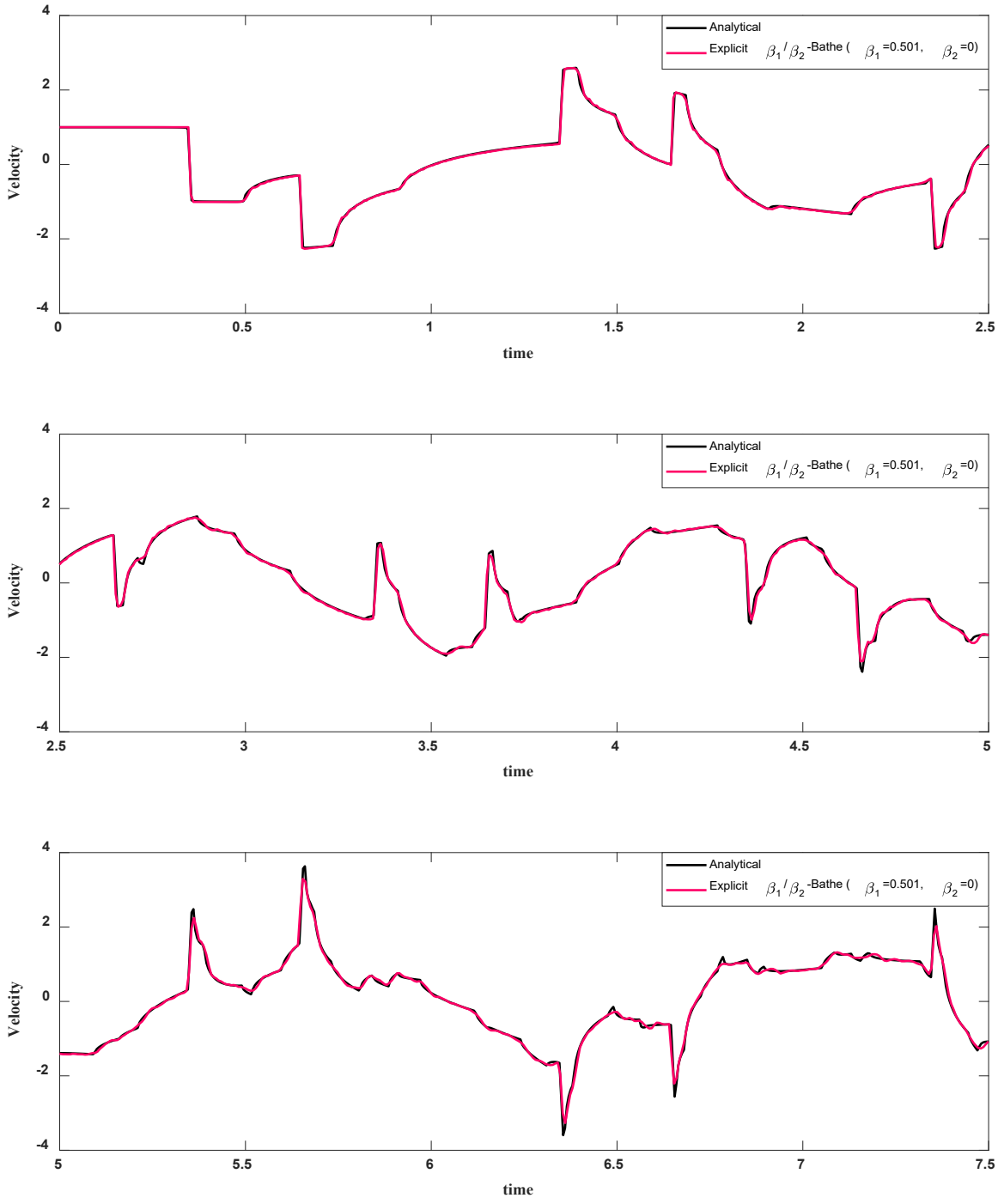


Fig.4.18 Predicted longtime velocity using the explicit β_1/β_2 -Bathe method, CFL = 1.998

Figs. 4.15 to 4.18 show the performance of the different schemes. We use again the recommended value of γ = 0.54 for the Noh-Bathe method. As shown, the Noh-Bathe and explicit β_1/β_2 -Bathe methods yield accurate results for the initial response, with the β_1/β_2 -Bathe scheme giving a slightly better accuracy. We also include the performance of the β_1/β_2 -Bathe method in solving for the longtime response, see Figs. 4.17 and 4.18. The longtime response is more difficult to solve accurately, because small solution errors accumulate, but using the parameters in Fig. 4.18, a remarkable accuracy is reached with the (only) first-order accurate solution scheme. Notice that in this case with a larger (just a slightly larger) time step more accuracy is achieved.

4.3 Implicit schemes

The major difference between an implicit time integration scheme and an explicit scheme is that in an implicit scheme we always consider the *full* dynamic equilibrium equations at time $t + \Delta t$ (or at least at some time point after time *t*) for solution and involve the mass, damping and stiffness matrices as coefficient matrices to the unknown solution vector to advance the solution from time *t* to time $t + \Delta t$. We also use in the Noh-Bathe and β_1/β_2 -Bathe explicit time integration methods the equations at time $t + \Delta t$ but only the mass matrix is used as a coefficient matrix to the unknown solution vector, and not the stiffness and damping matrices. Hence if the mass matrix is diagonal, the solution of the equations, like Eq. (4.18) with Eqs. (4.19) and (4.20), is computationally inexpensive. Recall that, quite differently, in the explicit central difference method, we use the equilibrium equations at time *t* to advance the solution from time *t* to time $t + \Delta t$.

Hence, in an implicit integration, an *effective* stiffness matrix is established – as the name implies, also involving the stiffness matrix – and equations need to be solved with a non-diagonal coefficient matrix. In each time step an equation solver is used to advance the solution. Each time step solution is therefore

computationally more expensive than when using an explicit solution scheme (with a diagonal mass matrix) but when employing an unconditionally stable implicit scheme in general much fewer time steps can be used.

4.3.1 The Bathe method

The basic idea driving the development of the implicit Bathe method of time integration was that we want the method to fulfill the following requirements, which all align with the conditions to satisfy the patch test:

– to be an implicit unconditionally stable second-order accurate scheme,

– to show no overshoot and good accuracy for time steps Δt with $\Delta t / T \leq 0.3$,

– to contain effective numerical damping and give a small error in period elongation,

– to solve problems accurately when displacements are imposed,

and very importantly --

– for a user to be able to solve highly nonlinear problems, reliably and accurately, and

– to be easy to use.

We note that when satisfying the above criteria, also the patch test is passed. An important aim aligned with the last two conditions, those of “solving nonlinear problems” and “easy to use”, is that highly nonlinear problems can be solved and ideally without adjusting any parameters. This is an important objective, particularly attractive for industrial solutions, see R. Kroyer, K. Nilsson, and K.J. Bathe [A] and Bentley Systems, Inc. ADINA [A]. Once published, and with the method widely available and used, further research was carried out for improvements of the implicit scheme. We focus on some valuable achievements in the next section.

As is well known, the Trapezoidal Rule largely fulfills the above requirements but does not

contain numerical damping. Indeed, the trapezoidal rule can show good accuracy in response predictions when $\Delta t / T \leq 0.3$, with no amplitude decays but with errors in the prediction of the periods (that is, the method shows period elongation). However, since numerical damping is not present, also inaccurate and spurious frequencies in finite element models are integrated, although very inaccurately because the time step to period ratio is too large. These frequencies and corresponding mode shapes are frequently only present in the model due to the modeling used and are not accurately representing frequencies of the continuum, hence, they are best suppressed to not contribute to the numerically calculated response, see Section 4.1.

As is also well known, the three-step Euler backward method is unconditionally stable, second-order accurate but gives too much numerical damping and hence is not sufficiently accurate. The amplitude decay and period elongation are too large.

Therefore, it seems judicious to combine these two schemes in order to reach a method that fulfills the above listed requirements and is simple to implement. This is accomplished in the Bathe method, see K.J. Bathe and M.M.I. Baig [A] and K.J. Bathe [E].

In the Bathe method, two sub-steps are used for each full time step Δt. This time stepping lends itself directly to using the three-step Euler backward method in the second sub-step. In the approach, the trapezoidal rule is employed for the first sub-step of length $\gamma \Delta t$ and the Euler method is used for the second sub-step of length $(1-\gamma)\Delta t$.

Let us consider first the simple case using $\gamma = 0.5$, see K.J. Bathe [E]. The trapezoidal rule gives for the first sub-step

$$\mathbf{M}\,{}^{t+\Delta t/2}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\Delta t/2}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\Delta t/2}\mathbf{U} = {}^{t+\Delta t/2}\mathbf{R} \tag{4.23}$$

$${}^{t+\Delta t/2}\mathbf{U} = {}^{t}\mathbf{U} + \frac{\Delta t}{4}({}^{t}\dot{\mathbf{U}} + {}^{t+\Delta t/2}\dot{\mathbf{U}}) \tag{4.24}$$

$$ {}^{t+\Delta t/2}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + \frac{\Delta t}{4}({}^{t}\ddot{\mathbf{U}} + {}^{t+\Delta t/2}\ddot{\mathbf{U}}) \tag{4.25} $$

where the usual notation is used for the matrices and vectors. These are three independent equations, and we can directly solve for the unknown displacements, velocities, and accelerations at time $t+\Delta t/2$. The solution requires the factorization of an effective stiffness matrix.

We then use the Euler backward method for the second sub-step

$$ \mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} \tag{4.26} $$

$$ {}^{t+\Delta t}\dot{\mathbf{U}} = \frac{1}{\Delta t}({}^{t}\mathbf{U} - 4\,{}^{t+\frac{\Delta t}{2}}\mathbf{U} + 3\,{}^{t+\Delta t}\mathbf{U}) \tag{4.27} $$

$$ {}^{t+\Delta t}\ddot{\mathbf{U}} = \frac{1}{\Delta t}({}^{t}\dot{\mathbf{U}} - 4\,{}^{t+\Delta t/2}\dot{\mathbf{U}} + 3{}^{t+\Delta t}\dot{\mathbf{U}}) \tag{4.28} $$

which on solving gives the displacements, velocities and accelerations at time $t+\Delta t$ .The calculations are performed like in the solution for the first sub-step, and if used as given here, a second factorization of an effective stiffness matrix is needed. However, we give below a procedure to avoid a second matrix factorization.

Originally the Bathe method was presented with the general parameter γ for the sub-steps of lengths $\gamma\Delta t$ and $(1-\gamma)\Delta t$ and then for nonlinear analysis specifically using $\gamma = 0.5$, see K.J. Bathe and M.M.I. Baig [A] and K.J. Bathe [E]. The need for using different effective stiffness matrices in the sub-steps does not necessarily present a disadvantage in nonlinear analysis because usually in such analyses, a number of tangent stiffness matrices need to be factorized in the Newton-Raphson iterations when solving for the response from time t to time $t+\Delta t$. Then using sub-steps, a faster convergence may be reached and the total number of tangent stiffness matrices established and factorized in the two sub-steps may

actually be less than when using the full step (in which case also convergence difficulties may be encountered).

The Bathe method is simple and an analysis shows that it does not overshoot and has good accuracy characteristics. Also, no special starting procedure needs to be used to calculate the response, and the sub-step calculations can simply be regarded as internal calculations to advance the solution for the full time step.

An analysis of the method also shows that the scheme performs well when displacements are imposed, see G. Noh and K.J. Bathe [B]. Furthermore, the method also remains unconditionally stable when physical damping is present, see C. Lee, K.J. Bathe and G. Noh [A]. Hence the patch test requirements discussed in Section 4.1 are all passed.

An important requirement is that the method be computationally effective when compared with other time stepping schemes. To measure a method against other solution schemes, we need to study various aspects. The first consideration is whether a user can actually solve the problems to be solved with the proposed method.

In linear analysis, any proposed effective method should be able to solve efficiently any linear well-posed dynamic problem. However, in nonlinear analysis, the method needs to be sufficiently robust to reach the solution, and the Bathe method was initially proposed for such analyses, see K.J. Bathe [A, E]. The objective in that development was to establish a method that could reliably be used in solving engineering problems when other methods failed, see e.g. R. Kroyer, K. Nilsson, and K..J Bathe [A], Z. Kazancı and K.J. Bathe [A], and K.J. Bathe [O].

Insight into the Bathe implicit time integration scheme is largely linked to studies performed on important further developments, presented in the next section, and is given in K.J. Bathe and G. Noh [A], G. Noh and K.J. Bathe [C, D, E], M.M. Malakiyeh, S. Shojaee and K.J. Bathe [A], and M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B], where also the effect of using different values of γ are discussed and the performances of the developments are compared with the predicted response using the Newmark method and the generalized α-method, see also S.B. Kwon, K.J. Bathe and G. Noh [A, B], and C. Lee, K.J. Bathe and G. Noh [A].

In linear analysis, we study the numerical errors and computational effort per time step. This requires to establish for a solution scheme the spectral radius, the amplitude decay and the period elongation as a function of $\Delta t / T$ for a single degree of freedom system, as we pursued in Section 4.2 for explicit schemes. We can then largely identify whether, with the given accuracy of the different methods per time step, the Bathe method is less or more effective than using another scheme. Although the solution effort per time step is larger, the Bathe method may be more effective because when using the method less time steps can be employed to have a given accuracy for the *total* solution.

Instead of using only two sub-steps, more sub-steps can naturally be employed, see K.J. Bathe and M.M.I. Baig [A] where some additional sub-stepping approaches are also considered.

We discuss the effectiveness of the Bathe method in the next section because this procedure is a special case of the more general methods, the ρ_∞-Bathe and β_1 / β_2-Bathe schemes. However, it is valuable to continue to recall how we want a desirable method to perform, as summarized at the beginning of this section -- second-order accuracy, no overshoot, effective numerical damping, and efficiency in solving nonlinear problems. Recent research has also shown that first-order accuracy can be advantageous in the

solution of wave propagation problems, see also Section 4.2, and hence ideally the method can also be used as a first-order scheme. As discussed below, this use is also possible with the Bathe implicit methods.

4.3.2 The ρ_∞-Bathe method

The ρ_∞-Bathe method represents a generalization of the Bathe method, see G. Noh and K.J. Bathe [D]. We present this procedure below and analyze the method with respect to stability, accuracy and computational effort. This discussion includes these considerations also for the Bathe method as a special case, and will also refer to the implicit β_1 / β_2-Bathe method.

As in the Bathe method, in the ρ_∞-Bathe method, we calculate the unknown displacements, velocities, and accelerations by considering the time step Δt to consist of two sub-steps. The sub-step sizes are $\gamma\Delta t$ and $(1-\gamma)\Delta t$ for the first and second sub-steps, respectively.

In the first sub-step, we use the trapezoidal rule for the equilibrium at time $t+\gamma\Delta t$ (as in the Bathe method)

$$\mathbf{M}\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\gamma\Delta t}\mathbf{U} = {}^{t+\gamma\Delta t}\mathbf{R} \tag{4.29}$$

$${}^{t+\gamma\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + \frac{\gamma\Delta t}{2}({}^{t}\dot{\mathbf{U}} + {}^{t+\gamma\Delta t}\dot{\mathbf{U}}) \tag{4.30}$$

$$^{t+\gamma\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + \frac{\gamma\Delta t}{2}({}^{t}\ddot{\mathbf{U}} + {}^{t+\gamma\Delta t}\ddot{\mathbf{U}}) \tag{4.31}$$

and in the second sub-step, we use the following relations for the equilibrium at time $t+\Delta t$.

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} \tag{4.32}$$

$$^{t+\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + \Delta t\ (q_0\,{}^{t}\dot{\mathbf{U}} + q_1\,{}^{t+\gamma\Delta t}\dot{\mathbf{U}} + q_2\,{}^{t+\Delta t}\dot{\mathbf{U}}) \tag{4.33}$$

$$^{t+\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + \Delta t\ (s_0\,{}^{t}\ddot{\mathbf{U}} + s_1\,{}^{t+\gamma\Delta t}\ddot{\mathbf{U}} + s_2\,{}^{t+\Delta t}\ddot{\mathbf{U}}) \tag{4.34}$$

where we use the usual notation. We note that Eqs. (4.29) to (4.34) are a generalization of Eqs. (4.23) to (4.28), now using not necessarily equal sub-step sizes, and Eqs. (4.33) and (4.34) are a very general Ansatz for solution.

To reach a method that depends only on a small number of parameters, we use $s_0 = q_0,\ s_1 = q_1$ and $s_2 = q_2$ and the parameters q_0, q_1 and q_2 are established to have stability and good accuracy. For the method to be unconditionally stable and requiring second-order accuracy, we use

$$q_0 = (\gamma - 1)q_1 + \frac{1}{2} \quad ; \quad q_2 = -\gamma q_1 + \frac{1}{2} \tag{4.35}$$

Hence for a given value of γ, the value of q_1 gives q_0 and q_2. To prescribe the amount of numerical damping in the high frequency range, we determine the value of q_1 using γ and a new parameter, the spectral radius ρ_∞,

$$q_1 = \frac{\rho_\infty + 1}{2\gamma(\rho_\infty - 1) + 4} \tag{4.36}$$

where the spectral radius ρ_∞ is defined as

$$\lim_{\Delta t / T \to \infty} \rho(\mathbf{A}_{\rho_\infty\text{-Bathe}}) = |\rho_\infty|, \quad \rho_\infty \in [-1,1] \tag{4.37}$$

and $\mathbf{A}_{\rho_\infty\text{-Bathe}}$ is the approximation operator of the ρ_∞-Bathe method for the full-time step. We use that convention for ρ_∞ throughout the book, hence when comparing the spectral radii of a two sub-step method, like the ρ_∞-Bathe method, and a single step method, like the Newmark method, a combined approximation operator should be employed for the single step scheme with two sub-steps for the full step, that is, the Newmark method would be applied twice, like in the Bathe method which also uses two sub-steps, see for example G. Noh and K.J. Bathe [E].

The ρ_∞-Bathe method has therefore two free parameters, γ and ρ_∞. We mostly use $\rho_\infty \in [0,1]$ but negative values of $\rho_\infty \in [-1,0]$ may also be employed, and some benefits using a larger value can arise, see G Noh and KJ Bathe [D].

We note that this method is therefore developed to be a second-order solution scheme, unless the values in (4.35) are not used for the expressions in (4.33) and (4.34). The β_1 / β_2-Bathe implicit scheme, on the other hand, can directly be employed as a second-order or first-order method, see the discussion below.

An analysis shows that using $\rho_\infty \in [0,1]$, the scheme provides the same effective stiffness matrix for each sub-step, and the maximum amplitude decay and the minimum period elongation with the following value

for γ:

$$\gamma_0 = \frac{2-\sqrt{2+2\rho_\infty}}{1-\rho_\infty};\ \ \gamma_0 = 0.5 \text{ if } \rho_\infty = 1 \tag{4.38}$$

Using the relation in Eq. (4.38), we have that the method is a one-parameter scheme for which only one effective stiffness matrix is calculated and factorized. However, still, for each full step, two forward reductions and two back-substitutions are needed, hence about twice the solution effort of methods that do not use sub-stepping is required, like when using the Newmark method, see KJ Bathe [A]. On the other hand, when using the Bathe method in linear analyses, larger time steps can frequently be employed for an overall required solution error, G Noh and KJ Bathe [E] and KJ Bathe and G Noh [A, C] and highly nonlinear problems might be solved more effectively, see R. Kroyer, K. Nilsson, and K.J. Bathe [A].

While the relation (4.38) has been derived assuming no physical damping, using this value when physical damping is present, gives also a stable solution, see C Lee, KJ Bathe, and G. Noh [A]. Since the method can also be used when displacements are imposed, and not just forces, the solution method passes the patch test.

Fig. 4.19 shows the spectral radii, percentage period elongations and amplitude decays of the ρ_∞-Bathe method with $\gamma = 0.5$ and γ_0 for various values of ρ_∞, see G Noh and KJ Bathe [D, E] The results in Fig. 4.19 are also directly applicable to the discussion in Section 4.3.1, where we considered the Bathe method

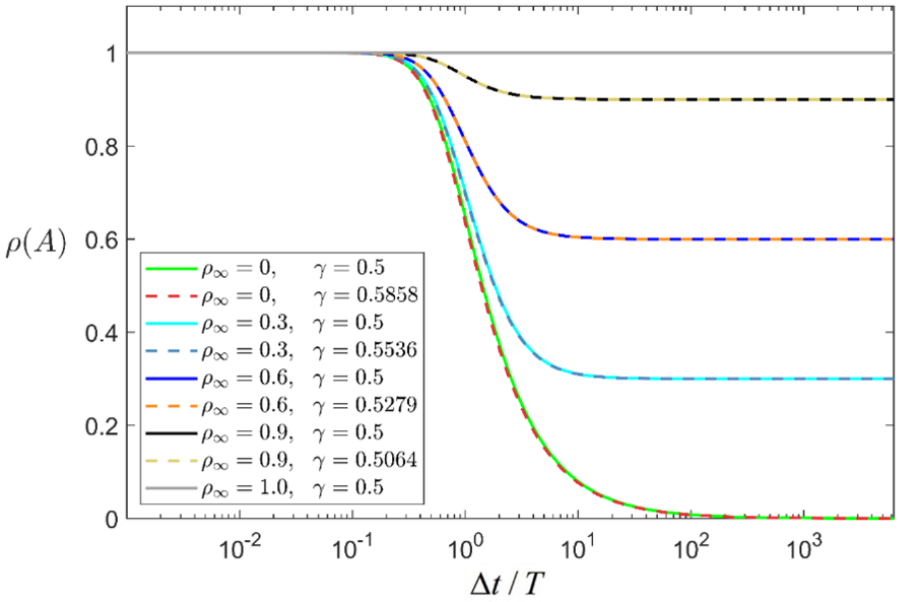
with $\gamma = 0.5$ and inherently $\rho_\infty = 0.0$. We notice that using $\gamma = 0.5$ provides practically the same curves as when using $\gamma = \gamma_0$ for all $\rho_\infty \in [0,1]$. In practice, the question must be which parameters to use, and here clearly the value corresponding to γ_0 in Eq. (4.38) with $\rho_\infty = 0.0$ is preferable over $\gamma = 0.5$ with the Bathe method, because only one effective stiffness matrix needs to be factorized. The numerical damping will then be provided as shown in Fig. 4.19 (b).

Considering these figures, we can conclude that provided a second-order scheme is preferably used, the values of γ_0 and $\rho_\infty = 0.0$ with a time step $\Delta t / T_0 \leq 0.2$ is in general effective, that is, at least 5 time steps per period of interest should be used.

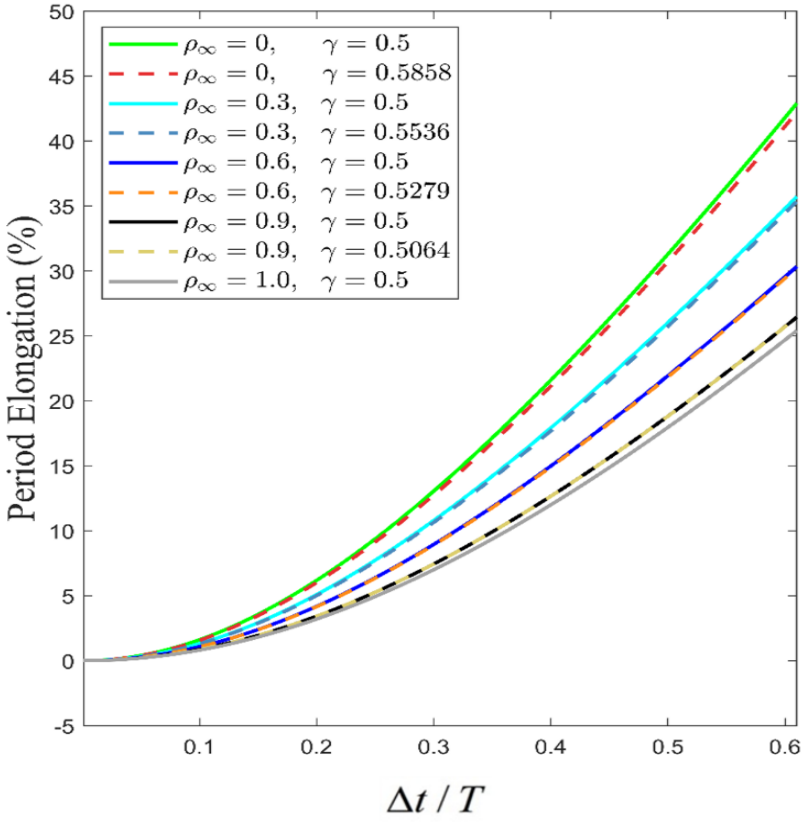
If with these parameter values the solution results are then deemed to be not sufficiently accurate, other values of the parameters γ and ρ_∞ can be selected, and also, a first-order scheme may be chosen (e.g. for wave propagation solutions) using different values for the parameters q_0, q_1 and q_2 . However, the choice of these values may be complex in engineering practice; a more direct way is then to use the implicit β_1 / β_2 -Bathe method, discussed next. Since the analyses of the ρ_∞ -Bathe and the implicit β_1 / β_2 -Bathe methods show a close relationship between the two procedures, if desired, also the appropriate parameter values of β_1 and β_2 of the β_1 / β_2 -Bathe scheme can be employed to evaluate the parameters q_0, q_1 and q_2 for use of the ρ_∞ -Bathe method with first-order accuracy, see the dicussion below and the relationships in (4.41).

In addition, we also have the option to use the ρ_∞ -Bathe method with first-order accuracy by using the parameter values given below in the comparison with the Newmark method, see Eqs. (4.43) and (4.45).

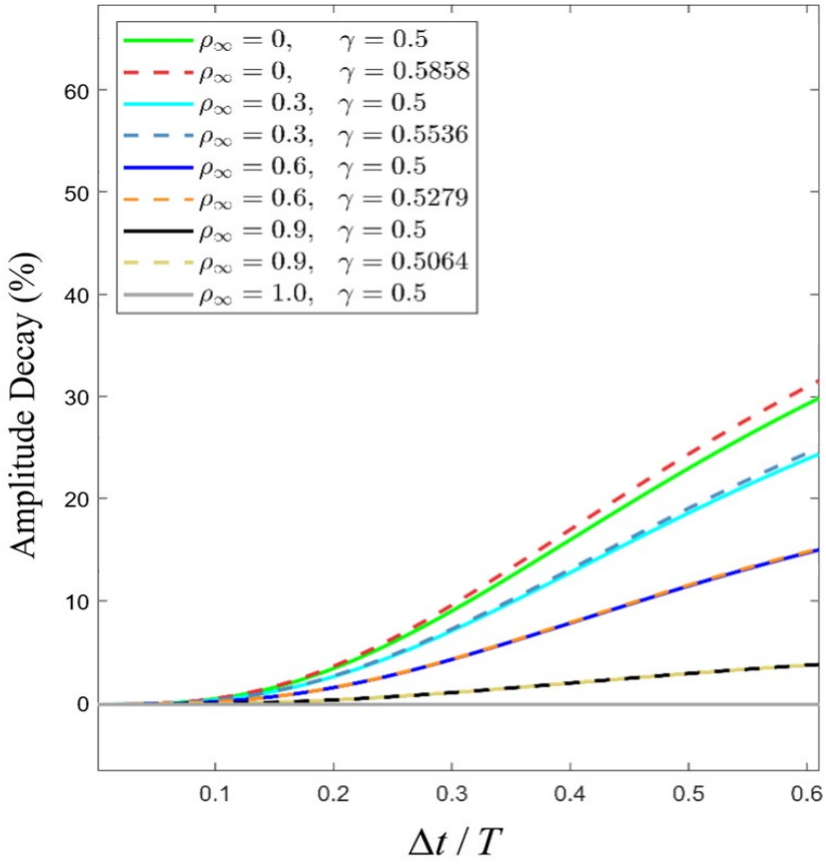
Let us next consider the implicit β_1 / β_2 -Bathe scheme.



(a) Spectral radius


$\rho_\infty = 0, \quad \gamma = 0.5$
$\rho_\infty = 0, \quad \gamma = 0.5858$
$\rho_\infty = 0.3, \quad \gamma = 0.5$
$\rho_\infty = 0.3, \quad \gamma = 0.5536$
$\rho_\infty = 0.6, \quad \gamma = 0.5$
$\rho_\infty = 0.6, \quad \gamma = 0.5279$
$\rho_\infty = 0.9, \quad \gamma = 0.5$
$\rho_\infty = 0.9, \quad \gamma = 0.5064$
$\rho_\infty = 1.0, \quad \gamma = 0.5$
Period Elongation (%)
50
45
40
35
30
25
20
15
10
5
0
-5
0.1
0.2
0.3
0.4
0.5
0.6
$\Delta t / T$

.



(b) Percentage period elongation and amplitude decay

Fig. 4.19 The ρ_∞-Bathe method when $\xi = 0$ with $\gamma = 0.5$ (solid lines) and $\gamma = \gamma_0$ (dashed lines) for various values of $\rho_\infty \in [0,1]$, G. Noh and K.J. Bathe [D].

The implicit β_1 / β_2-Bathe method was intoduced in M.M. Malakiyeh, S. Shojaee, and K.J. Bathe [A] and M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran, and K.J. Bathe [B] . In the first sub-step the Trapezoidal Rule as in the ρ_∞-Bathe method is used, see Eqs. (4.29) to (4.31).

For the second sub-step, instead of using Eqs. (4.33) and (4.34) a generalized trapezoidal rule is employed spanning over the three time points with the parameters β_1 and β_2

$$^{t+\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + (\gamma\Delta t)\left((1-\beta_1)^{t}\ddot{\mathbf{U}} + \beta_1{}^{t+\gamma\Delta t}\ddot{\mathbf{U}}\right) + ((1-\gamma)\Delta t)\left((1-\beta_2)^{t+\gamma\Delta t}\ddot{\mathbf{U}} + \beta_2{}^{t+\Delta t}\ddot{\mathbf{U}}\right) \tag{4.39}$$

$$^{t+\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + (\gamma\Delta t)\left((1-\beta_1)^{t}\dot{\mathbf{U}} + \beta_1{}^{t+\gamma\Delta t}\dot{\mathbf{U}}\right) + ((1-\gamma)\Delta t)\left((1-\beta_2)^{t+\gamma\Delta t}\dot{\mathbf{U}} + \beta_2{}^{t+\Delta t}\dot{\mathbf{U}}\right) \tag{4.40}$$

Considering Eqs. (4.39) and (4.40) when using $\beta_1 = \beta_2 = 1/2,$ we have the standard trapezoidal rule over each sub-step, and we can also show that the "standard Bathe method" of time integration given in Eqs. (4.23) to (4.28) is obtained with $\beta_1 = 1/3$, $\beta_2 = 2/3$ and $\gamma = 0.5$. To prove this relationship, we simply use these parameter values and Eqs.(4.30), (4.31), (4.39) and (4.40) to obtain the expressions for $^{t+\Delta t}\dot{\mathbf{U}}$ and $^{t+\Delta t}\ddot{\mathbf{U}}$ of the Bathe method.

Considering the essence of the procedure, we see that the method allocates in Eq. (4.39) different weights to the accelerations at times t, $t+\gamma\Delta t$ and $t+\Delta t$ for the calculation of the velocity vector, and similarly to the velocities in Eq. (4.40) for the displacement vector. We might note that the parameters β_1 and β_2 used in the above equations are in character totally different parameters to those used in the *explicit* β_1/β_2 -Bathe method presented in Section 4.2.2.

Mathematically, the method can be obtained from the ρ_∞-Bathe method by choosing the following parameter values

$$\begin{aligned} q_0 &= -\gamma(\beta_1 - 1) \\ q_1 &= (\beta_1 + \beta_2 - 1)\gamma - \beta_2 + 1 \end{aligned} \tag{4.41}$$

$$q_2 = -(\gamma - 1)\beta_2$$

and, as also earlier, $s_0 = q_0, s_1 = q_1, s_2 = q_2,$

To establish the above relationships, we use that the first sub-steps are identical and then compare only the governing equations of the second sub-steps of the ρ_∞-Bathe and the implicit β_1 / β_2-Bathe methods.

Considering the explicit schemes discussed in Section 4.2, we may note that such general relationship as given in Eqs. (4.41) is not present when comparing the explicit Noh-Bathe and β_1 / β_2-Bathe methods. The reason is that in the explicit Noh-Bathe method only the velocities are updated at the end of the sub-steps and in the explicit β_1 / β_2-Bathe method, both, the velocities and displacements are updated. Hence the same equations are only solved in these two procedures for the special case of $\beta_2 = 0$, $\beta_1 = 0.5$, and $q_0 = q_1 = 0.0$, and $q_2 = 0.5$.

The choice of $\rho_\infty = 0.0$ for the implicit β_1 / β_2-Bathe used in M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran, and K.J. Bathe [B] can be effective but is not necessary and indeed more accurate results may be obtained with another value of ρ_∞. The value of γ is given by the option used as shown in the two cases of structural vibrations and wave propagations discussed below.

Therefore using $\rho_\infty = 0.0$, the β_1 / β_2-Bathe method has the values of β_1 and β_2 to choose to obtain an effective first-order scheme for the analysis of wave propagations and to obtain an effective second-order scheme for the analysis of structural vibrations. This option and the simple structure of the method renders the scheme quite attractive. For ranges of recommended values for the parameters β_1 and β_2 we refer to M.M. Malakiyeh, S. Shojaee and K.J. Bathe [A] and M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B]. While the observations in these references are of value, they are rather broad for use when considering the many different types of dynamic problems that need to be solved accurately.

However, based on limited experience with the use of β_1 / β_2 -Bathe method, we can give the following recommendations for use of the implicit schemes that we discussed in this chapter:

For structural vibration problems, use a second-order method. The standard Bathe scheme is frequently effective, but the ρ_∞ -Bathe method using $\gamma = \gamma_0$ as defined in Eq. (4.38) and $\rho_\infty = 0.0$ can be more efficient because, for example, in linear analysis only one effective stiffness matrix is established and factorized.

Another choice is to use the implicit β_1 / β_2 -Bathe scheme for seond-order accuracy with the following values of the parameters:

$$\beta_1 = 0.75 - 0.25\sqrt{2},\ \beta_2 = 1/(3 - 4\beta_1),\ \text{and}\ \gamma = (\beta_2 - 1)/(2\beta_1 + \beta_2 - 2)$$

which gives to 2 digit accuracy $\beta_1 = 0.40$, $\beta_2 = 0.71$ and $\gamma = 0.59$ This choice of parameters corresponds also to using only one effective stiffness matrix in the step by step solution. Additional choices of values for the parameters are given in M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B]

For wave propgation problems, use the standard Bathe method with CFL = 1.0 or the ρ_∞ -Bathe scheme with $\gamma = \gamma_0$ as defined in Eq. (4.38) with $\rho_\infty = 0.0$ and CFL = 1.0, both methods show second-order accuracy. However, a first-order scheme can be more effective, in particular for the prediction of a long-time response and here the implicit β_1/β_2-Bathe method is attractive using, for example, $\gamma = 0.5$ and

$$1/3 < \beta_1 < 0.5 \text{ with } \beta_2 = 2\beta_1 .$$

The "best" values to use for β_1 and the CFL number depend on the problem considered and discretization used, in particular whether harmonic funcions are included, but the use of values $\beta_1 = 0.34$ to $\beta_1 = 0.43$ has resulted in accurate response solutions for different wave propagation problems. Here too, additional choices of values for the parameters are given in M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B], and see also K,T, Kim and K,J, Bathe [B].

It is particularly remarkable that when used as the first-order method, very accurate predictions have been obtained for wave propagation solutions with relatively long time durations, see M.M. Malakiyeh, S. Shojaee and K.J. Bathe [A] and KT Kim and KJ Bathe [B]. We demonstrate the performance of the method to some extent in Section 4.3.3.

Regarding the effect of physical damping, the recommended values of β_1 and β_2 given in M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B] have been derived assuming no physical damping, but further analysis shows that the values recommended in that reference result also in stable solutions when physical damping is included, see C. Lee, K.J. Bathe and G. Noh [A]. Since the method is related to the ρ_∞-Bathe scheme, the method can also directly be used to impose displacements and accurate solutions are obtained. Therefore, the implicit β_1 / β_2-Bathe method satisfies the patch test.

Considering the values for the parameters given above, we note that for the standard Bathe method no parameters are employed, and using the ρ_∞-Bathe scheme is also simple because only one parameter is used, namely ρ_∞. The implicit β_1 / β_2-Bathe method requires the setting of two parameter values, and

although the use of the above referred to values can be valuable, to reach in general the most accurate response prediction may require some numerical experimentation. Hence the use of the implicit β_1 / β_2 Bathe method is more involved.

We may note that in tackling the search for values of parameters for the "best" performance of the ρ_∞-Bathe scheme and β_1 / β_2-Bathe method, the use of machine learning procedures could be very effective to identify the optimal values of the parameters to use for an analysis, see Chapter 6 and T. Rabczuk and K.J. Bathe [A].

There are additional valuable observations regarding the ρ_∞-Bathe method which we briefly focus on next.

A first observation is that the Bathe method "contains" the Newmark method as a special case when the Newmark method is used with best parameters, see G. Noh and K.J..Bathe [E]. We recall that in the Newmark method we have two parameters, namely α and δ. For unconditional stability of the method we need to use

$$\alpha \geq 0.25(\delta + 0.5)^2 \quad ; \quad \delta \geq 0.5 \qquad (4.42)$$

The method is of second-order accuracy without numerical damping (but with period elongation) only when $\delta = 0.5$ and $\alpha = 1/4$ and is otherwise a dissipative first-order accurate method. When $\delta = 0.5$ and $\alpha = 1/4$ the Newmark method reduces to the Trapezoidal Rule.

For a comparison with the Bathe method we need to consider the use of the Newmark method

with two sub-steps per full time step. An analysis shows that the ρ_∞-Bathe method, based on Eqs. (4.33) to (4.36) when used with $\rho_\infty = 1$ and $\gamma = 1/2$ is identical to the Trapezoidal Rule.

Then considering the Newmark method with $\alpha = \kappa(\delta + 0.5)^2$, $\delta \geq 0.5$ and $0.25 \leq \kappa \leq 1.2$ we find that only the values κ = 1/4 and $0.5 < \delta \leq 1.5$ give "somewhat" acceptable shapes of amplitude decay, with larger κ values giving less acceptable curves. That is, only "somewhat" because the curves of amplitude decay curve opposite of what is desired. We would like a small amplitude decay at small values of time steps and then a rapid increase in amplitude decay as the time step Δt approaches $\Delta t / T \approx 0.3$, like seen in Fig. 4.19 (b). The curves obtained using the Newmark method do not show that curvature, see G Noh and KJ Bathe [E].

Hence the Newmark method is best used for dynamic solutions when we set $\alpha = 0.25(\delta + 0.5)^2$ with $\delta \geq 0.5$ and then the method is identical to the ρ_∞-Bathe method not using the relations (4.35) and (4.36) but using instead

$$q_0 = \frac{-4\delta^2+12\delta-1}{16\delta+8}, \quad q_1 = \frac{1}{2\delta+1}, \quad q_2 = \frac{2\delta+1}{8} \qquad (4.43)$$

and

$$\gamma = \frac{2\delta+1}{4} \qquad (4.44)$$

These relations hold for all $\delta \geq 1/2$.

With Eqs. (4.43) and (4.44), the ρ_∞-Bathe method is unconditionally stable and is a first-order accurate method except when $\delta = 1/2$, and thus $\gamma = 1/2$, because then the method is the two-step Trapezoidal Rule. We can also show that using $\alpha = 0.25(\delta + 0.5)^2$, that is, using the above defined κ equal to 1/4 , the

Newmark method provides the smallest numerical period elongation for all δ. Hence, for a given δ the period elongation is larger when κ is increased. This all means that the Newmark method with $\alpha = 0.25(\delta + 0.5)^2$ (that is, in its best use) is a special case of the ρ_∞-Bathe method when employed with the values given in Eqs. (4.43) and (4.44).

With Eq. (4.44), by further analysis we also find the useful relationship

$$\gamma = \frac{1-\sqrt{\rho_\infty}}{1-\rho_\infty}; \quad \text{and} \quad \gamma = 0.5 \ \text{if} \ \rho_\infty = 1 \tag{4.45}$$

Therefore, we see that the relations in Eqs. (4.43) to (4.45) are the parameter values to obtain a first order accurate ρ_∞-Bathe method with only one parameter, ρ_∞.

We may also observe that using Eqs. (4.43) and (4.44) together with Eq. (4.41) defining the parameters in the ρ_∞-Bathe method means that the parameters β_1 and β_2 in the implicit β_1 / β_2-Bathe scheme are also given, see also G Noh and KJ Bathe [E].

A second observation is that by choosing values of γ other than those we discussed above, namely outside the range $0 < \gamma < 1$ we can have third-order and even fourth-order accuracy of the solutions, see G. Noh and KJ Bathe [C].

A third observation is that by a judicious choice of γ and the loads at the times $t + \gamma\Delta t$, the usually second-order ρ_∞-Bathe method shows third-order accuracy, see S.-B. Kwon, K.J. Bathe and G. Noh [B]. This higher-order accuracy holds for displacements, velocities and the accelerations and is achieved by using a Newton-Cotes formula to apply the loads.

A fourth observation is that the patch test has been shown to be passed. In particular, the case of imposing displacements is properly solved. We may note here that, for example, the Trapezoidal Rule fails the test, see G Noh and KJ Bathe [F].

We present the patch test problem for imposing displacements, that we propose to solve, and the solution of that problem in Section 4.3.3.

Finally, a fifth observation is that the ρ_∞-Bathe method can directly be used also to solve the heat transfer equations in finite element analysis. Like in structural dynamics, we employ in heat transfer analyses in the first sub-step the trapezoidal rule for the equilibrium at time $t + \gamma\Delta t$

$$\mathbf{C}\,{}^{t+\gamma\Delta t}\dot{\boldsymbol{\theta}} + \mathbf{K}\,{}^{t+\gamma\Delta t}\boldsymbol{\theta} = {}^{t+\gamma\Delta t}\mathbf{Q} \tag{4.46}$$

$${}^{t+\gamma\Delta t}\boldsymbol{\theta} = {}^{t}\boldsymbol{\theta} + \frac{\gamma\Delta t}{2}({}^{t}\dot{\boldsymbol{\theta}} + {}^{t+\gamma\Delta t}\dot{\boldsymbol{\theta}}) \tag{4.47}$$

and in the second sub-step, we use the *Ansatz* of Eq. (4.33) for heat flow equilibrium at time $t + \Delta t$,

$$\mathbf{C}^{t+\Delta t}\dot{\boldsymbol{\theta}} + \mathbf{K}^{t+\Delta t}\boldsymbol{\theta} = {}^{t+\Delta t}\mathbf{Q} \tag{4.48}$$

$${}^{t+\Delta t}\boldsymbol{\theta} = {}^{t}\boldsymbol{\theta} + \Delta t(q_0\,{}^{t}\dot{\boldsymbol{\theta}} + q_1\,{}^{t+\gamma\Delta t}\dot{\boldsymbol{\theta}} + q_2\,{}^{t+\Delta t}\dot{\boldsymbol{\theta}}) \tag{4.49}$$

where $\mathbf{C}$ and $\mathbf{K}$ are the heat capacity and thermal conductivity matrices, respectively, $\mathbf{Q}$ is the vector of applied nodal heat loads , and $\boldsymbol{\theta}$ is the vector of nodal temperatures. We use the parameters q_0, q_1, q_2 as determined above, see Eqs. (4.35) to (4.38), to have the ρ_∞-Bathe method for the solution of the heat transfer equations.

Another important application of this time stepping procedure is the solution of the first-order transient Navier-Stokes equations. Furthermore, using this scheme together with the ρ_∞-Bathe method for the solution of the second-order equations in time, the two time integration methods (for the Navier-Stokes and the structural dynamics equations) can be employed effectively for the coupled solutions of fluid-structure interactions, see K.J. Bathe [A, L, O] and K.J. Bathe and H. Zhang [A] .

However, a heat transfer response is usually much smoother than a structural dynamics or wave propagation response and the good attributes of the ρ_∞-Bathe method (and the β_1/β_2-Bathe method) in comparison to the use of other methods may not be striking.

4.3.3 Illustrative solutions

The ρ_∞-Bathe method has been used in many applications and is also available in ADINA for wide applications, see for example R. Kroyer, K. Nilsson, and K.J. Bathe [A] and the references on ADINA given by Bentley Systems, Inc. ADINA [A].

We present in the following only a few applications to illustrate the use and performance of the time integration scheme. The solutions given are all reproduced from published papers that we refer to. Many more solutions are given in the references.

Solution of model with two springs – part of the patch test

We consider the solution of the simple model shown in Fig. 4.20. While "simple", this is a good problem to consider because the model reflects what is seen in practice and hence we use it as one of the problems in the patch test described in Section 4.1. The problem is used to identify whether the method imposes appropriate numerical damping in a structural vibration problem, see K. J. Bathe and G. Noh [A].

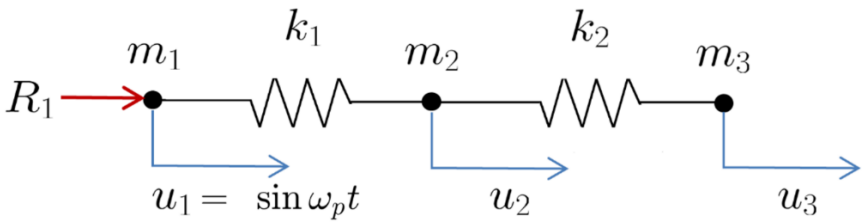


Fig. 4.20 Model problem of three degrees of freedom spring system,

$$k_1 = 1, k_2 = 10^7, m_1 = 0,\ m_2 = 1,\ m_3 = 1,\ \omega_p = 1.2,$$

K.J. Bathe and G. Noh [A] and G. Noh and K.J. Bathe [C].

The model consists of two springs, one of low stiffness, $k_1 = 1$ and one of very high stiffness. $k_2 = 10^{7.}$ A displacement is applied at the left end, two displacements u_2, u_3 and the reaction R_1 are to be solved for.

The two springs represent the essence of a complex finite element model, in which flexible parts (represented by k_1) and very stiff parts (represented by k_2) are present. The flexible parts could be due to thin plates or shells in bending and the stiff parts might be due to very stiff springs or beams merely used to impose constraints between degrees of freedom. This finite element model is to be solved using a time integration scheme with which the dynamic response of the flexible part is to be calculated accurately while a dynamic response of the stiff part is not sought (other than that the proper constraints are imposed on the flexible part).

In a mode superposition solution, the response within these stiff parts – a response that corresponds to very high artificial frequencies – would naturally not be included and we desire that the direct time integration "mimics" the mode superposition by directly filtering out this response. In addition, there should be no overshoot in the solution.

The spring system in Fig. 4.20 is governed by the following equations

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1+k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \\ 0 \end{bmatrix} \qquad (4.50)$$

As given in Fig. 4.20, for our study we use the stiffness and mass values given in Fig. 4.20 and prescribe the displacement at node 1 to be

$$u_1 = \sin \omega_p t \qquad (4.51)$$

with $\omega_p = 1.2$.

Since node 1 is subjected to the prescribed displacement over time, we rewrite Eq. (4.50) to solve only for the unknown displacements u_2 and u_3

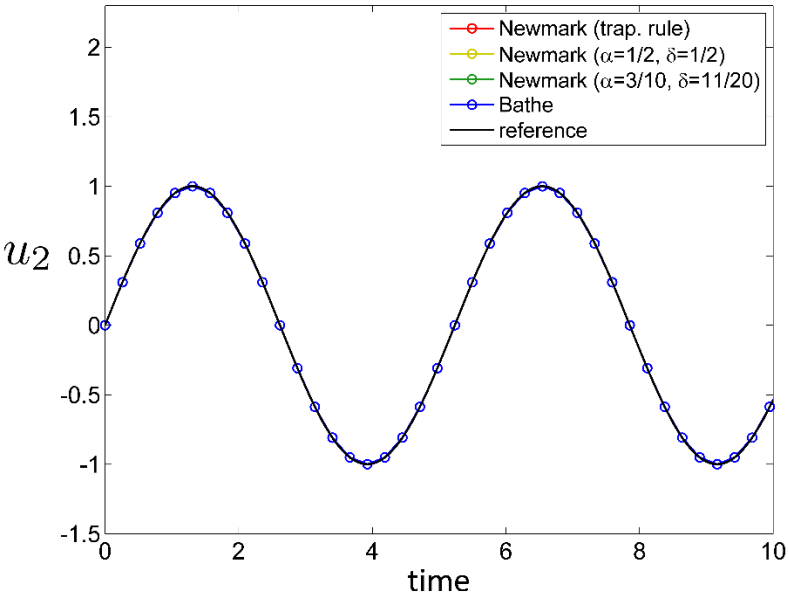
$$\begin{bmatrix} m_2 & 0 \\ 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{u}_2 \\ \ddot{u}_3 \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} k_1\, u_1 \\ 0 \end{bmatrix} \qquad (4.52)$$

after which the reaction is obtained from

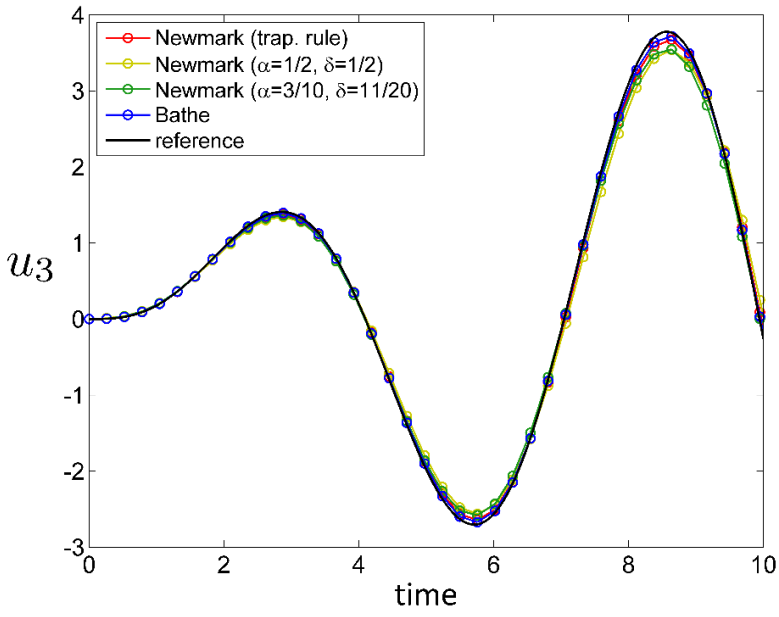
$$R_1 = m_1\, \ddot{u}_1 + k_1\, u_1 - k_1\, u_2 \qquad (4.53)$$

We solve the spring system using zero initial conditions for the displacements and velocities at nodes 2 and 3 (as is typically assumed in a complex multiple degrees of freedom structural analysis), and solve for the response over 10 seconds. For the solution we use the time stepping schemes for Eq. (4.52) and also calculate the reaction in Eq. (4.53). The time step used is $\Delta t = 0.2618$; hence we have $\Delta t \,/\, T_p = 0.05$, $\Delta t \,/\, T_1 = 0.0417$ and $\Delta t \,/\, T_2 = 131.76$, where $T_1 = 6.283$, $T_2 = 0.002$ are the natural periods of the system in Eq. (4.52) and $T_p = 5.236$ is the period of the prescribed motion at node 1.

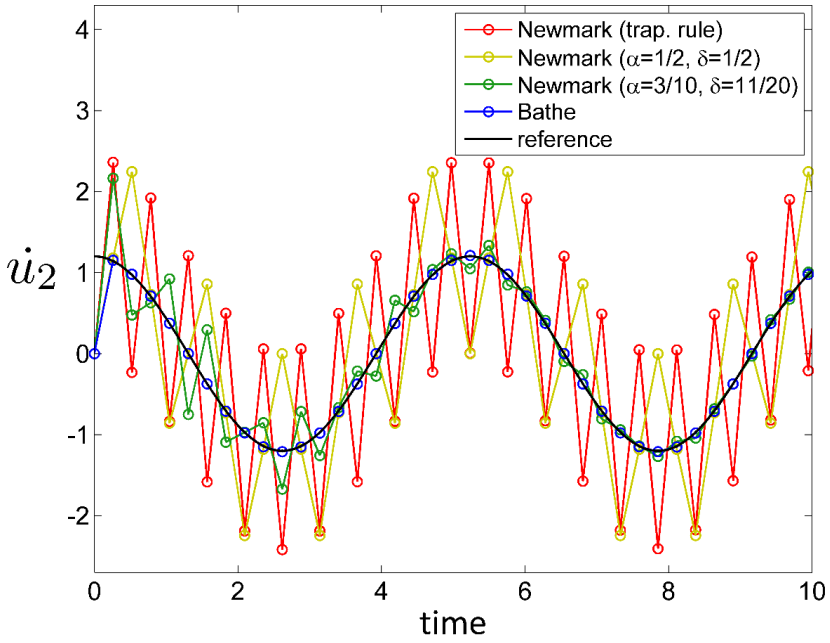
Figure 4.21 shows the response calculated with the Trapezoidal Rule, the Bathe method and the Newmark method with parameters α and δ to employ the Trapezoidal Rule and schemes to introduce some numerical damping.



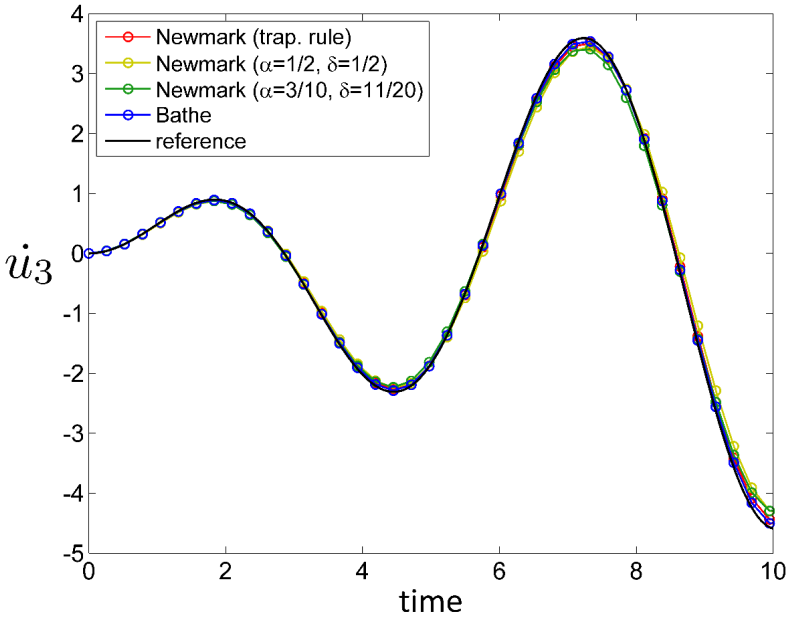
(a) Displacement of node 2 for various methods



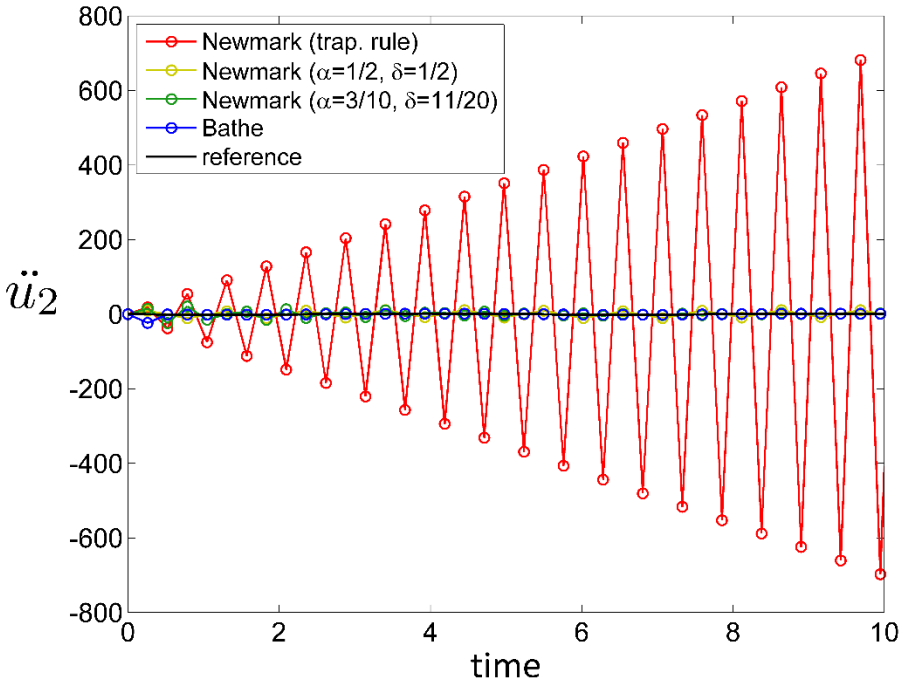
(b) Displacement of node 3 for various methods



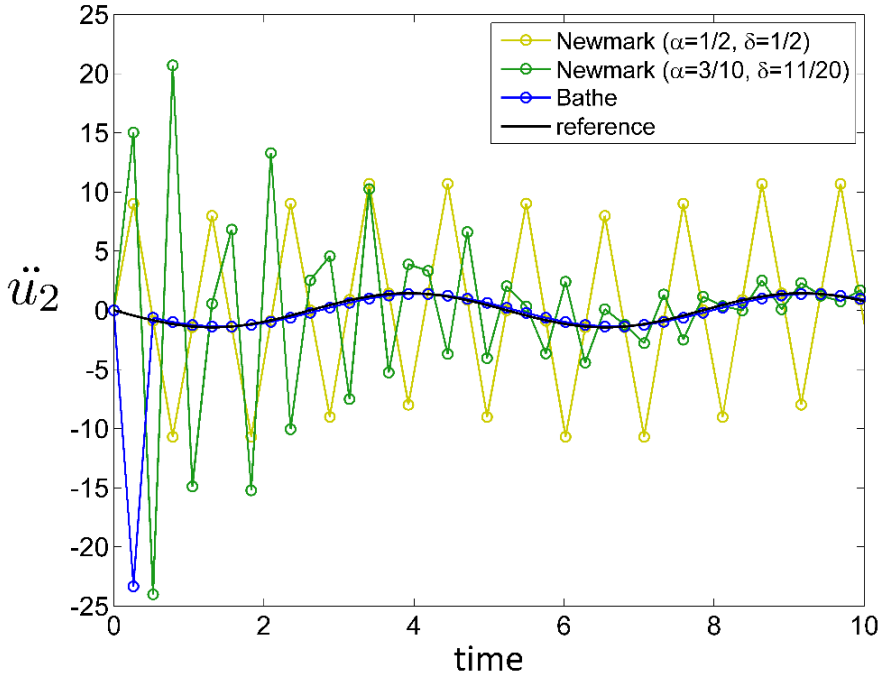
(c) Velocity of node 2 for various methods (the nonzero value at time = 0 is a consequence of the static correction)



(d) Velocity of node 3 for various methods

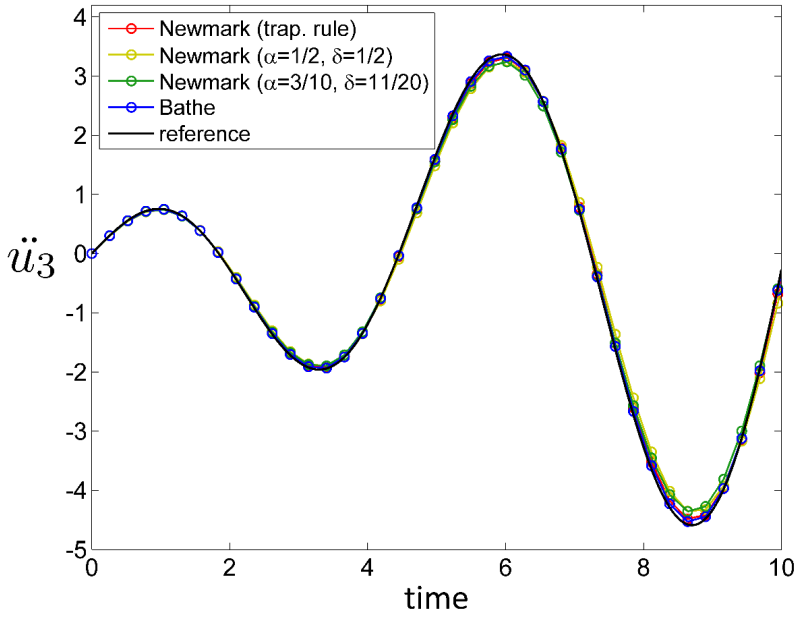


(e) Acceleration of node 2 for various methods



(f) Acceleration of node 2 for various methods, not using the Trapezoidal Rule.

For the first step in the Bathe method, the overshot is eliminated with good results throughout if we use only for the first sub-step of the first time step, $\alpha = 1; \delta = 3/4$.



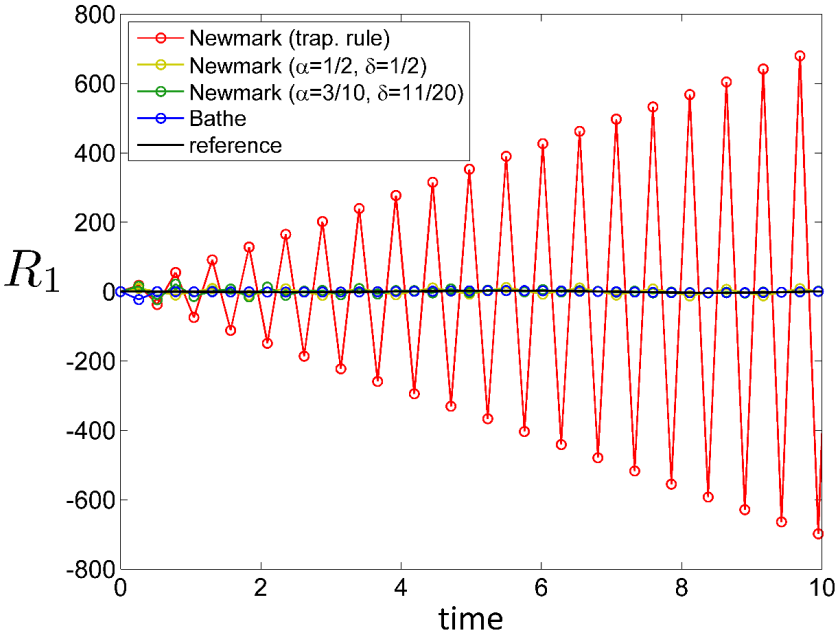
(g) Acceleration of node 3 for various methods.

Fig. 4.21 Solutions for the problem defined in Fig. 4.20.

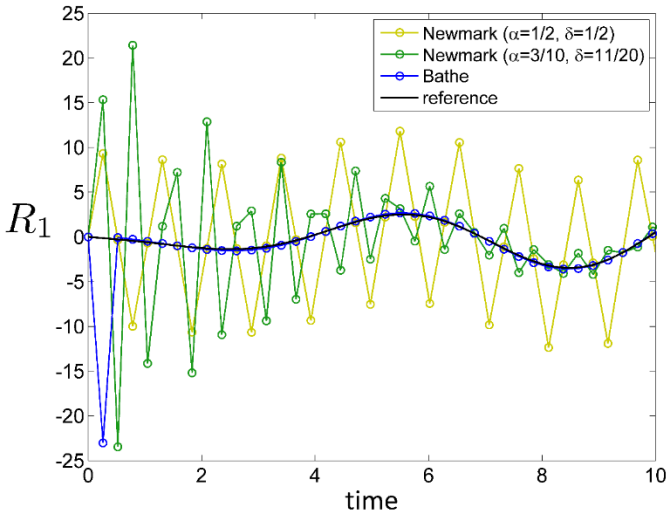
The results in Fig. 4.21 show that for all methods the displacements u_2 and u_3 are reasonably well predicted, but the velocity and acceleration at node 2 are only accurately calculated with the Bathe method. In particular, we should note that using the Trapezoidal Rule results in huge artificial oscillations resembling an instability for the acceleration at node 2.

Considering next the reaction at node 1, given in Fig. 4.22, we also see huge oscillations when the Trapezoidal Rule is used. To suppress this phenomenon, some damping needs to be introduced. In engineering practice, it may be physical damping when using the Trapezoidal Rule or the use of

different parameters using the Newmark method. Both approaches are not satisfying because appropriate physical damping, using e.g. Rayleigh damping with its parameters, or the Newmark method with appropriate values of its parameters for numerical damping need be selected. The “best” values of the parameters then depend on the problem to be solved and some experimentation with their values may be necessary.



(a) Reaction force at node 1 for various methods



(b)Reaction force at node 1 for various methods, not using the Trapezoidal Rule; in the Bathe method, the overshot is eliminated with good results throughout if we use only for the first sub-step of the first time step, $\alpha = 1;\ \delta = 3/4$.

Fig. 4.22 Predicted reaction force at node 1 of problem defined in Fig. 3.20

Imposing a prescribed displacement – part of the patch test

Figure 4.23 shows a very simple problem to be solved by the time integration scheme: a single truss (spring) element with two nodes, see G. Noh and K.J. Bathe [B]. At the left node, node 1, the varying displacement u_1 is imposed. The response at node 2 is to be calculated using the Trapezoidal Rule and the Bathe method. We consider two cases.

In case 1, we focus on what is typically done in practice, namely the velocity and acceleration response at node 1, both need to be calculated. The reason is that while the displacement is known, it is in practice not given analytically and hence the velocity and acceleration cannot be established analytically

either.

In case 2, we impose also the exact velocity and acceleration at node 1, because in this simple case, these values are known analytically. We are of course mostly interested in and focus on the case 1 results. Figures 4.24 to 4.26 give the calculated response, where "TR" refers to the Trapezoidal Rule and "w/ Ideal B.C." refers to the case 2 results.

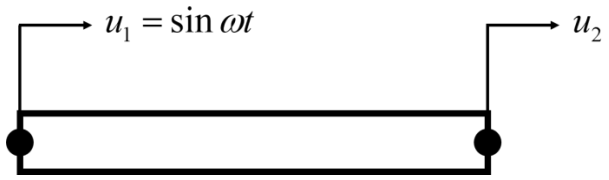


Fig. 4.23 A 2-node finite element under sinusoidal prescribed displacement; $\omega = 1.2$, G. Noh and K.J. Bathe [B].

Looking at the response of node 1 shown in Fig. 4.24, we see that for case 1 the calculated velocity and acceleration using the Trapezoidal Rule are very inaccurate. Indeed, the acceleration response is unstable although the method is unconditionally stable when only external loads are applied. The calculated response using the Bathe method is accurate, but for the first time step (only) we need to use $\alpha = 1$ and $\delta = 3/4$ for the first sub-step instead of the trapezoidal rule (like mentioned for the previous solution, see Figs. 4.21 and 4.22) .

Of course, for case 2, good results are obtained and are also shown in Fig. 4.24.

Considering node 2, see Fig. 4.25, we see that the response measured using the Bathe method is accurate, and using the Trapezoidal Rule the nodal displacement and velocity are reasonably predicted,

but in case 1 the acceleration is again in large error. The results are good, as expected, in case 2.

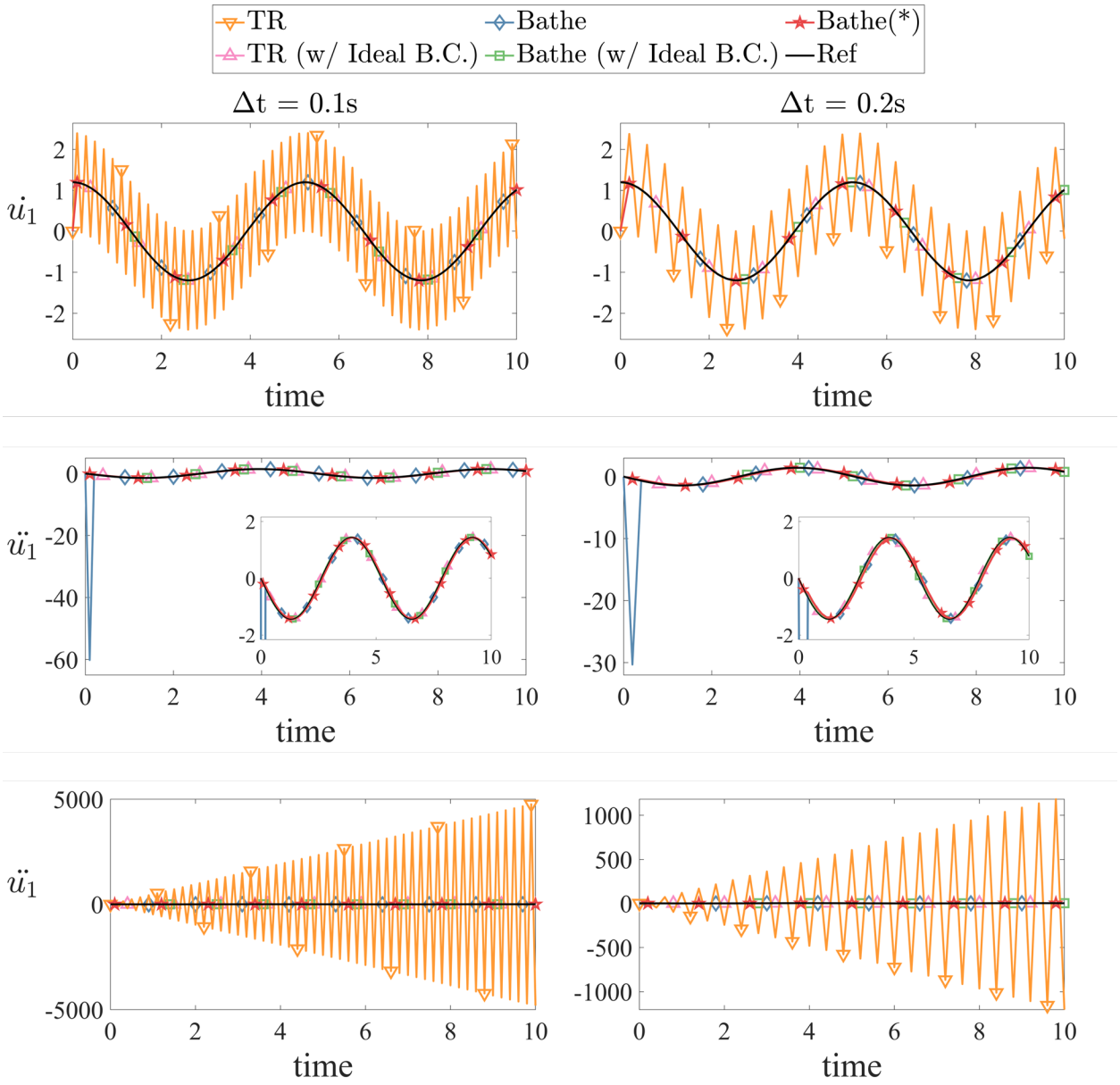


Fig. 4.24 Predictions of velocity and acceleration at node 1; "Bathe(*)" uses $\alpha = 1$ and $\delta = 3/4$ only for the first step, and $\alpha = 1/4$ and $\delta = 1/2$ otherwise; the results of the computed acceleration for "TR" are shown separately due to the different scale.

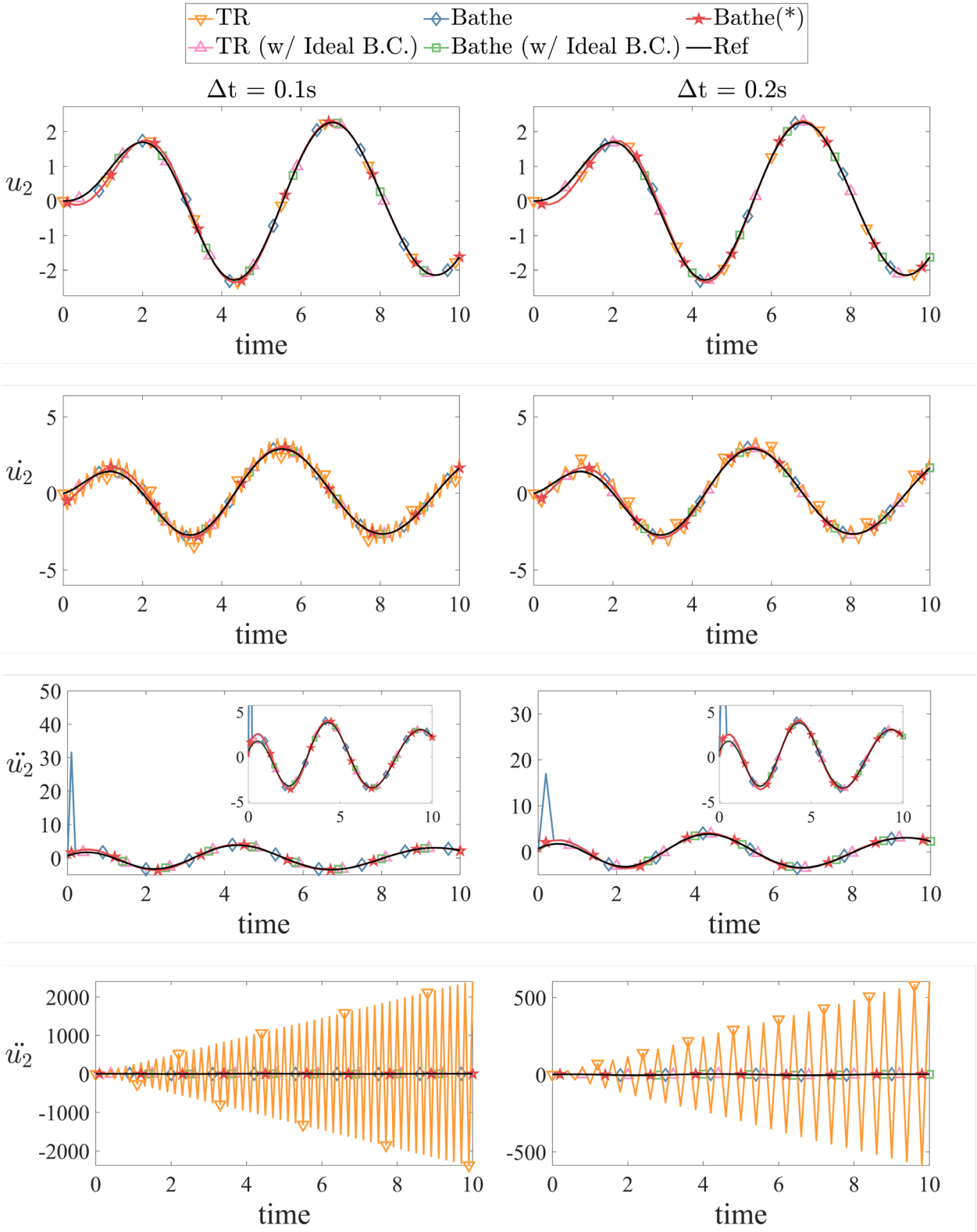

TR
Bathe
Bathe(*)
TR (w/ Ideal B.C.)
Bathe (w/ Ideal B.C.)
Ref
Δt = 0.1s
Δt = 0.2s
u_2
$\dot{u}_2$
$\ddot{u}_2$
$\ddot{u}_2$
time

Fig. 4.25 Predictions of displacement, velocity and acceleration at node 2; "Bathe(*)" uses $\alpha = 1$ and $\delta = 3/4$ only for the first sub-step, and $\alpha = 1/4$ and $\delta = 1/2$ otherwise; the results of the computed acceleration for "TR" are shown separately due to the different scale.

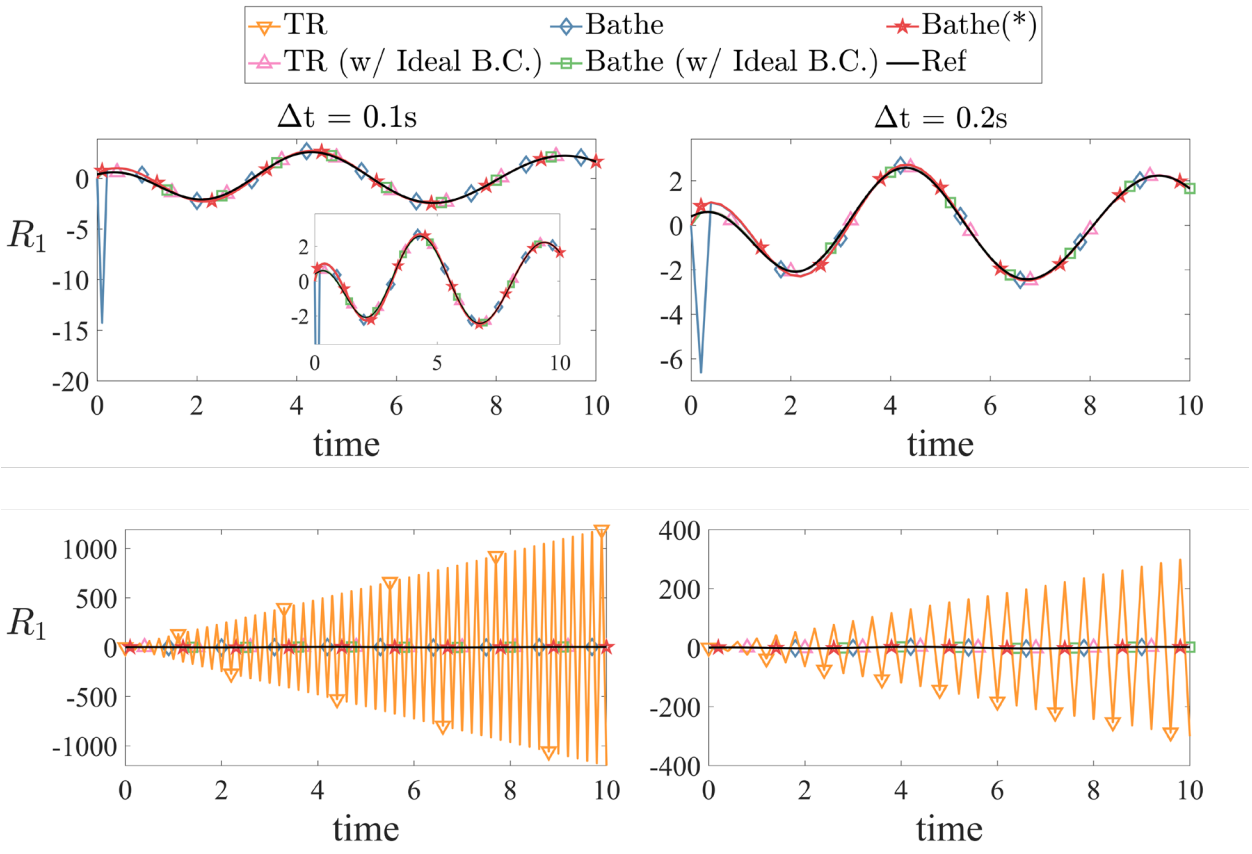


Fig. 4.26 Prediction of reaction force; "Bathe(*)" uses $\alpha = 1$ and $\delta = 3/4$ only for the first step, and $\alpha = 1/4$ and $\delta = 1/2$ otherwise; the results of the computed reaction for "TR" are shown separately due to the different scale.

For the results of the calculated reaction, see Fig. 4.26, the observations given above for the results in Figs. 4.24 and 4.25 are applicable.

A deep analysis and further results are given in G Noh and KJ Bathe [B] where it is also shown that the above observations are also seen when finite element systems with many degrees of freedom are solved.

The conclusion is therefore that if displacements are imposed in a transient direct integration solution it is important that the scheme passes the patch test including the case of imposing displacements.

A bi-material rod subjected to a step end load

We considered the solution of this problem already in Section 4.2.3 using explicit integration, see Figs. 4.12 to 4.14, and now solve the problem using implicit integration. Impact problems like this one are usually solved using explicit integration because a small time step is needed. However, it is also of interest to study the performance of an implicit direct integration scheme in the solution of wave propagations.

In the solution of wave propagation problems, we need to include some numerical damping to suppress spurious oscillations, see M. M. Malakiyeh, S. Shojaee, and K. J. Bathe [A], and G. Noh and K.J. Bathe [D]. Indeed, this problem is also a good problem to include in the patch test.

Here we use the implicit β_1 / β_2-Bathe method (or the ρ_∞-Bathe method if the parameters are converted using Eq. (4.41)). As in the explicit solution, we use a mesh of 1 x 800 equal size 4- node two-dimensional elements for discretizing the spatial domain and the time step is also given as in the explicit time integration solution.

Fig. 4.27 gives the calculated response with the selected parameters, CFL = 1.8519 and $\gamma = 0.5$. We see that the predicted response using the implicit time integration is quite accurate. The solution in Fig. 4.27 can be compared with the solutions obtained using the explicit Noh-Bathe and β_1 / β_2-Bathe methods in Section 4.2.3, see also M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A].

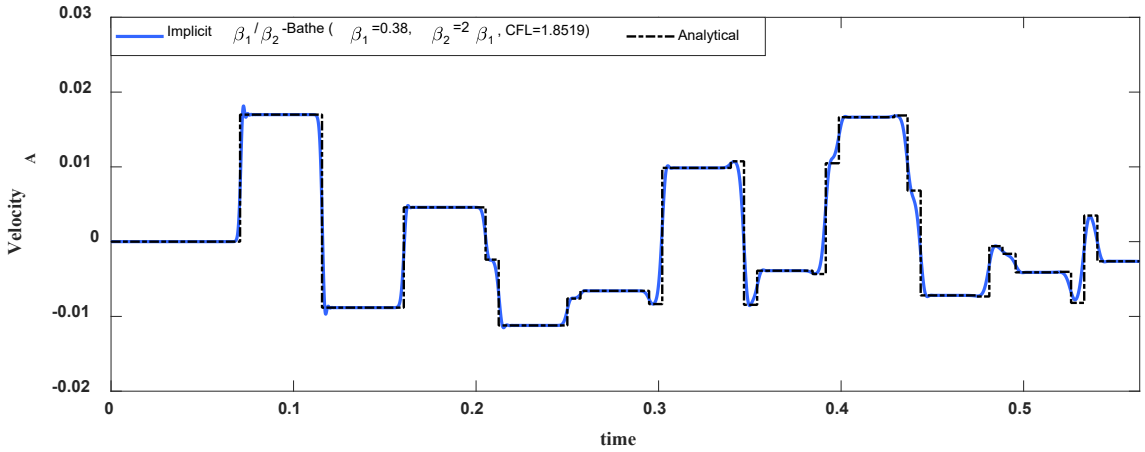


Fig. 4.27 Predicted velocity using the implicit β_1 / β_2-Bathe method at point A in the rod (see Fig. 4.7), M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A].

The pre-stressed square membrane subjected to an initial velocity over the central part

We consider the membrane problem already solved in Section 4.2.3, see Fig. 4.15. The same geometric and material conditions and finite element discretization are used, and as before the velocity at the center point is to be predicted. Fig. 4.28 shows the response obtained with the implicit β_1/β_2-Bathe method, which can be compared with the response calculated using the explicit Noh-Bathe and explicit β_1/β_2-Bathe methods, see also M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [A].

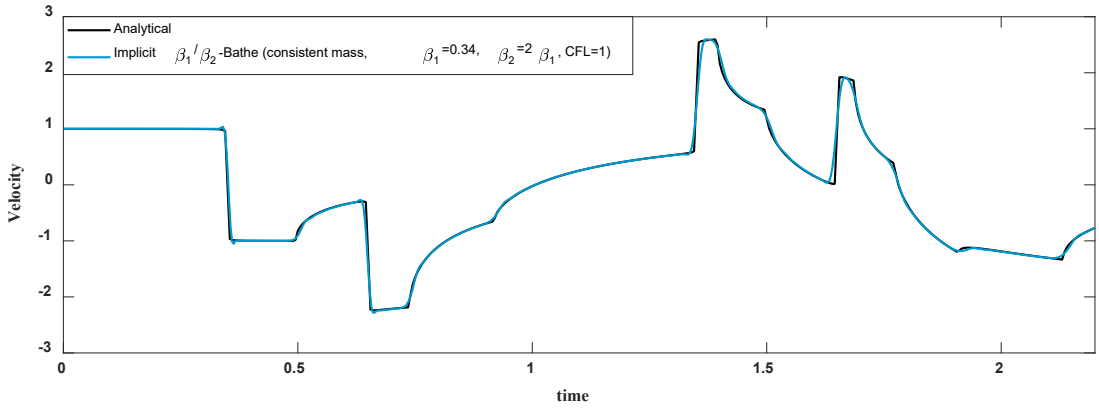


Fig. 4.28 Predicted velocity at center point using the implicit β_1/β_2 -Bathe method, consistent mass matrix, CFL = 1, $\gamma = 0.5$, M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe [A].

We see that the predicted responses using the implicit β_1/β_2 -Bathe method in Figs. 4.27 and 4.28 are very accurate. However, the values of the integration parameters used are different and prior solutions with the use of the method, or some numerical experimentation, is needed to identify good values of the parameters to use.

For many more solutions of various problems of transient analyses using the implicit schemes that we discussed above, we refer to K.J. Bathe and G. Noh [A], G. Noh and K.J. Bathe [B, C, D, E], S.B. Kwon, K. J. Bathe and G. Noh [A, B], C. Lee, K.J. Bathe and G. Noh [A], M. M. Malakiyeh, S. Shojaee and K.J. Bathe [A], M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe [B] and R. Kroyer, K. Nilsson, and K.J. Bathe [A], and Z. Kazancı and K.J. Bathe [A]. Furthermore, see also K.T. Kim and KJ Bathe [A] where the overlapping finite elements are used with the implicit β_1 / β_2 -Bathe method and good results have been obtained (see also Section 3.3.4).

In addition, there are other explicit and implicit solution methods using sub-stepping, and these may be closely related to the methods discussed above, but as mentioned in the introduction to this chapter, the objective in this chapter was not to survey this field of new developments but only to introduce the basic idea of using sub-steps and present some effective solution schemes. We can expect that many more developments will be forthcoming, but for each development, it is important to identify whether the patch test is passed (see Section 4.1) and that sufficiently difficult problems are solved to show the strengths of the proposed method.

4.4 Exercises

We suggest these Exercises to obtain further insight into the solution schemes discussed in this chapter. To complete an Exercise, the material given in KJ Bathe [A] may need to be known.

4.4.1

Prove that the Noh-Bathe explicit time integration scheme using Eqs. (4.10) and (4.11) is a second-order accurate solution scheme.

4.4.2

Show that the explicit β_1 / β_2 -Bathe time integration scheme is first-order accurate when using $\beta_1 \neq 0.5$ and second-order accurate when $\beta_1 = 0.5$.

4.4.3

Prove that the value of γ_0 given in Eq. (4.38) results in using the same effective stiffness matrux for the two sub-steps in the ρ_∞-Bathe time integration scheme. Consider first the value $\rho_\infty = 0.0$ and then the general case of Eq. (4.38).

4.4.4

Consider the ρ_∞-Bathe method using γ = 0.5 and ρ_∞ = 1.0. Show that with these values of parameters, the ρ_∞-Bathe method of time integration reduces to the use of the Trapezoidal Rule for each sub-step.

4.4.5

Consider the ρ_∞-Bathe time integration scheme with the parameters $\gamma = 0.5$, q_0 = 1/4, q_1 = 1/2, q_2 = 1/4 and show that the use of these parameters results in the use of the Trapezoidal Rule for each sub-step.

4.4.6

Show that the standard Bathe time integration scheme in Eqs. (4.23) to (4.28) is obtained when using the parameters $\gamma = 0.5$, $q_0 = q_1 = q_2$ = 1/3 in the ρ_∞-Bathe time integration scheme.

4.4.7

Consider the ρ_∞-Bathe method and show that with the parameters

$$q_0 = -\gamma(\beta_1 - 1)$$

$$q_1 = (\beta_1 + \beta_2 - 1)\gamma - \beta_2 + 1$$

$$q_2 = -(\gamma - 1)\beta_2$$

and $s_0 = q_0, s_1 = q_1, s_2 = q_2$

the implicit β_1 / β_2 -Bathe method is obtained.

4.4.8

Show that the standard Bathe time integration scheme in Eqs. (4.23) to (4.28) is obtained when using the parameters $\gamma = 0.5$, β_1 = 1/3 and β_2 = 2/3 in the implicit β_1 / β_2 -Bathe method.

4.4.9

Consider the Newmark method when optimally used as a special case of the ρ_∞-Bathe method and identify the corresponding values of parameters of the implicit β_1 / β_2 - Bathe method as a function of δ used in Eqs. (4.43) and (4.44).

.

4.4.10

Establish the detailed step-by-step equations to be used in transient heat tranfer analysis for use of the ρ_∞ -Bathe method and establish a table of the scheme.

5. The Bathe Subspace Iteration Procedure

In engineering, probably the most common solution procedures to calculate frequencies and mode shapes of large finite element systems are the Lanczos method, see K.J. Bathe [A] and the Bathe subspace iteration method, see K.J. Bathe [A, J, H, K].

The Lanczos method proposed prior to the development of the modern finite element methods, see C. Lanczos [A], has been much researched, and while initially unstable, the method is now considered a robust, reliable and effective solution scheme. However, when using the Lanczos method, care needs to be taken to check that multiple eigenvalues are correctly predicted. The Bathe method is also robust, reliable and efficient, and the use of the method can be advantageous. In many programs both methods are offered for solutions, see e.g. Bentley Systems, Inc., ADINA manuals [A].

In the following sections we refer to the original Bathe subspace iteration and then include a scheme recently developed to significantly accelerate the original method, see KT Kim and KJ Bathe [B].

5.1 Basic equations and considerations

We want to solve the generalized symmetric eigenvalue problem

$$\mathbf{K}\boldsymbol{\varphi} = \lambda\mathbf{M}\boldsymbol{\varphi} \tag{5.1}$$

where $\mathbf{K}$ and $\mathbf{M}$ are the stiffness matrix and the mass matrix of a finite element system with n degrees of freedom. The matrix $\mathbf{K}$ is assumed to be positive definite which can always be achieved by shifting, and a consistent mass matrix is always positive definite. A diagonal mass matrix can also be used, but of course only the finite eigenvalues can be solved for and hence the value q discussed below must satisfy $q \le n - r$, where r equals the number of zero diagonal entries in **M**, see K.J. Bathe [A].

In the solution scheme, we seek the smallest p eigenvalues and corresponding eigenvectors $(\lambda_i, \boldsymbol{\varphi}_i); i = 1,...p$, with

$$0 < \lambda_1 \le \lambda_2 \le .... \le \lambda_p, \tag{5.2}$$

which satisfy

$$\mathbf{K}\boldsymbol{\varphi}_i = \lambda_i \mathbf{M}\boldsymbol{\varphi}_i \tag{5.3}$$

and

$$\boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{M}\boldsymbol{\varphi}_j = \delta_{ij}, \tag{5.4}$$

$$\boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{K}\boldsymbol{\varphi}_j = \lambda_i \delta_{ij} \tag{5.5}$$

where δ_{ij} is the Kronecker delta. In the analysis of structures, most commonly the smallest eigenvalues (frequencies squared) are required on which we focus here. To calculate eigenvalues within an interval we apply a matrix shift and then proceed similarly.

Prior to the subspace iterations, we establish good starting vectors, see KJ Bathe [A].
Unless starting vectors are given by some other procedure, see below, we use a simple scheme to construct q linearly independent vectors in $\mathbf{MX}_0$, with $q > p$. These vectors are obtained using unit vectors that excite in **M** the degrees of freedom of the maximum values m_{ii}/k_{ii} and a random vector, with usually $q = \max\{2p, p+8\}$. With these linear independent vectors in $\mathbf{MX}_0$, we use the equations of the basic subspace iteration method (Eqs. (5.17) to (5.20) below) to **M**-orthonormalize the vectors to obtain $\mathbf{X}_1$ corresponding to the q-dimensional subspace E_1.

However, it can be much more effective to use starting vectors given by some other procedure. For example, if an optimization of a design is pursued, the eigenvectors of a finite element model of a previous

design may be good vectors to start the next frequency solution for the improved design. Also, if an approximate solution of the eigenvectors, for example using a component mode synthesis, has been obtained and must be improved, the vectors of the approximate solution can be good starting vectors, see K.J. Bathe and J. Dong [A]. The ability to use such starting vectors, to obtain convergence in a few iterations, is a particular strength of the iteration scheme for engineering analyses.

Another important strength is that the calculation procedure has no difficulties with the solution of eigenvalues of multiplicity *m*. The procedure is iterating with a set of vectors that span a subspace, and once the subspace has converged to the least dominant subspace sought of dimension *p*, all *p* eigenvectors sought, and associated eigenvalues, have converged. This approach renders the procedure very stable, including for the solution of eigenvalues with multiplicity. If the eigenvalue λ_r has multiplicity *m*, meaning

$$0 < \lambda_1 \leq \lambda_2 \ldots \leq \lambda_r = \lambda_{r+1} = \ldots. = \lambda_{r+m-1} \leq \ldots \leq \lambda_p$$

then the *m*-dimensional subspace corresponding to that eigenvalue will converge like all other iteration vectors, of course assuming that the number of iteration vectors used, *q*, is sufficiently large. Also, all eigenvectors will be **K-** and **M**-orthogonal (no spurious vectors will be found). Other solution schemes can have difficulty in solving for eigenvalues of multiplicity and may also give spurious eigenvectors.

5.2 The subspace iteration procedure – with an acceleration

For the presentation of the solution scheme, we need to establish the basic procedure, the convergence of the method and give some illustrative example solutions. The novelty compared to the procedure in K.J. Bathe [A] is that we include an acceleration process in the iterations which can significantly decrease the solution time.

5.2.1 The equations solved

The subspace iteration proceeds as described in K.J. Bathe [A, G, K], except that we now include a procedure to accelerate the convergence, see K.T. Kim and K.J. Bathe [B].

For the iteration $k = 1, 2, ...,$ we partition the iteration vectors which span the subspace E_k

$$\mathbf{X}_k = \left[\mathbf{\Phi}_k, \mathbf{X}_k^a, \mathbf{X}_k^b \right] \tag{5.6}$$

where the vectors are ordered according to the eigenvalue approximations corresponding to the vectors from the smallest to the largest approximate eigenvalues, and

$\mathbf{\Phi}_k$ stores the p_k vectors which have already converged to the required tolerance in the previous iterations. Hence, we have initially $p_1 = 0$. The rest of the iteration vectors in $\mathbf{X}_k$ are partitioned into $\mathbf{X}_k^a$ and $\mathbf{X}_k^b$. Both matrices are of order $n \times r_k$, that is, we use $r_k = (q - p_k)/2$. Of course, a different partitioning could be used but the basic approach would not change.

A strength of the procedure used for the improvement in the solution time is that the scheme is conceptually quite simple, like the original subspace iteration procedure. Let E_k^a be the subspace spanned by the vectors in $\mathbf{X}_k^a$, and use the same notation for the other subspaces, like we mentioned already above E_k for the vectors in $\mathbf{X}_k$. Then we can summarize the scheme as follows:

Step 1: Perform one inverse iteration using only $\mathbf{X}_k^a$ to obtain $\overline{\mathbf{X}}_{k+1}^a$ turning E_k^a to E_{k+1}^a. Extract from E_{k+1}^a the subspace $\overline{T}_{k+1}^a$ of dimension $\overline{t}_k$ that has "turned significantly" from E_k^a. The subspace $\overline{T}_{k+1}^a$ is spanned by $\overline{t}_k$ vectors given in $\overline{\mathbf{X}}_{k+1}^a$.

Step 2: Replace the least dominant subspace of dimension $\overline{t}_k$ of E_k^b given by the last $\overline{t}_k$ vectors in $\mathbf{X}_k^b$ by vectors calculated using the subspace $\overline{T}_{k+1}^a$ established in *Step 1*.

Thereafter, the subspace iterations are continued as in the original algorithm. Hence the only difference is that we do not use all of the usual vectors in $\mathbf{X}_k^b$ in the next iteration (as in the original subspace iteration) but replace $\overline{t}_k$ vectors to include a subspace established from $\overline{t}_k$ vectors that have undergone significant turning to reach $\overline{\mathbf{X}}_{k+1}^a$.

Therefore, overall, in essence, we perform one simultaneous vector iteration on the iteration vectors $\mathbf{X}_k^a$, the r_k vectors closest to convergence in $\mathbf{X}_k$ (with the vectors in $\mathbf{\Phi}_k$ already converged) to obtain $\overline{\mathbf{X}}_{k+1}^a$, then extract a $\overline{t}_k$ - dimensional subspace ($\overline{T}_{k+1}^a$) spanned by $\overline{t}_k$ vectors in $\overline{\mathbf{X}}_{k+1}^a$ (that have turned significantly) and use $\overline{T}_{k+1}^a$ to calculate a $\overline{t}_k$ - dimensional subspace to replace the subspace farthest from convergence in $\mathbf{X}_k$. This replacement can much improve the overall effectiveness of the subspace iteration procedure.

We next give the detailed equations used.

Establish which vectors to use for the subspace $\bar{T}_{k+1}^a$

As summarized above we solve

$$\mathbf{K}\overline{\mathbf{X}}_{\boldsymbol{k+1}}^{\boldsymbol{a}} = \mathbf{M}\mathbf{X}_k^a \tag{5.7}$$

and examine the vectors in $\overline{\mathbf{X}}_{k+1}^a$ to identify which vectors may be used to establish good "turning vectors" that accelerate the solution.

To identify the amount of turning each iteration vector in $\overline{\mathbf{X}}_{k+1}^a$ has undergone, we use

for $i = r_k,\ r_k - 1, \ldots, 1$ with $t_k = 0$

$$\hat{\mathbf{x}}_i = \overline{\mathbf{x}}_i^{a(k+1)} - \mathbf{X}_k \left((\mathbf{X}_\mathbf{k})^T \mathbf{M} \overline{\mathbf{x}}_i^{a(k+1)} \right) - \sum_{j=1}^{t_k} \mathbf{u}_j \left(\mathbf{u}_j^T \mathbf{M} \overline{\mathbf{x}}_i^{a(k+1)} \right) \tag{5.8}$$

where $\overline{\mathbf{x}}_i^{a(k+1)}$ is the i th column vector in $\overline{\mathbf{X}}_{k+1}^{a}$ and the last term is only included if $t_k \geq 1$. Here we use that the vectors in $\mathbf{X}_k$ are all **M**-orthonormalized. Hence in Eq. (5.8) we extract from the current iteration vector $\overline{\mathbf{x}}_i^{a(k+1)}$ the vector $\hat{\mathbf{x}}_i$ which is **M**-orthogonal to *all* vectors in $\mathbf{X}_k$ and *all* vectors $\mathbf{u}_j, j = 1, \ldots t_k$. Since convergence is reached fastest for the lower indexed vectors, and these will likely turn the least, we use in Eq. (5.8) the vectors in the order $i = r_k, \ldots, 1$. The procedure in Eq. (5.8) is the standard Gram-Schmidt process, see KJ Bathe [A].

Next, we evaluate whether the turning of $\overline{\mathbf{x}}_i^{a(k+1)}$ from E_k and the subspace spanned by the vectors $\mathbf{u}_j, j = 1, \ldots t_k$ was significant. If not, then the vector $\hat{\mathbf{x}}_i$ will be close to a null vector, compared to $\overline{\mathbf{x}}_i^{a(k+1)}$. Hence, we calculate

$$\alpha_i = \frac{\hat{\mathbf{x}}_i^T \mathbf{M} \hat{\mathbf{x}}_i}{\left(\overline{\mathbf{x}}_i^{a(k+1)}\right)^T \mathbf{M} \overline{\mathbf{x}}_i^{a(k+1)}} \tag{5.9}$$

and use a tolerance $tolt$ in the determination of the amount of turning. If $\alpha_i \leq tolt$, the turning has been too small for our purposes, and we go to the next value of i. If $\alpha_i > tolt$ we calculate

$$t_k = t_k + 1, \tag{5.10}$$

scale the vector $\hat{\mathbf{x}}_i$ to the **M**-orthogonalized unit length for the further use in Eq. (5.8) and store $\overline{\mathbf{x}}_i^{a(k+1)}$ in $\mathbf{v}_{t_k}$ for later usage

$$\mathbf{u}_{t_k} = \frac{\hat{\mathbf{x}}_i}{\sqrt{\hat{\mathbf{x}}_i^T \mathbf{M} \hat{\mathbf{x}}_i}} \tag{5.11}$$

$$\mathbf{v}_{t_k} = \overline{\mathbf{x}}_i^{a(k+1)} \tag{5.12}$$

Thereafter, we proceed to the next value of *i* until the search for the vectors that underwent good turning is completed, that is, the last value of *i* was $i = 1$. Let the last value of t_k reached be $\bar{t}_k$, then we have the vectors $\mathbf{v}_i$ with $i = 1,2,\ldots,\bar{t}_k$ established by Eq. (5.12), and these span the subspace $\bar{T}_{k+1}^a$ which is used to accelerate the solution procedure.

Establish a new more effective subspace $\tilde{E}_k^b$ ***to replace*** E_k^b

Given the subspace $\bar{T}_{k+1}^a$ and the space spanned by E_k^b , we have together a subspace of dimension $(r_k + \bar{t}_k)$ but we only want to continue working with a subspace of dimension r_k.

We achieve this aim by constructing $\widetilde{\mathbf{X}}_k^b = \left[\tilde{\mathbf{x}}_1^{b(k)},\ldots,\tilde{\mathbf{x}}_{r_k-\bar{t}_k}^{b(k)},\tilde{\mathbf{x}}_{r_k-\bar{t}_k+1}^{b(k)},\ldots,\tilde{\mathbf{x}}_{r_k}^{b(k)}\right]$ where $\tilde{\mathbf{x}}_i^{b(k)} = \mathbf{x}_i^{b(k)}$ for $i = 1,\ldots,r_k - \bar{t}_k$, that is, the first $r_k - \bar{t}_k$ column vectors in $\widetilde{\mathbf{X}}_k^b$ are those of $\mathbf{X}_k^b$. Hence $\widetilde{\mathbf{X}}_k^b$ contains vectors from the previous subspace iteration, not used in Eq. (5.7), and additional $\bar{t}_k$ vectors that we still need to determine.

To obtain the vectors $\tilde{\mathbf{x}}_{r_k-\bar{t}_k+1}^{b(k)},\ldots,\tilde{\mathbf{x}}_{r_k}^{b(k)}$ that satisfy in $\widetilde{\mathbf{X}}_k^b$ the condition $\widetilde{\mathbf{X}}_k^{b^T}\mathbf{M}\widetilde{\mathbf{X}}_k^b = \mathbf{I}$ (as the vectors in $\mathbf{X}_k^b$ satisfy this condition) and that are **M**-orthogonal to $\mathbf{X}_k^a$ and $\mathbf{\Phi}_k$, we proceed as follows for $i = 1,\ldots,\bar{t}_k$

$$\tilde{\mathbf{x}}_i = \mathbf{v}_i - \mathbf{X}_k^a((\mathbf{X}_k^a)^T\mathbf{M}\mathbf{v}_i) - \mathbf{\Phi}_k(\mathbf{\Phi}_k^T\mathbf{M}\mathbf{v}_i) - \sum_{j=1}^{r_k-\bar{t}_k+i-1}\tilde{\mathbf{x}}_j^{b(k)}\left(\left(\tilde{\mathbf{x}}_j^{b(k)}\right)^T\mathbf{M}\mathbf{v}_i\right) \tag{5.13}$$

$$\tilde{\mathbf{x}}_{r_k-\bar{t}_k+i}^{b(k)} = \frac{\tilde{\mathbf{x}}_i}{\sqrt{\tilde{\mathbf{x}}_i^T\mathbf{M}\tilde{\mathbf{x}}_i}}. \tag{5.14}$$

Here we may recall that the superscript *(k)* on the vectors denotes the current subspace iteration, and hence does not change in Eqs. (5.13) and (5.14) (and the same holds for the subscript *k* on the matrices in Eq (5.13)).

We call the vectors established in Eqs. (5.13) and (5.14) the "forward turning vectors". We note that the vector $\tilde{\mathbf{x}}_i$ is a linear combination of the already known vectors (to which $\tilde{\mathbf{x}}_i$ has been orthogonalized in Eq. (5.13)) and the $\overline{\mathbf{x}}_i^{a(k+1)}$ used in $\mathbf{v}_i$. Hence when $\tilde{\mathbf{x}}_i$ (normalized as given in Eq. (5.14)) is used in the next subspace iteration also $\overline{\mathbf{x}}_i^{a(k+1)}$ is implicitly employed and this accelerates the solution procedure.

With Eqs. (5.13) and (5.14) completed for all $\mathbf{v}_i$, the vectors in Eq. (5.14) to accelerate the solution have replaced $\bar{t}_k$ vectors in $\mathbf{X}_k^b$ to reach $\widetilde{\mathbf{X}}_k^b$ and we have the desired result

$$\left[\mathbf{\Phi}_k, \mathbf{X}_k^a, \widetilde{\mathbf{X}}_k^b\right]^T \mathbf{M}\left[\mathbf{\Phi}_k, \mathbf{X}_k^a, \widetilde{\mathbf{X}}_k^b\right] = \mathbf{I}$$

The procedure used in Eq. (5.13) is also the Gram-Schmidt process and the result is important for the stability of the subspace iteration procedure.

Proceed with the new vectors in $\widetilde{\mathbf{X}}_k^b$ instead of using $\mathbf{X}_k^b$

The algorithm now proceeds as in the original subspace iteration, see KJ Bathe [A].

Evaluate $\overline{\mathbf{X}}_{k+1}^b$ from

$$\mathbf{K}\overline{\mathbf{X}}_{k+1}^b = \mathbf{M}\widetilde{\mathbf{X}}_k^b \tag{5.15}$$

where we now have in $\overline{\mathbf{X}}_{k+1}^b$ the effect of the turning vectors for a faster solution. If no

turning vectors have been identified, we have $\widetilde{\mathbf{X}}_k^b = \mathbf{X}_k^b$ and the iteration scheme remains simply the original subspace iteration procedure. Such cases are encountered when the subspace iteration has almost converged, that is, the vectors in $\overline{\mathbf{X}}_{k+1}^a$ hardly turned from $\boldsymbol{X}_k^a$ and $\bar{t}_k = 0$.

In the next steps, we simply orthogonalize, as in the basic subspace iteration, the new vectors in $\overline{\mathbf{X}}_{k+1}$.

Construct $\overline{\mathbf{X}}_{k+1}$ with the column vectors spanning E_{k+1}

$$\overline{\mathbf{X}}_{k+1} = \left[\boldsymbol{\Phi}_k, \overline{\mathbf{X}}_{k+1}^a, \overline{\mathbf{X}}_{k+1}^b\right]. \tag{5.16}$$

Project the matrices $\mathbf{K}$ and $\mathbf{M}$ onto the subspace E_{k+1}

$$\mathbf{K}_{k+1} = \overline{\mathbf{X}}_{k+1}^T \mathbf{K} \overline{\mathbf{X}}_{k+1}, \tag{5.17}$$

$$\mathbf{M}_{k+1} = \overline{\mathbf{X}}_{k+1}^T \mathbf{M} \overline{\mathbf{X}}_{k+1}. \tag{5.18}$$

Solve for the eigensystem of the projected matrices

$$\mathbf{K}_{k+1}\mathbf{Q}_{k+1} = \mathbf{M}_{k+1}\mathbf{Q}_{k+1}\boldsymbol{\Lambda}_{k+1}. \tag{5.19}$$

Calculate an improved approximation to the eigenvectors

$$\mathbf{X}_{k+1} = \overline{\mathbf{X}}_{k+1}\mathbf{Q}_{k+1}. \tag{5.20}$$

Let us recall that $q = p_k + 2r_k$, with p_k equal to the number of eigenvalues earlier converged, and that we have ordered all eigenvalues in (5.19) according to increasing magnitudes. Then we need to now measure which additional eigenvalue approximations $\lambda_i^{(k+1)}$ in $\boldsymbol{\Lambda}_{k+1}$ have also converged, that is, satisfy

$$\left[1 - \frac{\left(\lambda_i^{(k+1)}\right)^2}{\left(\mathbf{q}_i^{rr(k+1)}\right)^T \mathbf{q}_i^{rr(k+1)}}\right]^{\frac{1}{2}} \leq tolc; \quad i = p_k + 1, \ldots, p \tag{5.21}$$

where $\mathbf{q}_i^{rr(k+1)}$ is the $(i - p_k)th$ column vector in $\mathbf{Q}_{k+1}^{rr}$, which is a $2r_k \times 2r_k$ submatrix obtained by partitioning $\mathbf{Q}_{k+1}$ into

$$\mathbf{Q}_{k+1} = \begin{bmatrix} \mathbf{Q}_{k+1}^{cc} & \mathbf{Q}_{k+1}^{cr} \\ \mathbf{Q}_{k+1}^{rc} & \mathbf{Q}_{k+1}^{rr} \end{bmatrix} \tag{5.22}$$

Here the submatrices $\mathbf{Q}_{k+1}^{cc}$, $\mathbf{Q}_{k+1}^{cr}$ and $\mathbf{Q}_{k+1}^{rc}$ are of order $p_k \times p_k$, $p_k \times 2r_k$ and $2r_k \times p_k$, respectively.

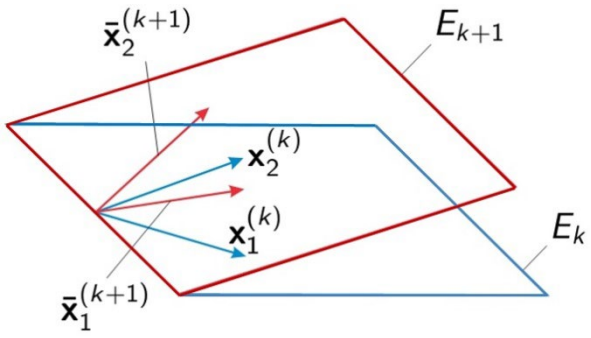
We should note that while we iterate with q iteration vectors, in the relation (5.21) we only test whether the smallest p eigenvalues have converged.

We then update the number of converged iteration vectors to p_{k+1} and continue the iteration if $p_{k+1} < p$, that is, increase k to $(k + 1)$ and go to Eq. (5.6).

After the number of required eigenvalues and vectors have converged, the Sturm sequence check is performed as in the original Bathe subspace iteration, see KJ Bathe [A].

Fig. 5.1 shows, for the simple case of two iteration vectors and $\mathbf{M} = \mathbf{I}$, geometrically the subspace E_{k+1} reached in the basic subspace iteration method and in the improved subspace iteration method. We can see that the subspace E_{k+1} reached in the improved scheme has turned significantly more from the subspace E_k than when using the basic scheme, for an analysis of this behavior, see the convergence analysis below.

(a)



(b)

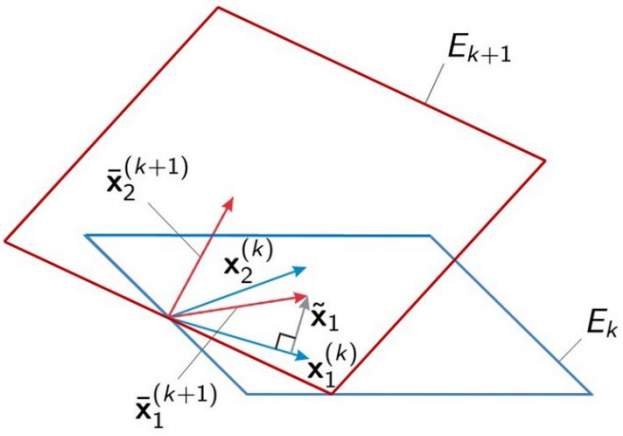


Fig. 5.1 Geometrical illustration of the subspaces E_k and E_{k+1} when $\mathbf{M} = \mathbf{I}$ and only two iteration vectors are considered: (a) in the basic method E_{k+1} is spanned by $\bar{\mathbf{x}}_1^{(k+1)}$ and $\bar{\mathbf{x}}_2^{(k+1)}$, (b) in the new method E_{k+1} is spanned by $\bar{\mathbf{x}}_1^{(k+1)}$ and $\bar{\mathbf{x}}_2^{(k+1)} = \mathbf{K}^{-1}\,\tilde{\mathbf{x}}_1$, where $\tilde{\mathbf{x}}_1$ is in the space spanned by $\bar{\mathbf{x}}_1^{(k+1)}$ and $\mathbf{x}_1^{(k)}$, hence $\bar{\mathbf{x}}_2^{(k+1)} = \mathbf{K}^{-1}(\bar{\mathbf{x}}_1^{(k+1)} + \gamma \mathbf{x}_1^{(k)})$ with γ determined by the Gram-Schmidt orthogonalization process, see Eq.(5.13); we notice that the $\mathbf{K}^{-1}$ $(\bar{\mathbf{x}}_1^{(k+1)})$ term is significant in driving the convergence because the subspace iteration scheme always establishes the best approximation to the eigenvalues and vectors, see KJ Bathe [A] and K.T. Kim and K.J. Bathe [C].

In practice, various details in the computations need attention, that is, some operations of Eqs. (5.6) to (5.20) can be directly parallelized, see KJ Bathe [G], some calculations can be reused, and some calculations can be performed more efficiently than formally given above, like the calculation of α_i, see K.T. Kim and K.J. Bathe [B] for some details on computational savings.

An analysis shows that the iteration effort in the first iterations using the accelerated subspace scheme, that is, when $p_k = 0$ or is close to zero, is only a small percentage larger than for the original subspace iteration. Of course, the actual increase in computational effort per iteration

depends on the specific eigenproblem considered, like the bandwidth of the matrices and the number of eigenpairs sought.

In the improved subspace iteration method, the most important reduction in solution time is achieved by using the forward turning vectors. In addition, the solution time is reduced due to r_k decreasing as *(k)* is increasing. Namely, the number of vectors used in the inverse iterations (in Eqs. (5.7 and (5.15)) is decreasing as p_k is increasing. However, this part of saving in solution time may not be important when only a relatively few eigenpairs are solved for, as it was the aim in the early use of the Bathe subspace iteration method in earthquake engineering, see K.J. Bathe [A, F, J].

5.2.2 Convergence of the solution scheme

As in the basic subspace iteration, we order the eigenvalues in increasing magnitude. Then, provided that the starting iteration vectors in $\mathbf{X}_1$ are not **M**-orthogonal to any of the eigenvectors sought we have, for $i = 1, \ldots, p$,

$$\lambda_i^{(k+1)} \to \lambda_i \quad and \quad \mathbf{x}_i^{(k+1)} \to \boldsymbol{\varphi}_i \quad as \quad k \to \infty \tag{5.23}$$

where $\mathbf{x}_i^{(k+1)}$ is the ith column vector in $\mathbf{X}_{k+1}$.

It is now of interest to identify the rate of convergence and compare this rate with the rate when using the basic subspace iteration scheme.

For the analysis we use the approach of convergence analysis for the original Bathe subspace iteration method, see K.J. Bathe [K], and focus in particular on the effect of the turning vectors.

First, we change the iterations to the basis of eigenvectors, but of course only conceptually,

$$\mathbf{X}_k = \mathbf{\Phi}\mathbf{Z}_k \tag{5.24}$$

where $\mathbf{\Phi} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_n]$ and since $\mathbf{X}_k^T\mathbf{M}\mathbf{X}_k = \mathbf{I}$, we have that $\mathbf{Z}_k^T\mathbf{Z}_k = \mathbf{I}$. Using this change of

basis in Eq. (5.7) leads to

$$\mathbf{\Lambda}\overline{\mathbf{Z}}_{k+1}^a = \mathbf{Z}_k^a \tag{5.25}$$

where $\mathbf{Z}_k^a$ is of order $n \times r_k$ and contains the first r_k vectors of $\mathbf{Z}_k$. Assuming that $q = 2p$, $p_k = 0$, hence $r_k = p$, and that all vectors in $\overline{\mathbf{Z}}_{k+1}^a$are employed for use of corresponding forward turning vectors, the subspace E_{k+1} is spanned by $\left\{\overline{\mathbf{Z}}_{k+1}^a, \bar{\bar{\mathbf{Z}}}_{k+1}^a\right\}$ with $\bar{\bar{\mathbf{Z}}}_{k+1}^a = \mathbf{\Lambda}^{-1}\overline{\mathbf{Z}}_{k+1}^a$ because the subspace iteration always establishes the best approximation to the eigenvalues and vectors.

This analysis shows the effectiveness of using the forward turning vectors in the subspace iterations: in essence, one iteration including the turning of vectors has the effect of two iterations in the basic scheme, see also Fig. 5.1. Consequently, and based on further analysis, we have the general result that the iterations converge to the eigenvectors at the rate, see K.T. Kim and K.J. Bathe [B]

.

$$\frac{\left\|\mathbf{z}_i^{(k+1)} - \hat{\mathbf{e}}_i\right\|_2}{\left\|\mathbf{z}_i^{(k)} - \hat{\mathbf{e}}_i\right\|_2} \leq \left(\frac{\lambda_i}{\lambda_{q+1}}\right)^2 + o(\varepsilon_i^{(k)}); i = 1, \ldots, p. \tag{5.26}$$

where $\hat{\mathbf{e}}_i$ is the *i'* th eigenvector sought and $\mathbf{z}_i^{(k)}$ is the approximation to that vector after *k-1* iterations.

Here we assume, as mentioned already, that we are using $q = 2p$ and $p_k = 0$, that the iteration vectors are not deficient in the eigenvectors sought, and that in each iteration all p vectors are turning vectors. We further assume that the iteration vectors are ordered corresponding to the magnitudes of the approximate eigenvalues, from the smallest to the largest values.

With this notation, the relation (5.26) shows that the iteration vectors $\mathbf{x}_i^{(k)}$, $i = 1, \ldots, p$, converge with the rates $\left(\frac{\lambda_i}{\lambda_{q+1}}\right)^2$. The rates of convergence of the corresponding eigenvalues are $\left(\frac{\lambda_i}{\lambda_{q+1}}\right)^4$, $i = 1, \ldots, p$, because these are calculated from the Rayleigh quotient.

These convergence rates may be compared with the rates of convergence of the basic subspace iteration, in which the iteration vectors $\mathbf{x}_i^{(k)}$, $i = 1, \ldots, p$ converge with the rate $\frac{\lambda_i}{\lambda_{q+1}}$ and the corresponding eigenvalues with the rate $\left(\frac{\lambda_i}{\lambda_{q+1}}\right)^2$, $i = 1, \ldots, p$. Hence the improved method converges in considerably less iterations.

For illustration, consider the convergence of the new scheme in a simple example, see K.T. Kim and K.J. Bathe [B], in which we use the diagonal matrix $\mathbf{K} = diag(1,2,\ldots,12)$ and $\mathbf{M} = \mathbf{I}$. We want to solve for the smallest three eigenvalues, hence $p = 3$. We use $q = 2p$, $p_k = 0$ and $\bar{t}_k = p$, and we compare the results with those from the basic subspace iteration method also using $q = 2p$. In both solutions, six random vectors are used as the starting vectors (since using the usual algorithm for establishing the starting vectors would in this simple case result in immediate convergence of the subspace iteration method).

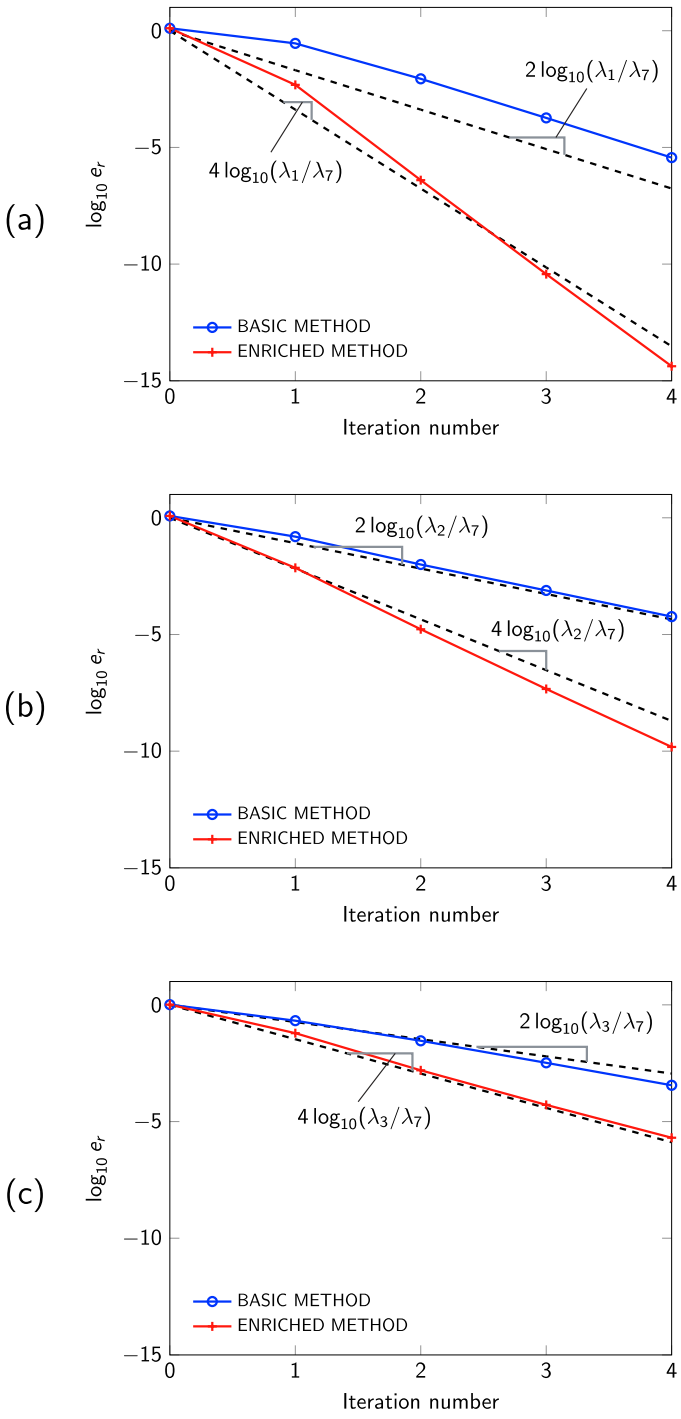


Fig. 5.2 Relative errors measured in the subspace iteration solution of the simple 12 degrees of freedom example problem, with (a) to (c) for the first, second and third eigenvalues, K.T.Kim and K.J. Bathe [C].

Figure 5.2 shows the relative errors of the calculated eigenvalue approximations, denoted as $e_r = \frac{(\lambda_i^{(k)} - \lambda_i)}{\lambda_i}$ when using the basic subspace iteration method and the subspace iteration method with the acceleration. The computed rates in the iterations follow quite closely the theoretical rates, given by the dashed lines. These solutions show the effectiveness of the improved scheme.

5.2.3 Illustrative solutions

The objective in this section is to briefly demonstrate the performance of the subspace iteration method through some solutions. In all cases, we use q iteration vectors, with $q = 2p$, and calculate all required p eigenvalues and vectors with $tolc = 10^{-6}$ for convergence.
Additional problem solutions are given in K.T. Kim and K.J. Bathe [B] and S. Lee and K.J. Bathe [C] and in results obtained using the ADINA program, see Bentley Systems, Inc. [A].

A program can be downloaded from the MIT web page of K.J. Bathe. However, a number of the subroutines in the implementation of the enriched scheme are those also used in the original subspace iteration method and are given in K.J. Bathe [A].

Analysis of a ring

We consider the solution of a ring structure, shown in Fig. 5.3 for the lowest frequencies of the structure.

The structure is modeled using meshes of equal 8-node brick elements. The finest mesh used consists of 22 x 22 x 380 elements through the thickness, width and over the circumference, respectively. The consistent mass matrix is used in each case. The number of degrees of freedom n and the mean half bandwidth m of the **K** matrix after factorization are increasing in the three case considered given in Fig. 5.4. The solution was obtained using a laptop with a single core Intel 2.40 GHz.

The figure gives the solution times of the accelerated subspace iteration method and the basic scheme. We note that the improved subspace iteration uses about 1/3rd the computational time and that the time increases linearly in proportion with the number of eigenpairs sought. This linear increase is valuable and is due to the acceleration procedure used but whether it is reached depends on the problem solved. Indeed, also a better than a linear proportional increase may be obtained, see for example, the solution of the exhaust manifold given next.

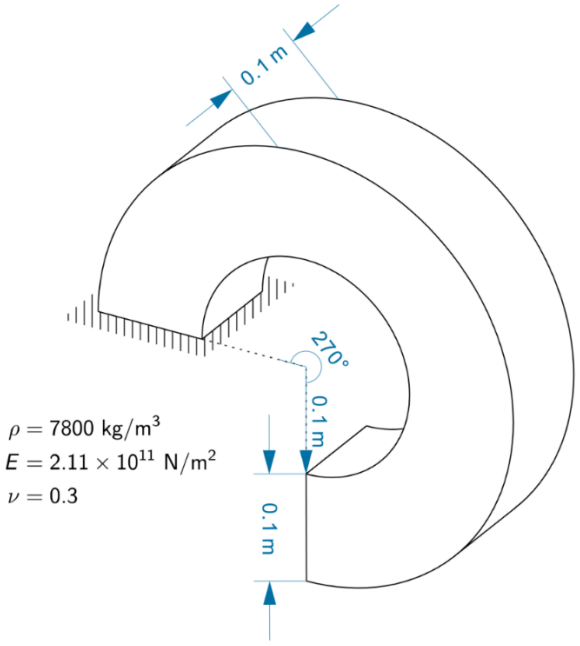
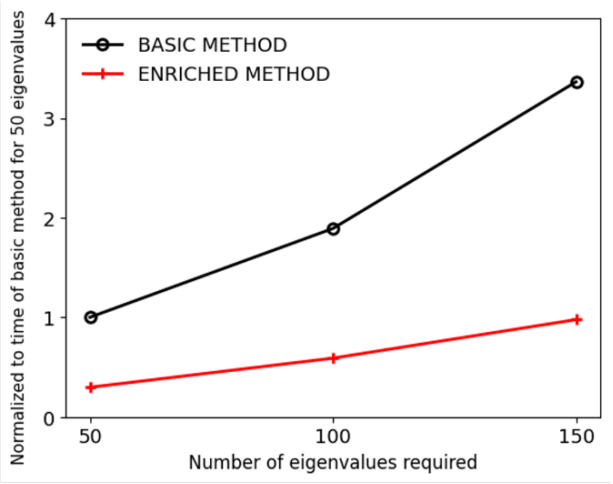
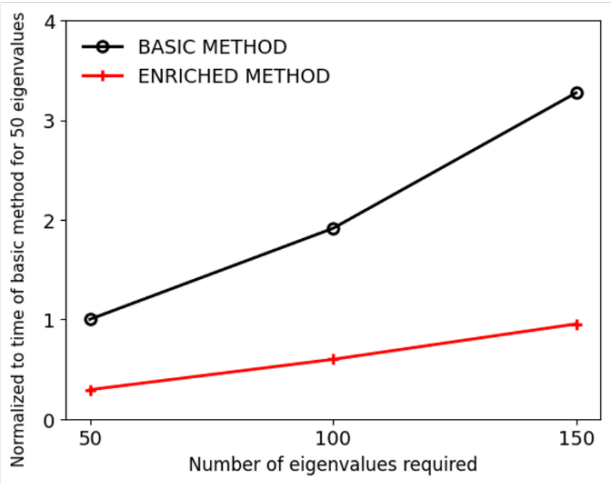


Fig. 5.3 The ring considered for the frequency solutions, K.T. Kim and K.J. Bathe [C].

(a)



(b)



(c)

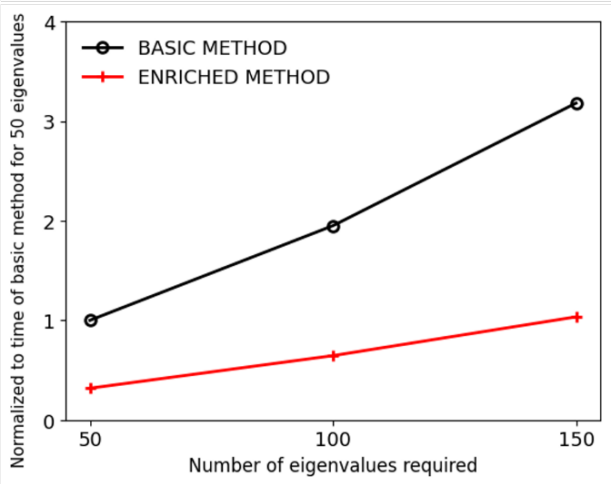


Fig. 5.4 Solution times solving the ring problem of Fig. 5.3, all times are normalized to the time used for the problem for p =50; (a) n = 216,000, m = 675; (b) n = 389,880, m = 1,083; (c) n = 603,060, m = 1,587.

Analysis of an exhaust manifold

This larger problem was run using ADINA in ADINA R & D, Inc. prior to the purchase of ADINA by Bentley Systems, Inc. [A]. The geometry of the problem is shown in Fig.5.5 and the number of degrees of freedom are 2,220,273. The model includes contact conditions, and was run on a Dell 2-processor computer, using Linux with 16 cores, in SMP parallel processing. We include this problem to merely show that practical engineering problems are solved very efficiently.

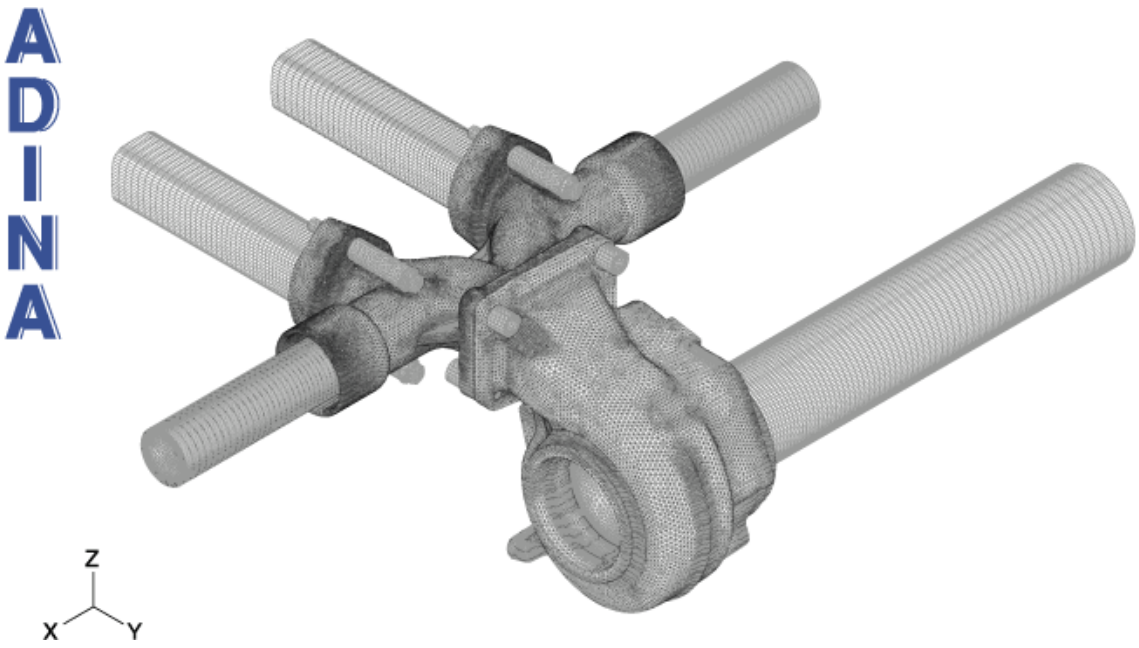


Fig. 5.5 The model of the manifold exhaust solved for its lowest (nonzero) frequencies

Table 5.1 gives the solution times for the calculations of 50 to 300 eigenpairs using the accelerated subspace iteration method. We notice the effectiveness of the solution procedure and that the solution times increase approximately linearly with the number of eigenvalues and vectors calculated.

Table 5.1 Solution of exhaust manifold, number of equations = 2, 220,273

Number of frequencies and mode shapes	Bathe method Solution times in minutes
50	4
100	7
200	11
300	23

Note that in this solution a better than a linear increase in solution time used has been reached. Using the 4 minutes expended for 50 eigenpairs as a basis, we would have in a linear proportional increase 8, 16, and 24 minutes for the 100, 200, and 300 eigenpairs, but instead have 4, 7, 11, and 23 minutes.

5.3 Exercises

We suggest these Exercises to obtain further insight into the solution scheme discussed in this chapter. To complete an Exercise, the material given in KJ Bathe [A] may need to be known.

5.3.1

Consider the generalized eigenvalue problem. Prove that for an eigenvalue of multiplicity *m,* there are infinitely many eigenvectors but only *m* linearly independent such vectors.

5.3.2

Show that the procedure in Eq. (5.8) is the standard Gram-Schmidt process, now applied to achieve orthogonality to many vectors.

5.3.3

Prove that the following **M**-orthonormality holds

$$\left[\boldsymbol{\Phi}_k, \mathbf{X}_k^a, \widetilde{\mathbf{X}}_k^b\right]^T \mathbf{M}\left[\boldsymbol{\Phi}_k, \mathbf{X}_k^a, \widetilde{\mathbf{X}}_k^b\right] = \mathbf{I}$$

where the matrices in this relationship are defined in Section 5.2.1.

5.3.4

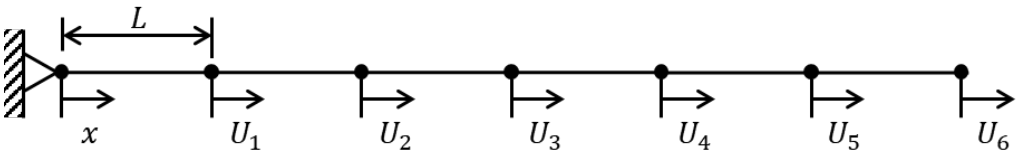
Prove that in the subspace iteration scheme, the eigenvalue approximations are always obtained from the eigenvector approximations using the Rayleigh quotient.

5.3.5

Show that if in the subspace iteration with q iteration vectors the rate of convergence to the ith eigenvector is $\left(\frac{\lambda_i}{\lambda_{q+1}}\right)^2$, then the rate of convergence to the corresponding eigenvalue is $\left(\frac{\lambda_i}{\lambda_{q+1}}\right)^4$.

5.3.6

Consider the six degrees of freedom one-dimensional bar shown below. The bar can move only into the x-direction and is modelled using six equal-length 2-node traditional truss elements. Establish the stiffness matrix **K** and consistent mass matrix **M.** Then show how to use the above discussed subspace iteration scheme when q = 4. In particular, show how to use Eqs. (5.13) and (5.14) in the procedure.



Young's modulus E
Mass density ρ
Cross-sectional area A

6. Machine Learning to Enrich Finite Element Procedures

The presentations in the above chapters and in K.J. Bathe [A] focus on classical finite element procedures and some latest developments. Many of these procedures are already quite mature and are already in wide use in research and industry. Because of their maturity and effectiveness, new more valuable procedures to replace those now in use, although still needed, might be difficult to reach.

However, a new field has emerged as an avenue to significantly enrich finite element procedures – namely the field of Machine Learning (ML). The techniques of ML can be developed to significantly propel the use of the current finite element procedures to new avenues of analyses, in that new models and more complex and effective simulations in engineering and the sciences can be pursued.

Machine Leaning is a field of Artificial Intelligence (AI), which is receiving major attention in many areas and is already used abundantly. In recent years, research in AI has accelerated and lately major advances are predicted for numerous applications.

The word “Artificial Intelligence” was coined by John McCarthy a professor at Dartmouth College who, with colleagues in the field, organized the Dartmouth 1956 workshop on AI, see J. Moor [A]. This workshop is frequently regarded as a seminal event for Artificial Intelligence as a field. Prior to that workshop, the idea that computers may simulate intelligence was already investigated, notably by Alan Turing with the “Turing test” a test to determine if a machine can exhibit intelligent behavior, see A.M. Turing [A]. and S.B. Cooper and J. Van Leeuwen [A].

With the field of AI now huge, we focus below only on a subfield of AI, namely ML techniques. We discuss some basic ML procedures and developments in their use in conjunction with finite element procedures, including the use with classical finite element methods, constitutive modeling, structural health monitoring, and the in the use of digital twins. However, the presentation is by no means exhaustive, in that by far not all of the many developments are mentioned that have been pursued and are already used.

The field of ML has progressed quite fast during the recent years, and new developments are continuously appearing. Hence it is too early to give a detailed exposition of *advanced* ML procedures that are seen as widely accepted to be most valuable, and that will remain most valuable for a long time. Such assessment is best given in some years when the advanced schemes have somewhat matured. This situation is different from the current state of finite element methods. However, we can still focus on those ML procedures that are *fundamental* and for that reason will likely be used for a long time.

Namely, the *basic* approach in ML will hardly change and it is of much value to understand this approach – if also only to spur further research in the field. In this vein, the purpose of including this chapter is to focus on the basic approach, give a “simple” introduction to ML procedures (with some examples) and on illustrating the

possible significant impact that the use of ML may have in the applications of finite element procedures. Much more information on the methods mentioned here and other ML schemes are given in the references, and the developments referred to therein. In addition, it is important to always keep in mind that many more valuable advances will probably be forthcoming in the years to come.

6.1 Some basic machine learning procedures

In the use of ML methods, invariably much data, and frequently a voluminous amount of data, needs to be processed. Indeed, ML procedures are most useful when a massive amount of data needs to be analyzed to extract valuable information for use of prediction. In finite element analysis, we are primarily interested in predicting the behavior of a structure or system when subjected to certain loads, and these loads may be applied during the construction, during the structure's life time, and on its dismantling. Hence it is the aspect of an effective prediction of structural behavior that is frequently of interest when combining finite element procedures with ML techniques.

In ML techniques, some basic procedures are employed in various forms. Some of these basic techniques have been employed already in some other form for a long time, while others have been developed only recently. Figure 6.1 lists some basic ML techniques.

The figure also refers to some Neural Networks (NN) that have been specifically developed for use in ML, some only during the recent years, see T. Rabczuk and K.J. Bathe [A].

The premise of using ML with finite element procedures is that the effective use of ML methods can enhance the analysis methods, and we briefly discuss some basic schemes and NN in the following sections. There are of course, many additional basic procedures used in NN, like optimization algorithms but these are quite standard and are described in many books, see for example J. Nocedal and S. J. Wright [A].

We should also note that there are many NN used in areas other than computational mechanics, hence not mentioned here, as for example described in F.J. Montana, E. Cueto and K.J. Bathe [A]. This reference also lists various software offerings that can be used to build ML models (e.g. TensorFlow, OpenNN) and languages for ML projects (e.g. Python). Such widely available ML tools use various standard computational procedures (like

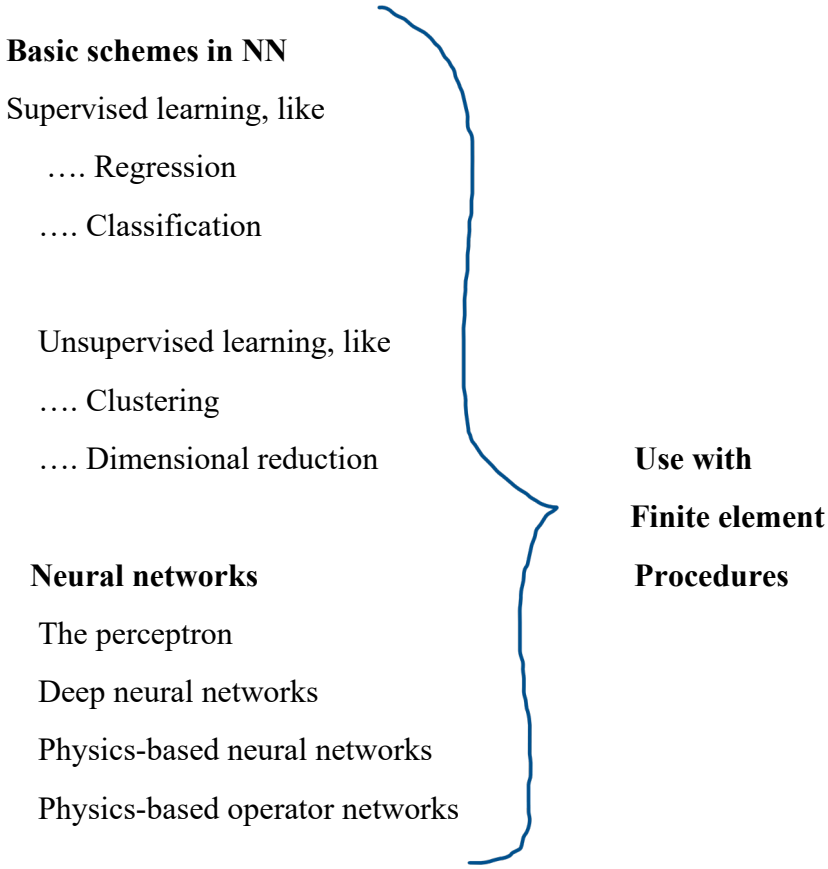


Figure 6.1 Some basic schemes and some NN in ML possibly enhancing Finite Element Procedures

curve fitting) valuable in establishing solutions based on ML. Indeed, the widely available software offerings to use as tools to build specific ML procedures has much contributed to the exponential growth in the use of ML methods (like the early widely available finite element programs NASTRAN, SAP IV and NONSAP contributed to the rapid growth of use of finite element methods, see Chapter 1).

6.1.1 Supervised and unsupervised learning

The aim in using ML is that once a NN has been trained with data, the network can be used to predict data in general (of course within the space of the training but without further training). This prediction can be of much value in finite element analysis.

Supervised learning

The NN consists in general of nodes and layers that mimic the thought and decision processes of the human brain, see for some details Section 6.2. Conceptually, in supervised learning the process is to train the NN with known data (also called labeled data), obtained through physical observations, physical experiments, or numerical experiments using, for example, a finite element program. The training involves the use of known input data and the corresponding known output data. For example, for given strain data of a solved problem (hence the stress data is known) with a certain geometry and loading, the NN shall obtain the known stress data. This requirement means the NN must be trained for the given strain data to iterate towards and achieve the corresponding stress data (to an acceptable error in predicted stress). Once this training has been achieved in a general setting, the NN can be used to *predict* stresses for any given strain data, within the space of training, not knowing a priori the stress solution.

However, in some cases it is difficult or almost impossible to establish the full training data referred to above, specifically, the correct output data to the given input data is unknown. Only output data that is reasonably close to the correct data is known. In this case a transfer learning step can be included in which some known data from earlier solved cases is employed and then the usual ML scheme is used. In particular the initial weights and biases may be chosen from known data, from a single data set or an ensemble of data sets. Various procedures are employed in the transfer learning step, see for example K.R. Weiss and T. M. Khoshgoftaar [A]. Transfer learning can also be used to reduce the computational effort in using NN, see e.g. S. Goswami, C. Anitescu, S. Chakraborty and T. Rabczuk [A] and H. Guo, X. Zhuang, P. Chen, N. Alajlan and T. Rabczuk [A], and J.N. Heidenreich, C. Bonatti and D. Mohr [A].

Two basic procedures widely used in NN are *regression* and *classification,* and these procedures are trained and used in supervised learning. As already implied above, the common feature in the procedures is that ordered known data are used and *predictions* are made. More specifically, in regression, functions are established using the known data for *predictions* of unknown data, and in classification, decision processes are established which are used for *predicting* to which class a given item belongs.

Unsupervised and other learning procedures

In the case of unsupervised learning, the input data is not ordered according to features, but is simply input to the NN as “raw data”. Unsupervised learning procedures are typically used to cluster data (like for structures, to cluster these according to a concrete or a steel construction, the number and size of windows, damage undergone, and so on) or to seek a reduction of the dimensionality of the data.

Hence, *clustering* and *dimensional reduction* are two procedures trained and used in unsupervised learning. The common feature in these procedures is that unordered data is used and important information is extracted. In clustering, groups or clusters in the given data based on similarities are established. In dimensional reduction, features, or combinations of them, in the available data are ordered according to greater and less importance. The automatic clustering and dimensional reduction of data in a NN (that is, the automatic ordering) can be very useful for subsequent use of the previously unordered data.

In addition, there are semi-supervised, and reinforced or active learning driven procedures. In semi-supervised learning, the approach is based on a combination of supervised and unsupervised learning. In active learning, the approach is to have the NN automatically change the decision processes depending on the result achieved in the NN. If the error, measured in some norm, is large, the NN changes the decision process with the aim to reduce the error. In all these approaches, the solution errors shall be minimized which means that optimization algorithms need to be used. Hence, such algorithms are abundantly employed in NN computations, see Section 6.2.

In the following two sections we discuss as examples one application in supervised learning, that is, regression, and one in unsupervised learning, that is, dimensional reduction.

6.1.2 Regression

The basic task of regression algorithms is to work on data to find similarities and trends, with the objective to be able to predict. Hence regression algorithms are used to predict unknown numerical values based on known input data. The algorithms are trained in supervised learning and then applied usefully in engineering and the

sciences, for example in predicting product performances and biological events. A regression helps to uncover relationships between variables and make informed predictions for future data points.

There are many regression algorithms, including Linear Regression, Polynomial Regression, Support Vector Regression, Decision Tree Regression, see F.J. Montana, E. Cueto, and K.J. Bathe, and we shall focus here only on the main basic concepts.

An important tool is a linear regression scheme, see T. Rabczuk and K.J. Bathe [A].

Consider that for given data points x_i as inputs (like age of a building) we have the data y_i (cost of repairs per year) as outputs. These data points are plotted as y_i versus x_i, $i = 1,...N$, and then for predictions of the probable cost of repairs of buildings of that kind, a regression expression may be used.

In the simple linear regression algorithm, the objective is to find a straight line to approximate the relationship for the *N* data points.

Let us consider the more general case of $\tilde{y} = \boldsymbol{w}^T\boldsymbol{\varphi}(x) + b$, where the value of $\tilde{y}$ shall approximate the value of y for any value of x. The vector of weights $\mathbf{w}$ lists constants to be determined for the basis functions in $\boldsymbol{\varphi}(x)$, *x* is the feature and *b* is an offset.

A common procedure to establish the unknown weights and the offset is to minimize a *cost function*, given by the mean square error between the known data points and the approximation

$$C = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i)^2 \qquad (6.1a)$$

or substituting for $\tilde{y}_i$ we obtain

$$C = \frac{1}{N}\sum_{i=1}^{N}([\mathbf{w}^T\boldsymbol{\varphi}(x_i) + b] - y_i)^2 \qquad (6.1b)$$

This expression is minimized with respect to the weights listed in $\mathbf{w}$ and the offset b. In linear regression, the predictive function is established using only one basis function corresponding to a straight line, hence $\mathbf{w}^T\boldsymbol{\varphi}(x) = m\,x$ with m being the weight. Using this assumption and the input data at x_i (that is y_i), C is given by

$$C = \frac{1}{N}\sum_{i=1}^{N}([m\ \mathrm{x}_i + b] - y_i)^2 \qquad (6.2)$$

Taking the partial derivatives of C with respect to m and b , to look for the minimum error, and inserting the data points (x_i, y_i), we can solve for m and b to obtain the predictive function, a straight line $\tilde{y} = m\,x + b$.

Of course, in practice, $\boldsymbol{\varphi}(x)$ may contain more basis functions, for example, we may use $x, x^2, x^3, ...$

to have

$$\mathbf{w}^T\boldsymbol{\varphi}(x) = w_1x + w_2x^2 + w_3x^3... \qquad (6.3)$$

or use also other than polynomial terms. The process of calculating the weights w_i is as given above using the minimization of the error between the approximation and the given data.

To evaluate the accuracy of the approximation, we can calculate

$$SS = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}(x_i) - y_i)^2 \qquad (6.4)$$

and normalize *SS* to a reasonable value, like the square of the mean value of the assumed function over the solution domain.

This procedure can be generalized to other schemes of regression, like ridge regression, support vector regression and Gaussian process regression, see for example P. Wei and M. Beer [A]. In each case available data is employed and functional expressions are evaluated to predict unavailable data, like additional data points not used (for example, because not known yet) in the establishment of the predictive function.

Also, this approach can mathematically be widened to have a probabilistic regression, in which case mean estimates and uncertainty estimates are evaluated, like in Gaussian Process regression. Then we do not only have a prediction of data but also obtain a measure of confidence in the predicted data. For a general overview of ML approaches for uncertainty quantifications, see Y. Shi, P. Wei, K. Feng , D.-C. Feng and M. Beer [A].

6.1.3 Dimensional reduction

In some analyses, we may have many features in the data, with each feature representing in essence a "dimension", like for a building of steel or concrete construction, the features of number of windows and type of windows, number of stories, number of shear walls and elevator shafts, the features of different electrical connections, the features of the level and flow of the ground water, type of foundation, and so on. If our objective is to establish the vulnerability in an earthquake loading, certain features are much less important than others when characterizing the structure. In this case the NN shall identify the important features and in this way reduce the dimensionality of the data.

As mentioned earlier, dimensional reduction is an application of unsupervised learning.

The process of dimensional reduction can be likened to the process of solving for the eigenvalues and eigenvectors (frequencies and mode shapes) of a finite element model of a structure in that these eigenpairs represent the mass and stiffness characteristics of the model. If the structure and hence the finite element model are subjected to certain dynamic loadings, only a small subset of all eigenvalues and vectors may need to be used for the dynamic response analysis, thus a dimensional reduction is employed very efficiently for the response solution. For example, in earthquake loadings frequently only the lowest frequencies and associated mode shapes of a structure need to be solved for and used in a mode superposition solution, because only these lowest frequencies and associated mode shapes are excited by the loading, see K.J. Bathe [A].

Hence, in frequency and mode superposition analyses, we, in essence, reduce the dimensionality from the total number of degrees of freedom of the model, which could be from one thousand to over a million, to the number of frequencies and corresponding mode shapes used, typically from ten to a few hundred. The mode shapes in an analysis are the base vectors of the new basis reached in the reduction of the dimensionality from the total number of degrees of freedom to the number of modes.

This calculation of eigenvectors can be regarded in general as a principal component analysis (PCA). Given data with certain features, and possibly many of them, we want to find the principal features and their “directions”. For example, in earthquake loading, to establish the vulnerability of a steel building with the features mentioned above, we need to identify the major contributors to render the building vulnerable and this requires establishing a matrix and finding the principal components (features and values of maximum significance or relevance). Using the principal component analysis, this objective can be achieved by establishing the covariance matrix **S** with entries

$$S_{jk} = \frac{1}{n}\sum_{i=1}^{n}(x_{j(i)} - \overline{x}_j)(x_{k(i)} - \overline{x}_k) \qquad (6.5)$$

where *i* corresponds to the sample considered, with *n* equal to the total number of samples, and *j, k* are the features. The overbar denotes the mean value.

For the above example of a steel building, *j* and *k* might be the number of stories, number of small windows, number of large windows, number of elevator shafts, and number of shear walls. In this case, only 5 features are considered.

The eigenvalues and vectors of **S** correspond to the principal components of the combinations of features and "geometrically" their directions, which indicate the coupling of effects. Assume that *j* and $k = 1, \ldots m$, with *m* equal to the number of features, then **S** is of size $m \times m$ and the largest eigenvalue and corresponding vector indicate how the *m* features are coupled to each other in the most pronounced manner.

These thoughts regarding the PCA are closely related to frequency solutions widely used in dynamic analyses. We also refer to the method of proper orthogonal decompositions (POD) in which a response solution may be expanded in modes that are superimposed (like in mode superposition solutions). We use a POD basis in Section 6.3.2.

The principal component analysis and other methods to reach a reduction of dimensionality are, for example, discussed in T. Rabczuk and K.J. Bathe [A].

6.2 Neural networks (NN)

Having briefly discussed some basic procedures used in ML, and specifically in Neural Networks we now aim to discuss to some NN that have been published, are already employed, or show significant promise. However, there is still much research needed to further develop NN and their use with finite

element methods. The aim in the presentation given below is to present the basic process of NN solutions and then to briefly "showcase" their use in Section 6.3 in some areas. For an introduction to NN as a category of Computational Intelligence (a part of Artificial Intelligence) see also AP Engelbrecht [A] and for recent advances see T. Rabczuk and K.J. Bathe [A] and the further references given below.

In essence, our aim is to have a NN mimic the human thinking process, that is, with a given input data produce a desired output. Using finite element procedures with that output can generalize and greatly widen the space of applications of finite element methods.

6.2.1 The basic process of a NN

A neural network consists, in essence, of many artificial neurons arranged to work together in layers.

For the purpose of a simple explanation, let us consider a single artificial neuron: here some data is passed to the neuron, say x, the neuron processes the data and produces an output, say y. The processing may involve a number of simple steps, mostly like the input x is multiplied by a weight w, to obtain $z = wx$, and then passed through a filter described by a function $f(z)$. Using the sigmoidal function, we have with y = $f(z)$

$$y = \frac{1}{1+e^{-\alpha z}} \tag{6.6}$$

where α is a constant that may be adjusted. However, many different functions can be used, like the tanh (z) and the swish (z) functions, see for example A.P. Engelbrecht [A]. These are called *activation functions* because they cause or activate a particular action. Here, and below, we consider only one parameter, that is α, but of course functions with more parameters can in principle be used.

Naturally, in this simple process of a single neuron only very limited actions can be performed, but when many such neurons work together in a network, with each neuron possibly having different weights and filters, many operations can be performed simultaneously and immense computing possibilities can arise. We may recall that in the human brain, billions of neurons work together to gather information, store the information and process it with each applying simple operations towards a particular result.

Instead of using a single artificial neuron, let us consider the case of two artificial neurons, each obtaining input to process, with the objective to obtain a useful output, see Fig. 6.2. The input consists of the bias and the two data x_1 and x_2. The bias and data are passed after multiplication by the weights $w_1,...,w_6$ to the two artificial neurons, which are called *nodes*. Thus, a weighted input is used for both nodes which then apply, respectively, the activation functions $f_1(z_1), f_2(z_2)$ to obtain the output.

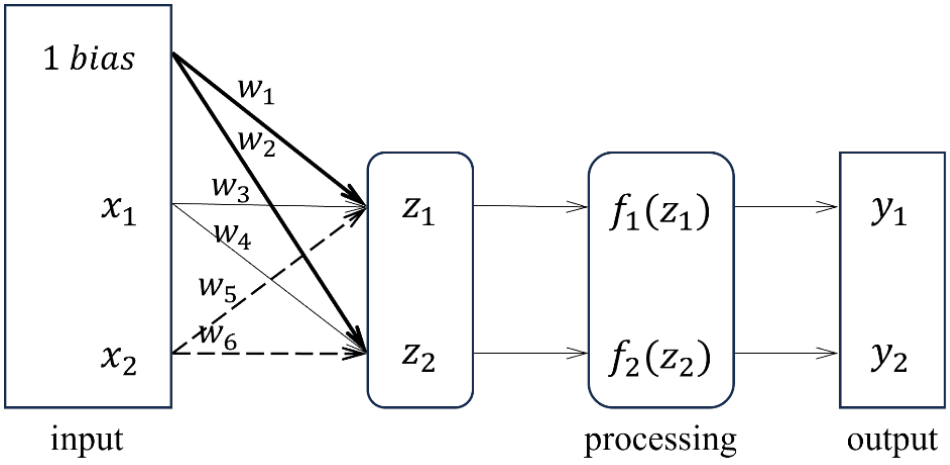


Fig. 6.2 A simple NN with an input layer of x_1, x_2 and the bias b =1.0, with two nodes in the processing layer

As shown schematically in Fig. 6.2, the NN consists of the *input layer, the processing layer* for processing, and the *output layer* (to give the output). Here we assume that the adjustable variables are the 6 weights (including the bias) and 2 constants in the two activation functions of the NN. Therefore, a total of 8 parameters can be adjusted and in fact be optimized to obtain a desired output. Considering a supervised learning process, the optimization is performed by working with a loss function to establish the error between the calculated output y_1, y_2 and the known "reference" values y_1^r, y_2^r. The superscript "*r*" denotes "reference" which in practice means observed values in physical measurements or in a numerical simulation, like a finite element analysis. We discuss the use of a loss function briefly in the next section.

The action of this single and simple processing layer in Fig 6.2 can be mathematically written in the form

$$\mathbf{y} = \mathbf{f}(\mathbf{z}) \tag{6.7}$$

where $\mathbf{z}$ lists z_1 and z_2 and is defined as

$$\mathbf{z} \coloneqq \mathbf{W}\mathbf{x} \tag{6.8}$$

In Eq. (6.8), the matrix $\mathbf{W}$ contains the weights and the vector $\mathbf{x}$ contains the bias as its first entry, hence $x_0 = b$, and the second and third entries are x_1 and x_2. Hence the matrix $\mathbf{W}$ is of order 2×3.

Equations (6.7) and (6.8) can be generalized in that the input may consist of *n* values and there may be *m* nodes in the NN processing unit to produce *m* output values. Then using again the bias in the vector $\mathbf{x}$, that is, $x_0 = b$, the matrix $\mathbf{W}$ is now of order $m \times (n+1)$ and these weights can now be optimized in addition to the *m* parameters in the activation functions (as mentioned above, using one parameter per node, but more parameters could be used in each node).

The process described above relates to the early work of WS McCulloch and W. Pitts [A], F. Rosenblatt [A], and B. Widrow and M.E. Hoff [A]. However, the NN in Fig. 6.2, even with more nodes in the processing layer, can hardly be used to solve complex engineering problems.

During recent decades major advances have been reached by generalizing the NN to multiple processing layers. These layers can contain a variable number of nodes, different weights from layer to layer, different powerful activation functions and learning processes to optimize the weights and parameters. The complete network is referred to as a *deep* neural network and can be used for the solution of complex problems.

6.2.2 Deep neural networks

The deep NN generalize the simple NN which uses a single processing layer, like shown in Fig. 6.2. The generalization to reach a deep NN is based on using more processing layers than the single one presented and discussed in

the previous section. The word "deep" refers to the fact that more than a single processing layer is used; all processing layers are also referred to as *hidden layers* because they are not seen and "work in the background". There may be many processing layers, like hundreds or more.

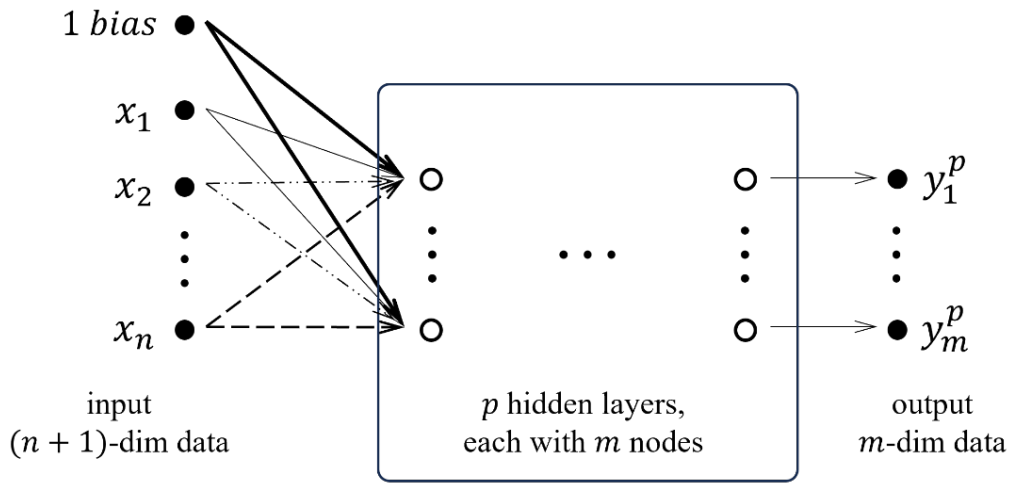


Fig. 6.3 A deep NN with p hidden layers, each layer has m nodes, but different numbers of nodes in the layers may also be used

Figure 6.3 shows schematically a deep NN. The NN is still quite simple because it assumes that each processing layer consists of m nodes, that the required output also consists of m items, and there are no final weights applied from the last processing unit to the output.

The mathematical statements in Eqs. (6.7) and (6.8) are directly applicable but now used sequentially, for each of the processing layers to obtain from the input $\mathbf{x}$, which includes the bias and the associated weight, the output $\mathbf{y}^p$. Referring to Fig. 6. 3, the input $\mathbf{x}$ including the bias is a vector of length n+1 and with the p processing layers in each layer containing m nodes, we can write

$$
\begin{aligned}
\mathbf{y}^1 &= \mathbf{f}(\mathbf{z}^1); \quad \mathbf{z}^1 := \mathbf{W}^1\mathbf{x} \\
\mathbf{y}^2 &= \mathbf{f}(\mathbf{z}^2); \quad \mathbf{z}^2 := \mathbf{W}^2\mathbf{y}^1 \\
\mathbf{y}^3 &= \mathbf{f}(\mathbf{z}^3); \quad \mathbf{z}^3 := \mathbf{W}^3\mathbf{y}^2 \\
&\cdots \\
\boldsymbol{y}^p &= \boldsymbol{f}(\boldsymbol{z}^p); \quad \mathbf{z}^p := \mathbf{W}^p\mathbf{y}^{(p-1)}
\end{aligned}
\tag{6.9}
$$

where we assumed that no further biases are used, hence the vectors $\mathbf{y}^i$ in these equations are all of dimension m, and we use the same activation function $\boldsymbol{f}$ in each layer, but with a different value of the constant. Then the whole process contains now $[m \times (n+1) + m \times m \times (p\text{-}1)]$ independent weights and an additional $m \times p$ parameter values in the activations (assuming only one parameter per activation function in the layer). Hence if in a simple example, n = 10, $m = 6$, $p = 4$, we have 198 variables to optimize. In practice, much more input data, many more nodes and different layers and activation functions may be used. Hence we clearly need an effective process to achieve the optimization, that is, to predict the desired reference values, as we briefly discuss below.

We note that this process of optimization assumes supervised learning since the input data is used to predict the output data which is assumed to be known. Hence, here we want to train the NN to predict the desired output (see Section 6.1.1). Once the NN has been trained, the network can be used for other input data of the same kind and predict the correct (prior unknown) output data.

The key to the optimization process is to use a loss function (the “loss” function measures how far off the model's predictions are from the actual target values). Typically, the root-mean-square error

between the predicted values in $\mathbf{y}^p$ and the "correct" (that is, target) values in $\mathbf{y}^r$ is used

$$RMSE = \sqrt{\frac{1}{m}\sum_{i-1}^{m}(y_i^p - y_i^r)^2} \qquad (6.10)$$

where the components of the vectors $\mathbf{y}^p$ and $\mathbf{y}^r$ are the values y_i^p and y_i^r, respectively. If there are more results on errors than just those considered in Eq. (6.10), these would be included in the same way. Then if the *RMSE* is too large, compared to a single reference value (like a norm of $\mathbf{y}^r$) the weights and the parameter values in the activation functions are changed to reach a sufficiently small error. Various standard solution schemes are used for this optimization, like those based on steepest descent procedures, see for example, J. Nocedal and S. J. Wright [A], M.J. Kochenderfer and T. A. Wheeler [A] and S. Sra, S. Nowozin and S. J. Wright (eds) [A].

Looking at the overall process, we see that the use of the NN is based on an iterative scheme in which the input data is passed with weights to the first processing layer, filters (the activation functions) are used, after which the calculated data with new weights is passed to the next layer, again applying filters, and so on, until the output layer is reached. In the training phase, the thus reached output data is compared with known "desired" data, like obtained from other computations (e.g. finite element simulations) or physical measurements, and if the error in the prediction is too large (measured in a norm, see (6.10)), automatically adjustments to the ingredients of the NN are applied (the weights and activation functions are adjusted using optimization algorithms), and the process is repeated. One such sweep, passing through the entire training dataset, from the input to the output

through the NN is called an epoch and hundreds or thousands of epochs are typically carried out to reach convergence.

The NN described above is a relatively simple *Feed-Forward NN*, abbreviated as FFNN. It is a "simple" FFNN because of the assumptions used (and already mentioned). We assumed that the same number of nodes, *m*, is used in each processing layer while in practice, the processing layers may have different numbers of nodes. We also assumed that *m* output data is sought, when in practice the number of output items may be different than the number of nodes used , and furthermore we assumed that no final weights are applied to the data that is passed from the last hidden layer to the output. Hence, we can see that many variations are possible, in particular when a large number of processing units and a large number of nodes are used.

Another important procedure is how the weights and values of parameters are updated. Here the chain rule of differentiation is employed, in which the predicted result, y_i^p , is differentiated with respect to the considered weight or value of parameter in the activation function. This differentiation requires to proceed backwards through the NN from the result value to the specific weight (or parameter value). The calculation described is embedded in the optimization of the NN procedure.

The process we described above focused on the training of the NN with the objective of employing the NN in all its subsequent use on different (but reasonable, thus applicable) input data, see Section 6.1.1. However, we also want to ensure that the training is performed in the correct and most effective way, and here the validation is important, frequently performed during the training. Once the training and validation have been carried out, the testing is

important. Assume that the training uses *p independent data*, the validation uses *q* independent data with usually $p > q$, and that *t* independent data are used for testing. Note that the test set is used to evaluate the performance of *the trained* neural network. Both, the validation set and test set are "prior unseen data" and do not directly affect the training of the NN. It is important that in this complete process all sample data are *independent,* that is, all data are not dependent on any other data of the process.

Considering the validation during the training in more detail, assume that thousands of epochs are needed for convergence in the training, then the validation may be carried out each 20 or 40 epochs, to see whether the result so far reached in training is still reasonable. Since the data used for training and validation are independent of each other, but applicable to the problem considered, the validation loss will hardly be exactly the same as for training but it should be reasonably close. The loss can be smaller or larger for the validation because not only different but also less data is used. The purpose of the validation is, for example, to identify whether an *overfitting* of the data is encountered. This may happen when too few data points are available for the training, or with *outliers* (data out of the general expectations) to fit the weights and parameter values. The objective is always to obtain a *parsimonious* result (the simplest and yet largely applicable result).

Once the training and validation have been successfully completed using some training and validation data, the NN is tested employing the additional independent *t* data. The result should be reasonable, and of course be "in the space of predictability". This means that if the NN has been trained and validated for a particular purpose, using a given "space of input and output" training and validation data, then the NN can be used for prediction of unknown data in that space and good results should be expected.

While FFNN can be employed in many applications, there are important other NN that may be

more appropriately used, like *Convolutional* NN (CNN), *Bayesian* NN (BNN) and *Recurrent* NN (RNN), see for example T. Rabczuk and K.J. Bathe[A]. Some applications are found in classifying photos using CNN, in incorporating probabilities into data using BNN, and language recognition using RNN. All these NN can be explored for use in finite element analysis, see Section 6.3.1.

The development of NN for simulations also opens the possibility of using hybrid approaches, in which certain parts of a domain are modeled with NN because only a global response is needed and other parts, where detailed stress predictions are needed, are modeled with finite element procedures. An example of a solution might be the earthquake analysis of a building with soil structure interaction, in which the large soil domain is modeled using NN and the structure using finite elements. In this way it is possible to draw on the strengths of each of these approaches.

For our purposes, to use finite element procedures with NN in ML, we next mention some significant shortcomings of the otherwise useful FFNN when employed as above. These shortcomings spurred a very important development, namely, the embedding of physical laws into the NN procedures. The NN thus established are referred to as physics-informed neural networks and physics-informed neural operators.

6.2.3 Physics-informed neural networks and operators

The FFNN we discussed above are very useful for many applications, however we need to seek improvements for use in computational mechanics. The main reasons can be summarized as

follows.

The FFNN requires a large amount of data for training, validation and testing, and, as mentioned already, many iterations (epochs) may be needed for convergence. Hence, the cost of the solution can be prohibitive because considerable computer resources are used. Then although having performed thousands of iterations to solve a mechanics problem, the result may not be useful in that the solution does not satisfy the well-known laws of mechanics. Indeed, the obtained solution may be unphysical. Also, rerunning the solution with just slightly changed input data may give a much different result.

In addition, a major disadvantage using the NN referred to above is that analysts do not obtain any physical insight into the solution reached. The lack of physical insight inhibits a deeper understanding of the results and new, creative thinking regarding the problem solution. Specifically, in engineering design and analysis of a structure, the analysis results need to be understood in the context of the possible performance of the design in order to create a more advantageous solution.

The underlying reason for all these disadvantages is that while we try to mimic the working of the human brain with the artificial NN, the capabilities of the NN layers and nodes are far from being able to store, process, learn and creatively use information as do the neurons of the human brain. Particularly important in the decision process to reach a solution to an analysis problem is to apply the knowledge and experience that researchers have accumulated over hundreds of years. This knowledge and experience is embodied in the physical laws of mechanics and thermodynamics.

These observations were realized by researchers and developers, who then accomplished important advances by incorporating physical laws in the computations processed by NN, thus creating

Physics-informed NN, abbreviated as PINN, see M. Raisi, G.E. Karniadakis, [A], M. Raisi, P. Perdikis and G.E. Karniadakis [A, B, C], and A. Chakraborty, T. Wicks, T. Rabczuk, and X Zhuang [A], and O. Sedehi, A.M. Kosikova, C. Papadimitriou, and L.S. Katafygiotis [A]. In the use a PINN, we focus on solving specific partial differential equations like those governing Navier-Stokes fluid flows. The established physical laws are incorporated in the NN, see Section 6.3.1 and 6.3.2 for some illustrative solutions.

A more general approach is to use *Physics-Informed Neural Operators*, abbreviated as PINO. The distinguishing feature is that a PINO focuses on solving parametric partial differential equations by learning a solution mapping between input and output function spaces in a more general and unified manner, see N. Kovachki, Z Li, B Liu, K Azizzadenesheli, K. Bhattacharya, A Stuart, A. Anandkumar [A].

Related to the above developments, we also have those published in Z. Li, N. Kovachki, C. Choy, B. Li, J. Kossaifi, S. P. Otta, M. A Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli, and A. Anandkumar [A], G.E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang and L Yang [A], J. D..Toscano, V. Oommen, A. J. Varghese, Z. Zou, N.A..Daryakenari, C..Wu and G.E. Karniadakis [A], and A. Chakraborty, T. Wicks, T. Rabczuk, and X. Zhuang[A].

In all cases, the basic idea is to restrict the results of solutions using a NN to satisfy the geometry conditions and all the basic principles of mechanics, like equilibrium, conservation of momentum, and

conservation of energy. With this achieved, the number of iterations needed for convergence in a ML procedure is much reduced, the solutions make physical sense and can be more trusted to be useful predictions.

A natural approach is to use procedures close to those employed in finite element analysis but embedded in a NN. This approach can be pursued by using NN for specific tasks, like the constructions of the strain-displacement matrices, or even stiffness matrices, see e.g. J. Jung, K. Yoon, and P.S. Lee [A].

Another approach is to consider the known partial differential equations governing the physical problem and use interpolations, collocation, least squares or energy methods to satisfy all governing differential equations of the problem (to a prescribed tolerance in errors) like in traditional finite element analysis. Assuming that the minimization of
the potential energy is sought, the NN would be used to establish the specific set of equations to solve and optimization algorithms would then be employed to obtain the solution of the problem. This closely resembles the solution process of traditional finite element analysis, except that all computations are now performed using a deep NN. Since no "new" solutions or closer approximations to the actual response of a physical system are obtained, no further insight than using a finite element analysis is reached. However, once a NN has been trained (which may require large computational resources), the premise is that NN solutions to (similar) problems can be obtained computationally less costly than using another numerical method.

Since traditional finite element analyses are now performed very effectively to solve very complex problems in engineering and the sciences, emphasis is placed on reaching more effective NN techniques

for such analysis tasks. These objectives are pursued in J. Sirignano, K. Spiliopoulos [A] by establishing a NN to solve using a “deep Galerkin process” the governing differential equations, and in E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang and T. Rabczuk [A] using a deep NN for the minimization of the total potential. To attain that minimum, the total potential energy was established at many points over the geometry, where the points might b e generated using a finite element mesh generator, or could be generated like in a meshless scheme, see S. De and K.J. Bathe [A]. The NN used ramp shaped functions in the neurons, which act like finite element interpolation functions. These functions can also carry higher-order degrees of freedom, like used in the overlapping finite elements, see Chapter 3. J. A. A. Opschoor_ P. C. Petersen and C. Schwab [A] analyzed such ramp functions and identified good interpolation characteristics.

When seeking effectiveness in the NN solutions, it can be important to use mapping of domains to increase the speed of convergence and the accuracy of the solution, see V.M. Nguyen-Thanh, C. Anitescu, N. Alajlan, T. Rabczuk and X. Zhuang [A], W. Cao, J. Song and W. Zhang [A], and LW Chen and N Thuerey [A], the type of PINN to use is important, see for example, Y. J. Dai, Y. R. An, Z. Li, J.H. Zhang and C. Yu, C. [A] and possibly a neural operator is best employed (see above).

Many other research efforts are also still directed to increase the analysis capabilities and effectiveness of NN in ML procedures for general analyses of structures and fluids. While traditional finite element and finite volume methods are in general still more reliable and effective for applications in industries and the sciences, the above brief discussion shows already the tremendous breadth, possibilities and in particular the potential of using NN.

An area with great potential is also the modeling in biological and biomedical sciences,

see M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold and E. Kuhl [A].

There is another important point in the development of a NN for common analyses, namely the reliability of the NN to *always* reliably give valuable results (after the NN has been trained, validated and tested). As discussed above, the overall approach is that a NN will use a highly nonlinear procedure to obtain a solution, even when the analysis problem considered is a linear analysis. Hence the verification of any NN for a wide range of industrial solutions is a particular difficult task. In finite element analysis, we have the patch tests, which are widely used to verify finite element techniques, see for example Sections 2.2.3, 3.2.1, 4.1 and K.J. Bathe [A]. When training a NN, also validation during the training and testing thereafter are performed but each of these processes are completed through nonlinear solutions. Ideally, a stronger verification would be possible.

Considering contributions in analyses through ML, in general, strong achievements are those *advancing* and enlarging *the* possible solution possibilities – that is, for the analyst to reach solutions that with traditional finite element procedures are very difficult or impossible to reach, for example, due to modeling restrictions in geometries (numerous geometry details to be captured), material complexities (complex inhomogeneities to be represented) or complex loadings (possibly of stochastic character). If ML procedures are used to capture such phenomena while satisfying the basic laws of mechanics, the field of traditional finite element analysis is much enriched. In addition, there are the advances offered through the ability to process huge amounts of data (like in the use of digital twins and structural health monitoring).

Below we consider an important area to enrich finite element procedures through the use of physics-based NN, namely the development of more complex material descriptions, see Section 6.3.1.

However, ML procedures can also complement finite element procedures in other valuable ways, like in the solution of optimization problems, in the use of digital twins and in structural health monitoring, as briefly addressed in Sections 6.3.2 to 6.3.4.

6.3 Some applications of NN with finite element methods

The objective in this section is to briefly discuss and present the use of NN with finite element procedures. In these applications we can see some possibilities of using ML in computational mechanics, and in particular in finite element analyses.

The presentation given in this chapter focuses only on some applications with many more starting to develop. We consider here briefly the use of ML in constitutive modeling, in optimization (we recall that optimization procedures are inherently already employed in NN solutions), in digital twins and in structural health monitoring. These are already large important fields for use of ML but we can visualize much more usage, like in identifying recommended analyses, depending on the structure considered and experiences documented in databases. The ML procedures can look through massive amount of data to identify recommended analyses for a specific structure. This possibility can be of much value because, for example, considering earthquake loading, structures can be affected quite differently in different seismic zones. In addition to such overall recommendations obtained by ML procedures, when looking through available data also details to reach more effective analyses

might be established. The ML procedures may identify suggestions in modeling, selection of material models, load applications, and for specific applications, identify optimal values of solution parameters, for example, in direct time integration schemes and parallel processing.

With finite element methods now well established and widely used in engineering and the sciences, we can also think of hybrid approaches to leverage the "best model" in using finite element methods and ML techniques. For certain analyses, ML procedures may be employed in the areas of analysis which do not require very accurate solutions while finite element procedures are used for those regions where accurate stress predictions are needed. For example, in fluid-structure interaction problems, the fluid may be modeled using ML methods when only the pressure on the structure is needed, and the the structural response would be solved using traditional finite element procedures. Further possible applications are found in soil-structure interaction analyses.

6.3.1 Constitutive modeling using physics-based NN

One use of ML is in the development of more encompassing constitutive equations. The response of complex materials can now only be described to a limited degree although many researchers have focused much effort on obtaining more general descriptions of materials with complex internal structures, to take into account short term and long-term effects, and general loading conditions, including temperature effects. Below we consider two areas of development in which some achievements have been reached.

Large strain elastic material behavior

The development of NN to represent large strain elastic material behavior has already been pursued with some success, specifically to model the nonlinear elastic behavior of rubber-like materials, for a review of models, see MC Boyce and EM Arruda [A] There is a large variety of materials that need to be modeled, that is, isotropic and anisotropic rubber materials as used in the motor car and consumer industries, soft biological tissue as found in organs and muscle, and many synthetic rubber-like materials which we encounter abundantly in everyday life.

A very widely used approach is to assume a material behavior based on polynomials or other functions spanning over the complete stress-strain domain to represent the physical test data. This idea is in essence used in the classical models like the Mooney-Rivlin and Ogden models, see KJ Bathe [A].The constants used in these material models are frequently calculated by regression algorithms, see Section 6.1.1.

A generalization of models based on polynomials and other functions spanning over the complete stress-strain data is proposed by K Linka and E Kuhl [A] using NN. In this development, the physical behavior is represented by polynomials and exponential functions like for classical models but with more terms and in a generalized form. For example, the NN may process data using the various expressions or functions of the classical models in combination. The material constants in the functions are obtained from the weights used in the NN. The approach employs known input data for the NN (physically measured or otherwise established data) to calculate the free energy as output from the NN, from which the stress state is calculated. Once the free energy and prediction of stresses converged to good results in the training and validation, and then also in the testing, the NN represents in essence a very general nonlinear elastic material model – but only as general as can be represented by the functions used

in the NN. This shortcoming leads to the idea of using local functions to represent the material behavior.

To satisfy the basic laws of mechanics using local representations, one approach is to use a reduced expression of a general hyperelastic material behavior, as introduced by T Sussman and KJ Bathe [C]. In this model, the strain energy is expressed in a separable form using the principal logarithmic strains. The physical test results of force versus stretch are directly interpolated by a number of local splines. The approach allows for a very general behavior to be represented as seen in physical test data, including softening, followed by hardening, subsequent local softening, and so on. The constants defining the local splines connected in sequence are calculated by the solution of a set of equations.

The Sussman-Bathe model assumes isotropic behavior in large strains. This restriction was removed by M. Latorre and F.J. Montáns [A, B, C], who extended and refined the development to include anisotropic hyperelastic behavior and included a data reduction procedure based on smoothing regression B-splines. This data reduction procedure is effective to obtain smooth and stable stored energies, also in the presence of fuzzy experimental data or data from multiple specimens. Anisotropy can be important , for example, when modeling biological materials, see M. Latorre, E. Peña and F. J. Montáns [A], and X. Romero, M. Latorre and F. J. Montáns [A].

In the following, we focus on presenting a very simple example (just for illustrative purposes) to show how a NN can be used to predict a material behavior. We consider a rubber-like incompressible material.

Let us assume that the incompressible material can be described by the invariants ${}_0^tI_1$ and ${}_0^tI_2$ with ${}_0^tI_3 = 1$ of the Cauchy-Green deformation tensor, see K.J. Bathe [A],

$$ {}_0^t\mathbf{C} = {}_0^t\mathbf{X}^T \, {}_0^t\mathbf{X} \tag{6.11} $$

established from the deformation gradient ${}_0^t\mathbf{X}$. Here the superscript *t* represents the time considered (that is, the state of deformation considered) and the subscript *0* represents the original state. The objective is to calculate the free energy function $\psi({}_0^t\mathbf{X})$ which is used to calculate the stresses.

Let us furthermore assume that the material can be modeled as a generalized Mooney-Rivlin model given by

$$ \begin{aligned} \psi({}_0^tI_1, {}_0^tI_2) &= C_1({}_0^tI_1 - 3) + C_2({}_0^tI_2 - 3) + C_3({}_0^tI_1 - 3)^2 + C_4({}_0^tI_1 - 3)({}_0^tI_2 - 3) \\ &+ C_5({}_0^tI_2 - 3)^2 + D_1\left(e^{D_2({}_0^tI_1 - 3) + D_3({}_0^tI_2 - 3)} - 1\right) \end{aligned} \tag{6.12} $$

where $C_1, C_2, C_3, C_4, C_5, D_1, D_2$ and D_3 are constants, usually determined from physical experiments.

We aim to establish a NN to capture a somewhat general physical material behavior and express this behavior through a material model implicitly given in the NN by the expression (6.12) . For this purpose we use the simple NN presented in Fig. 6.4. Since the NN solution shall correspond to the material description implied by Eq. (6.12), and this Mooney-Rivlin model is based on physics, we actually use

implicitly a simple physics-based NN procedure.

The functions used to evaluate the free energy in Fig. 6.4 are denoted as f_1, f_2 and f_3 where f_i may contain any useful expression, like those in the classical models. The weights w_{ij} in products multiply the functions and determine the physical material constants. We can see now that once convergence has been reached considering a specific case of material, the NN in fact represents a material model – of course with the assumptions embedded in the functions f_i, $i = 1,2,3$.

Another way of looking at the procedure is to think of the functions f_i as basis functions that span a subspace, and the NN evaluates the best approximation to describe the material in that subspace. The input is the deformation gradient ${}^{t}_{0}\mathbf{X}$ for many different stress states, the output is the free energy function described by the functions f_i and the weights.

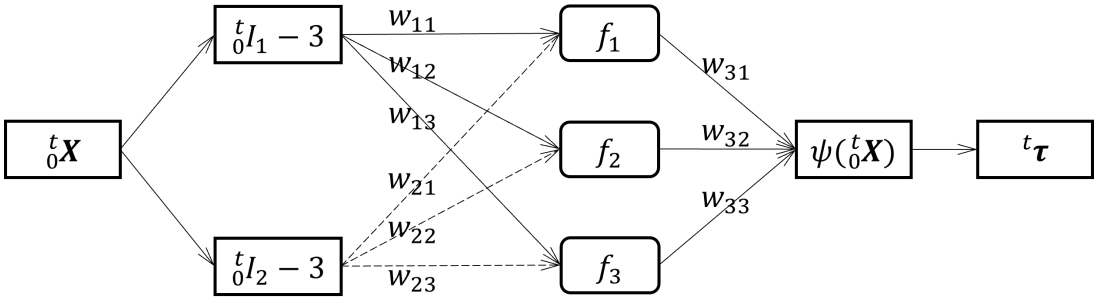


Fig. 6.4 A NN to calculate the free energy function $\psi({}^{t}_{0}\mathbf{X})$ and output the Cauchy stresses ${}^{t}\boldsymbol{\tau}$

Looking closer at the simple NN in Fig. 6.4 , it only consists of one processing layer of three nodes, with each node processing on the simple functions $f_1,\ f_2, f_3$, respectively. The input ${}^{t}_{0}\mathbf{X}$ is weighted and the result from the processing layer is also weighted before being passed to the output. If we define $x_1 = ({}^{t}_{0}I_1 - 3),\ x_2 = ({}^{t}_{0}I_2 - 3)$, and use $f_1 = x,\ f_2 = x^2, f_3 = \exp(x) - 1$ in the NN of Fig. 6.4, we can see that the following holds

$$\psi({}^{t}_{0}\mathbf{X}) = w_{31}(w_{11}x_1 + w_{21}x_2) + w_{32}(w_{12}x_1 + w_{22}x_2)^2 + \mathrm{w}_{33}(\mathrm{e}^{w_{13}x_1 + w_{23}x_2} - 1) \tag{6.13}$$

where we note that, for example, the first term on the right-hand side is a result of the weights w_{11} and w_{21} operating, respectively, on $({}^{t}_{0}I_1 - 3)$ and $({}^{t}_{0}I_2 - 3)$ as shown in Fig. 6.4 and the weight w_{31} is operating on the result prior to being passed to $\psi({}^{t}_{0}\mathbf{X})$.

In Eq. (6.13) each product of constants seems to be only one unknown, but the NN will need to recognize how to establish all nine constants to obtain the eight unknowns in the material model given in (6.12). The expression is used to calculate the stress state corresponding to the deformation gradient that was input, see KJ Bathe [A], and this stress state is compared to the known value (obtained through another analysis or physical data) using the loss function, see Eq. (6.10). Once the iteration has converged, after possibly many epochs automatically adjusting the weights and other parameters in the NN, during the training, validation, and testing, we have in the NN a material model with the material constants given by the weights. A restriction of the model is that it is based on the functions $f_1,\ f_2, f_3$ – the basis – here with the linear, quadratic and exponential terms. The

solution will be in the subspace spanned by these base vectors, where it is important to note that the the functions $f_1,\ f_2,\ f_3$ are smooth and valid for $-\infty \le x \le +\infty$ where $x = x_1$ or x_2 .

To illustrate the ML solution, using the NN in Fig. 6.4.to obtain material constants, we need to have physical test results. Since we do not have such results obtained in a laboratory, we use as physical test results the data generated in a "pseudo-physical way", that is, by a finite element analysis using ADINA. Fig. 6.5 shows the "physical specimen" subjected to loading which is increased resulting into a large strain response.

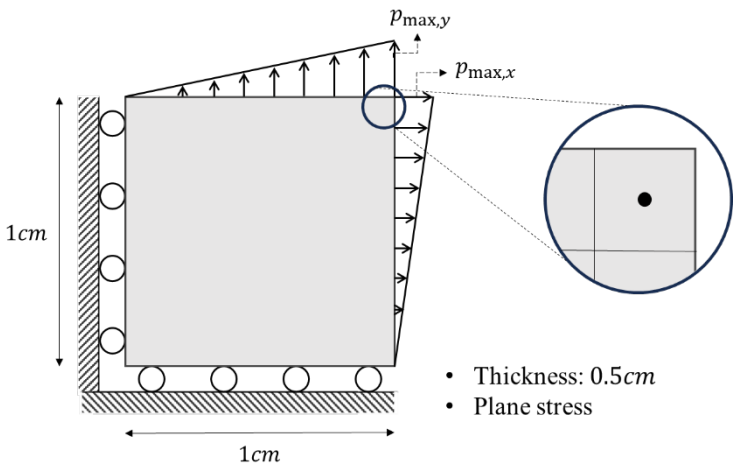


Fig. 6.5 Two-dimensional specimen used to generate the training, validation and test data

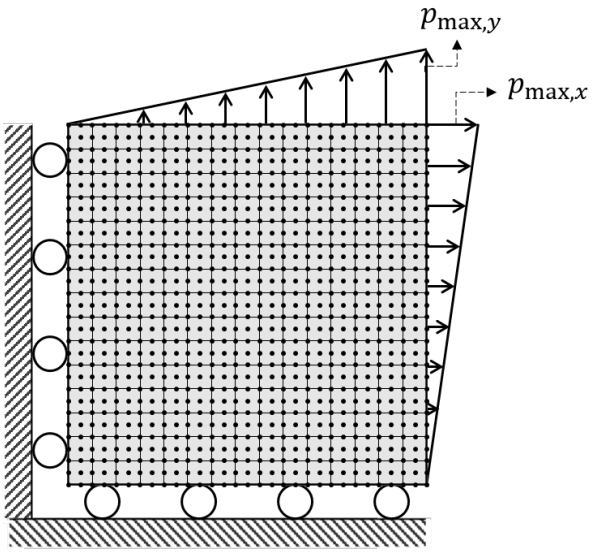
The maximum pressure values $p_{\max,x}$ and $p_{\max,y}$ are varied for different load cases. To obtain the "pseudo-physical data" we use for the specimen in Fig. 6.5 , just for an experiment, the *basic* Mooney-Rivlin material model given by

$$\psi({}^t_0I_1, {}^t_0I_2) = C_1({}^t_0I_1 - 3) + C_2({}^t_0I_2 - 3) \tag{6.14}$$

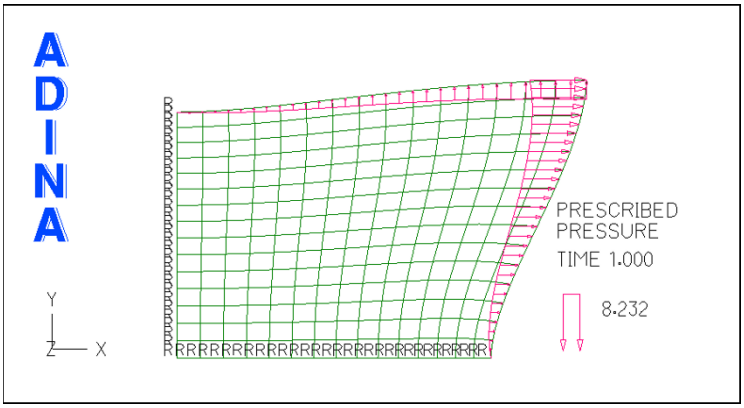
with the constants $C_1 = 1.908\ [MPa]$ and $C_2 = 0.65\ [MPa]$.

The *general* Mooney-Rivlin law in Eq. (6.12) that we want to represent in the NN of Fig. 6.4 clearly contains the *basic* Mooney-Rivlin law and hence also represents the material behavior which we will measure in our pseudo-physical experiment.

We discretize the specimen by a mesh of 15×15 9-node elements and apply the loads using the load-displacement control (LDC) method available in ADINA, see Bentley Systems, Inc. [A]. Fig. 6.6 shows the finite element model used with the applied loads. Then measuring the deformation gradient and the Cauchy stress in the center of the corner element of the specimen, see Fig. 6.6, for different load levels reached, we generate 20 cases for training the NN, eight cases for validation and ten cases for testing. Table 6.1 shows some typical values generated.



(a) The specimen in its original configuration with the displacement boundary conditions and the loading in the x- and y-directions



(b) A typical deformed configuration reached, notice that the applied horizontal pressure is larger than the vertical pressure; the roller supports are shown by "R"

Fig. 6.6 The specimen of Fig. 6.5 discretized using a mesh of 15×15 9-node elements

Table 6.1 Data obtained in the "pseudo-physical laboratory" (using ADINA)

Training data

	Deformation Gradients					**Cauchy Stresses**		
	$({}_0^t\mathbf{X})_{11}$	$({}_0^t\mathbf{X})_{12}$	$({}_0^t\mathbf{X})_{21}$	$({}_0^t\mathbf{X})_{22}$	$({}_0^t\mathbf{X})_{33}$	$({}^t\boldsymbol{\tau})_{11}$	$({}^t\boldsymbol{\tau})_{12}$	$({}^t\boldsymbol{\tau})_{22}$
1	0.550141	-0.645791	0.664779	0.909481	1.075670	-2.454420	-1.179020	0.468414
2	0.478127	-0.587594	0.572784	1.631980	0.895368	-2.365400	-3.328240	9.379630
3	1.428480	0.400373	-0.410126	0.627027	0.943484	5.812440	-1.665100	-2.341140
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	1.704700	0.071072	0.039126	1.001320	0.586795	13.120600	0.587802	4.990380

Validation data

	Deformation Gradients					**Cauchy Stresses**		
	$({}_0^t\mathbf{X})_{11}$	$({}_0^t\mathbf{X})_{12}$	$({}_0^t\mathbf{X})_{21}$	$({}_0^t\mathbf{X})_{22}$	$({}_0^t\mathbf{X})_{33}$	$({}^t\boldsymbol{\tau})_{11}$	$({}^t\boldsymbol{\tau})_{12}$	$({}^t\boldsymbol{\tau})_{22}$
1	-0.623026	-1.134640	0.759842	-0.190059	1.019830	2.845660	-1.332070	-2.643290
2	-0.282690	0.717225	-1.124470	-0.658351	1.007450	-2.564490	-0.792446	3.102630
3	0.202199	1.003660	-1.139740	-0.552096	0.968729	-0.152406	-3.951080	2.645610
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	1.430350	0.204033	-0.211020	0.736935	0.911470	5.725920	-0.741617	-1.617710

Test data

	Deformation Gradients					**Cauchy Stresses**		
	$({}_0^t\mathbf{X})_{11}$	$({}_0^t\mathbf{X})_{12}$	$({}_0^t\mathbf{X})_{21}$	$({}_0^t\mathbf{X})_{22}$	$({}_0^t\mathbf{X})_{33}$	$({}^t\boldsymbol{\tau})_{11}$	$({}^t\boldsymbol{\tau})_{12}$	$({}^t\boldsymbol{\tau})_{22}$
1	1.677860	0.060982	-0.030583	0.857512	0.694132	11.155300	0.004348	1.903380
2	1.351640	0.050613	0.039230	1.176320	0.629730	8.032140	0.487563	6.107960
3	0.925559	-0.081514	0.081921	1.029230	1.042430	-1.162000	-0.042214	-0.102006
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	0.338624	-0.796028	0.771197	1.390670	0.921823	-1.650630	-4.162230	7.110040

For the NN solution using the network in Fig. 6.4, we identified that a scaling factor SF is best applied to the functions $f_1, f_2, f_3,$ for better convergence and accuracy, and we choose by numerical experiments $SF = 1000$. The activation functions are scaled by SF^{-1} and the weights in the last layer are scaled by SF. This scaling results in better convergence and accuracy for the weights and hence for the estimates of the material constants.

The results obtained are shown in Fig. 6.7 for the loss functions of training and validation as a function of the number of epochs used. The loss is calculated using Eq. (6.10) with the reference values given by the known Cauchy stresses at the center of the corner element shown in Fig. 6.5 (and Fig. 6.6) and the predicted stresses given by the calculations in the NN from the input of the deformation gradient. In the solution process, the weights are optimized to continuously decrease the loss as the number of epochs increases.

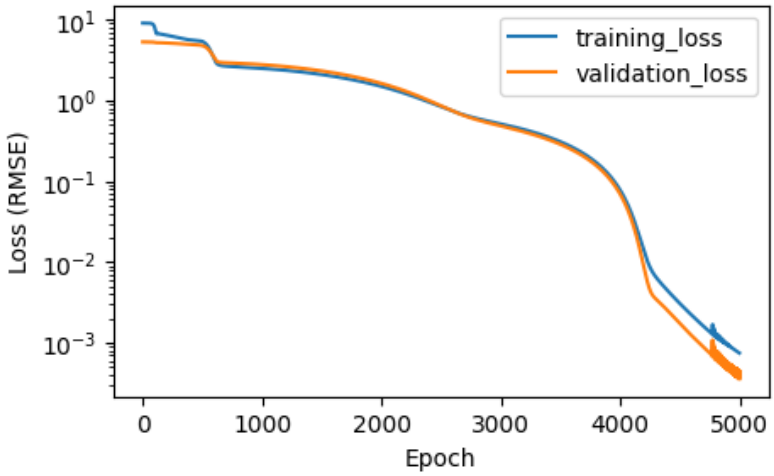


Fig. 6.7 Training and validation loss in the evaluation of the material behavior of the specimen of Fig. 6.5

With the training and validation completed, the testing of the NN was performed (hence employing the calculated weights obtained in the training and validation) using the test data given in Table 6.1, The *RMSE* between the given Cauchy stresses in Table 6.1 and the predicted stresses (with the input of the deformation gradients) by the NN was 0.0006358 [MPA], which is an acceptable result.

A very interesting result is that during the training and validation we can use the weights calculated by the NN to establish the material constants of the law in Eq. (6.12). The constants are obtained by comparing the right-hand sides of Eqs. (6.12) and (6.13)

$$C_1 = w_{31}w_{11}, \qquad C_2 = w_{31}w_{21},\ C_3 = w_{32}w_{12}^2, \qquad C_4 = 2w_{32}w_{12}w_{22},\ C_5 = w_{32}w_{22}^2,$$
$$D_1 = w_{33},\ D_2 = w_{13},\ D_3 = w_{23} \tag{6.15}$$

We thus can plot the results for the constants as a function of the epochs completed as shown in Fig. 6.8.

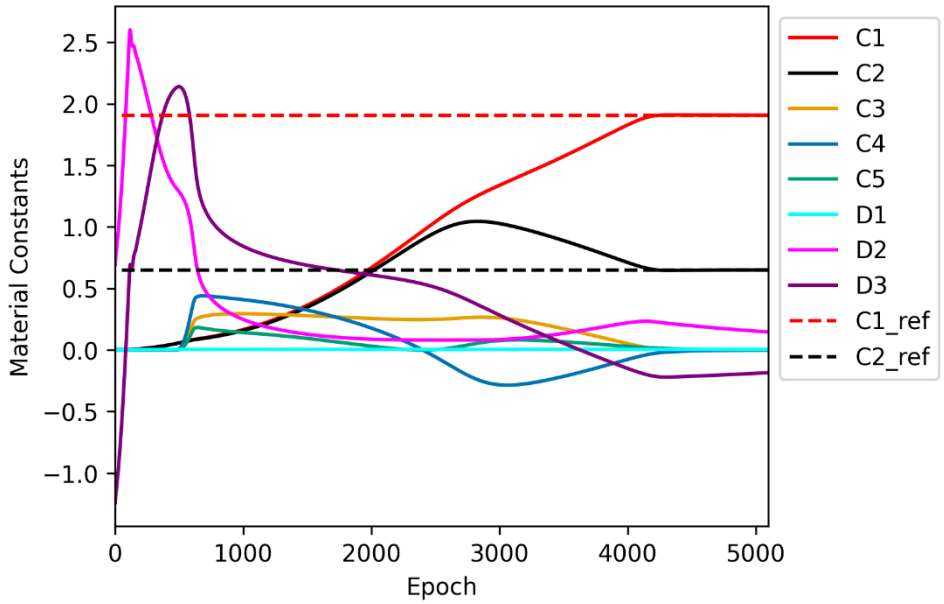


Fig. 6.8 Calculated material constants as a function of completed epochs using Eq. (6.15)

Once convergence has been reached, shown by a small loss in Fig. 6.7 and convergence of the curves in Fig. 6.8, Eq. (6.15), gives the predicted material constants. We then have $C_1 = 1.908,\ C_2 = 0.650$ $C_3 = 0.001,\ C_4 = -0.002, C_5 = 0.001,\ D_1 = 0.004,\ D_2 = 0.153,\ D_3 = -0.190$. These values are close to the actual constants used to generate the material response in our "pseudo-physical laboratory" in that $C_1 = 1.908$ and $C_2 = 0.65$ are well predicted and the other constants are introducing an almost negligible effect in the material law of (6.12) with $D_1 = 0.004$ almost equal to zero. We could expect such results because, as mentioned already above, the functions $f_1,\ f_2,\ f_3$ used in the NN "contain" the *basic* two-parameter Mooney-Rivlin material law.

While this is a somewhat simple solution case, we can now study whether and how the NN can be used to establish material constants also of more general material behavior. In all cases, if the generalized Mooney-Rivlin law of Eq. (6.12) can describe the material behavior, we can expect that also the NN of Fig. 6.4, once trained, can be used to represent the material response.

However, there are at least two shortcomings in engineering practice using models with functions like those in the above example. Firstly, an appropriate known analytical material law needs to be chosen that at least approximately can represent the material behavior and then, assuming a NN shall be used, the constants need to be established using an appropriate NN that ideally can give a more encompassing prediction of the material behavior. This requires that the NN "contains" the possibility of representing the chosen material law plus "additional behavior". Secondly, rubber-like materials can exhibit *local* changes in behavior that need to be represented by *local* functions. Global functions spanning from large compression to large tension can hardly represent such variations in behavior.

Let us recall that local changes are also difficult to capture in a general stress analysis using the classical Rayleigh-Ritz process with global functions, see K.J. Bathe [A]. When used in the plane-stress analysis of a plate with a hole, the local stress changes around the hole can hardly be captured using the global function spanning over the complete analysis domain. This difficulty spurred the development of the finite element method in which local functions are used (the finite element interpolation functions) to interpolate the displacements locally. As a result, the stress analysis of the plate with a hole, although some time ago regarded a difficult analysis problem, is now a rather simple problem to solve with the finite element method. Similarly, using the local approach of T. Sussman and K.J. Bathe [C], M. Latorre and F.J.

Montáns [A, B] more complex material behaviors can be modeled, in particular when used in a NN.

While much has been achieved already using NN in modeling general elastic material behavior, see references given above, further advances are sought in particular as more complex materials are developed, for example materials with different constituents, for different purposes in their use.

Inelastic material behavior

The modeling of inelastic materials is much more difficult, in that the history which the material has undergone needs to be captured, see M. Kojic *a*nd K.J. Bathe [A]. Some early efforts using NN to solve problems in plasticity and fracture were pursued by P.S. Theocaris and P.D. Panagiotopoulos [A], P. D. Panagiotopoulos and Z. Waszczyszyn [A] and Z. Waszczyszyn and L Ziemianski [A] . Since then, much further research was published but the description of general nonlinear geometric and material response using NN is very difficult and still in a relatively early development – when considering to reach reliable and effective material response predictions. As examples, we refer to M. Miñano and F.J. Montáns who used the WYPiWYG (an acronym for: What You Prescribe is What You Get) modeling approach to predict quite general damage but only for soft materials [A], D. W. Abueidda, S. Koric, N.A. Sobh and H. Sehitoglu, H [A], who pursued the use of deep learning to represent inelastic material behavior, J.N. Fuhg, C.M. Hamel, K Johnson, and R Jones and N Bouklas [A], who investigated the representation of inelastic behavior having only limited data available, and D. W. Abueidda, Z. Kang, S. Koric, K. A. James and I.M. Jasiuk [A] who pursued the optimization of elastoplastic architected materials to reach enhanced performance.

The use of recurrent neural networks (RNN) seems particularly promising because variables can be used to store and update the history that a material has been subjected to, see for example, A. L. Frankel, R.E. Jones, C. Alleman and J.A.Templeton [A], F. Ghavamian and F. Simone [A], C. Bonatti and D. Mohr [A] and C. Bonatti, B. Berisha and D. Mohr [A].

The great potential of using NN to describe inelastic material behaviors needs to still be explored. Classical material models are well established, have good predictive capabilities, and are widely used in research and industry. However, the disadvantage of these models is that they focus only on certain aspects of material behavior, like the classical plasticity models focus on material yielding, cyclic stress and strain behaviors, temperature and possibly time effects. However, considering materials used in practice, the initial behavior, long-time effects, like changes in stiffness due to micro-fracture, rusting, etc. should also be considered in detail. A good example is the constitutive description of reinforced concrete materials, involving different compression and tension behaviors, crushing and cracking in three-dimensional conditions, temperature and aging effects, reinforcing steel with slippage in certain loading conditions, and so on. To describe all these effects and their interactions in the way classical material models are formulated would lead to a very complicated material model that would be very difficult to implement in a finite element program. The difficulties reaching a somewhat general model of concrete still tractable for use to only represent the overall physical behavior resulted in the simple use of general input curves, see KJ Bathe and S Ramaswamy [A]. For a more general and much more useful concrete model, ML procedures provide much potential.

More generally, we can foresee that ML will provide the means to establish descriptions of much more

encompassing inelastic material behaviors, particularly using physics-based NN or physics-based ML. The ML procedures might use classical descriptions in very general forms to satisfy the required criteria of physics, in conjunction with physical test results, and field measurements accumulated in a database. The training, validation and testing would be carried out as described above to reach a material model of much wider applicability than currently available.

6.3.2 Fluid flow using physics-informed NN for structural optimization

As mentioned in Section 6.2.3, a very important further development in the use of NN are the use of *Physics-informed NN* (PINN) and *Physics-informed neural operators* (PINO) , see M. Raissi, P. Perdikaris and G.E. Karniadakis [C] and N. Kovachki, Z Li, B. Liu, K Azizzadenesheli, K. Bhattacharya, A Stuart and A. Anandkumar [A]. The objective is here to obtain the solution of partial differential equations through the use of a neural network or neural operator. The input is the information
required to describe the problem and the output is the solution. The input varies depending on the problem solved and the neural network or operator used. The key is here that once a NN has been trained,
it can be used to obtain solutions computationally very inexpensively – within the space of problems considered. For example, if we consider a Navier-Stokes fluid flow problem through a certain geometry,
we may represent the geometry and boundary conditions in some generic form, and pose the problem to solve for the velocities and pressure. The physical conditions are imposed on the network to be
trained to obtain accurate solutions. In the realm of PINN and PINO, the basic laws of mechanics are imposed to be satisfied and as a consequence in training and testing is possible using a reasonable number of test cases.

In particular, since after training the NN can solve problems very effectively, it can directly be

employed in optimization processes, for which the tools are already available in the ML procedures, see T. Rabczuk and K.J. Bathe [A].

As an illustrative example using a simple approach, which still requires rather many samples for training and testing, we consider a NN *solution-based approach* of the Navier-Stokes equations governing the flow of an incompressible fluid. The equations governing the physical situation are then the momentum equations and incompressibility condition which when considering a fluid flow described in a Cartesian framework are, see K.J. Bathe [A],

$$\rho\left(\frac{\partial v_i}{\partial t} + v_{i,j}v_j\right) = \tau_{ij,j} + f_i^B$$
$$\tau_{ij} = -p\delta_{ij} + 2\mu e_{ij} \qquad (6.16)$$

and

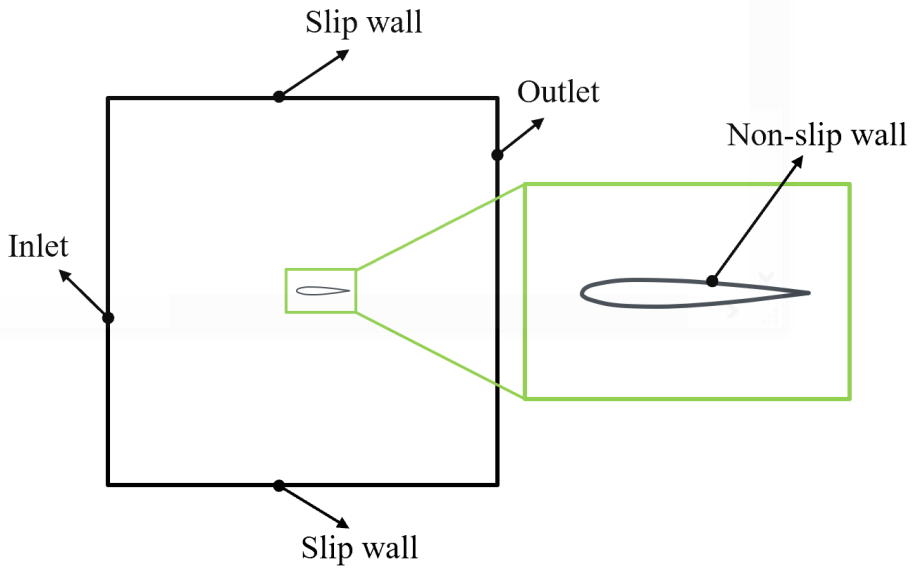
$$v_{i,i} = 0 \qquad (6.17)$$

where the v_i, $i = 1, 2, 3$ are the velocity components in the Cartesian system, ρ is the fluid mass density and μ the viscosity, p is the pressure, the τ_{ij} are the stress components, e_{ij} the velocity strains, and the f_i^B are the externally applied body force components, all used with the standard indicial notation.

For the example solution, we consider the steady-state fluid flows around the geometry of an airfoil. The problem is schematically shown in Fig. 6.9 with a typical airfoil geometry, given by the CST functions, see B.

M. Kulfan [A]. Our aim is to train a NN to obtain the solution for the fluid flow around the airfoil very quickly so that, for example, an optimization process to find the optimal airfoil geometry corresponding to a given objective (like the maximum lift) could be performed very efficiently. The solution we pursue is based on a data-driven approach (a considerable amount of data is used) incorporating the physics of the problem (all data is based on the solution of the Navier-Stokes equations). In that sense, therefore, the solution is obtained using a physics-informed NN.

The training of the NN uses the results of the fluid flow and pressure solutions around the airfoil for many airfoil geometries given by the CST functions; hence many solutions of Eqs. (6.16) and (6.17) (without the time effects) need to be carried out prior to the training of the NN. While any numerical solution scheme



Inlet condition:

$$V_x = U\cos(\alpha)$$

$$V_y = U\sin(\alpha)$$

Fig. 6.9 The steady-state fluid flow problem considered, schematically; $U =$ 51.48 [m/s], $\alpha = 2$ degrees; the free-flow pressure = 1 atm. The airfoil geometry is given by CST functions, see B. M. Kulfan [A]

can be used to obtain the flow and pressure solutions, the AMORE procedure is particularly effective, see Section 3 6. In the solution, undistorted regular elements are used in the interior of the domain, while near the airfoil geometry distorted overlapping elements are employed without loss of solution accuracy, see J. Huang and K.J. Bathe [A], and W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R C. Mesquita [D], The geometry of the airfoil can be easily

varied to consider many airfoil samples. For the flow solutions, the mesh remains the same – the mesh of regular elements in the interior of the fluid domain does not deform (remains stationary) while the mesh of overlapping elements near the airfoil adapts to the new geometry with element distortions without significant loss of solution accuracy. Then also, for the next flow solution for airfoil $(i + 1)$, to speed up the iterations, the previously obtained solution for airfoil (i) can be used as the starting vector in the iterations to solve the nonlinear finite element equations for airfoil $(i + 1)$.

To establish the NN, 1,360 airfoil samples were used for training, the validation was performed using 131 samples and the testing with 30 samples. Hence a total of 1,521 solutions of the Navier-Stokes equations around 1,521 different CST airfoil geometries were obtained prior to the training, validation and testing of the NN. A mesh of 16,384 nodal (grid) points was used for the flow solutions.

A NN was then employed with a basis established through the Proper Orthogonal Decomposition (POD) of the

1,360 fluid flow solutions for training using the Singular Value Decomposition (SVD). We only use 1,360 fluid flow solutions for training because for validation and testing the additional 161 solutions need to be independent of the data used in training.

Also, since we are only interested in the pressure solutions, the SVD is only performed on the calculated pressure fields. The SVD is established by evaluating the eigenvalues and eigenvectors of the pressure fields similar to but more general than the process used in a mode superposition solution in dynamics, see K.J. Bathe [A], as we discuss below.

It is valuable to consider in more detail how the SVD is established. Having obtained the numerical solution of Eqs. (6.16) and (6.17) for the airfoil *case 1*, we extract the pressure solution at the 16,384 nodal points, normalize these pressure values by dividing them by the maximum pressure value encountered at any of the nodes and list these normalized values in the vector $\mathbf{p}_1^T$

$$\mathbf{p}_1^T = [p_1 \cdots p_n] \tag{6.18}$$

where p_i is the normalized pressure value at nodal point *i*, with $i = 1, \ldots, 16{,}384$. We do the same for each of the 1,360 airfoil samples used in training, hence we have 1,360 cases, and list the results in the rows of the matrix $\mathbf{P}^T$, which is thus a matrix of $1{,}360 \times 16{,}384$.

To obtain the SVD we solve for the lowest *q* eigenvalues and corresponding orthonormalized eigenvectors $\boldsymbol{\varphi}_i$ of the matrix $\mathbf{S} = \mathbf{P}\mathbf{P}^T$

$$(\mathbf{S} - \lambda_i \mathbf{I})\boldsymbol{\varphi}_i = \mathbf{0} \tag{6. 19}$$

and have

$$\boldsymbol{\Phi}^T \mathbf{S}\, \boldsymbol{\Phi} = \boldsymbol{\Lambda}; \quad \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \mathbf{I} \tag{6. 20 }$$

with the diagonal matrix $\boldsymbol{\Lambda}$ listing the calculated eigenvalues, $\boldsymbol{\Lambda} = diag\,(\lambda_1,...,\lambda_q)$, and the matrix $\boldsymbol{\Phi}$ storing the corresponding eigenvectors, $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_q]$. For this solution, the subspace iteration scheme discussed in Chapter 5 is effective.

We can now express any of the pressure 1,360 solutions used to establish $\mathbf{S}$ approximately as

$$\overline{\mathbf{p}} = \sum_{i=1,..,q} \alpha_i \boldsymbol{\varphi}_i \tag{6 21}$$

where the error between the values in $\overline{\mathbf{p}}$ and the "exact" nodal pressures $\mathbf{p}_{exact}$ of the corresponding numerical solution is small if q is sufficiently large. Of course, the α_i would need to be determined (for example, by minimizing the error between $\overline{\mathbf{p}}$ and $\mathbf{p}_{exact}$). The matrix $\boldsymbol{\Phi}$ represents the POD basis used in the NN solutions.

Returning to the discussion of the example we established the POD basis using q = 64. This basis is the means to obtain computationally very effectively the pressure for any possible airfoil geometry of interest once the NN with this basis has been trained, validated and tested. To train the NN in Fig. 6.10, we used the same 1,360 flow solutions as for the construction of the matrix **S**. Since q is smaller than 1,360 , there will be an error in the calculated $\overline{\mathbf{p}}$ compared to the $\mathbf{p}_{exact}$ but this error is minimized by the NN. We also used the additional 131

cases for the validation (again obtaining $\bar{\mathbf{p}}$ compared to the $\mathbf{p}_{exact}$) . The training and validation gave the loss functions as a number of epochs shown in Fig. 6.10. The mean square error (MSE) was calculated using all grid point values in all of the 1,360 cases for training and the 131 cases for validation. In the fluid flow and pressure solutions of the different airfoils, the nodal positions were fixed, so no interpolation was needed.

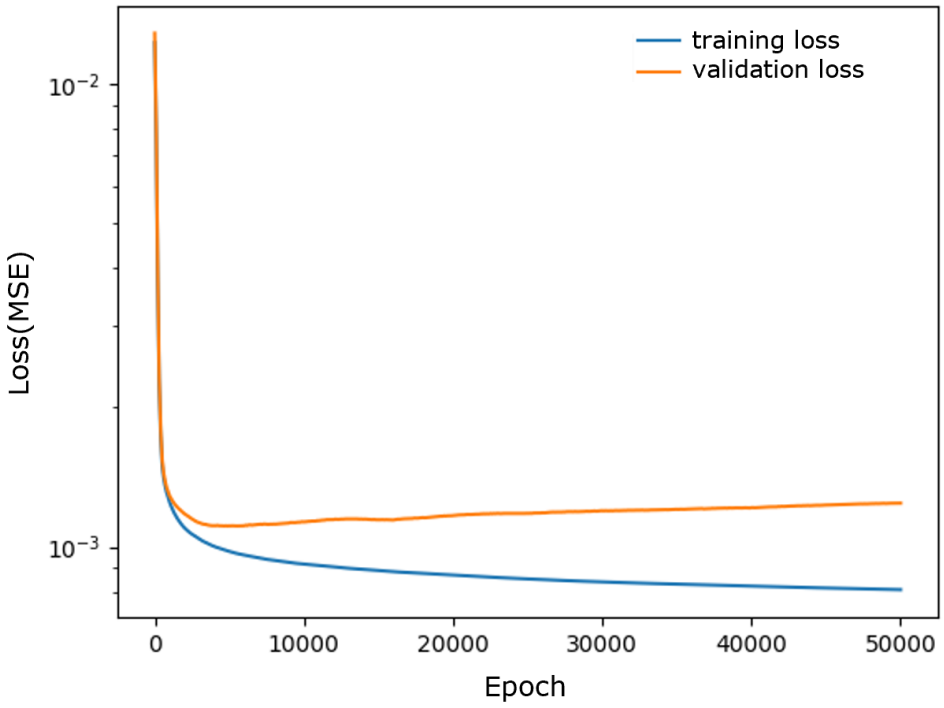


Fig. 6.10 Loss in training and validation of the example considered

The figure shows that both losses are reasonably small and hence the trained NN can next be used for testing, for which we employed the additional 30 cases and obtained the $RMSE = 1.24\times10^{-3}$ for the pressures along the airfoils (see Eq, 6.10). This is an acceptable error and hence we now have a NN to very efficiently calculate the pressure solutions due to the Navier-Stokes fluid flow conditions described in Fig. 6.9 around any airfoil of the general geometry given by the CST functions.

Of course, the initial training and validation uses a considerable amount of computational effort, but thereafter the pressure solutions are obtained very efficiently. Hence once the NN has been trained, it can be used very effectively in a general optimization process, like to find the optimum airfoil geometry for lift. The maximization (or minimization) process is performed as is usual in optimization, but the tools to use are already available in the NN and in general in ML procedures.

The key in the solution is to establish the POD basis and various analyses can be employed to obtain a basis effectively. But then, given the POD basis of pressures and nodal positions, there are many details that can be important for the NN to work efficiently, like the detailed design of the NN used, the weights and parameters employed, the interpolations of the pressure values, if needed, between nodal positions, and the search for the minimum error. Nevertheless, the above brief discussion of the example indicates already how a NN may be trained and then thereafter be used very efficiently to obtain pressure solutions.

It is interesting to compare the NN solution using the POD basis with the mode superposition solution of the dynamic response of a finite element model. In a mode superposition solution, only *one* finite element model is considered, the required frequencies and mode shapes are calculated, the new basis is given by the calculated mode shapes, the governing finite element equations are transformed to the new basis, the dynamic response is calculated in that basis and then transformed back to the degrees of freedom of the finite element model.

Hence in the traditional mode superposition solution, only *one* new basis for a single finite element model is established whereas in the NN solution briefly discussed above, we used 1,360 different solutions for the 1,360 different airfoil geometries to establish the basis vectors in $\boldsymbol{\Phi}$ (see Eqs. 6.19) and (6.20). These vectors thus represent the combined effect of all solutions for all 1,360 airfoil geometries considered. As a consequence, we

can look at the POD basis used in Eq. (6.21) as spanning “the subspace of most importance” for calculating the pressure fields of the airflows in the problem considered with any CST geometry of airfoils. This fact then, after validation and testing, enables the NN to solve automatically for the pressure field of *any* Navier-Stokes

fluid flow subject to the boundary conditions in the domain shown in Fig. 6.9 with different airfoils described by the CST functions.

6.3.3 Digital twins

While ML procedures can already be valuable and effective in certain applications, as we discussed above, the methods are frequently not yet competitive to achieve very accurate solutions in complex structural and fluid flow problems, like obtained with traditional finite element and other numerical methods. These traditional procedures have been available and been offered in commercial codes for some decades and are widely accepted, whereas ML procedures have only recently come to the forefront for engineering applications. In addition, there can be some skepticism, naturally, about using a ML approach in analyses and design practices, when this approach shall replace – and not only add to -- established practices. see M.Z. Naser [A].

The key is to integrate ML techniques with traditional analysis procedures in those areas where ML can add in a valuable way to current analysis and design practices. We discussed and exemplified some such ML procedures above, but consider in this and the next section broader areas, in which ML adds significant capabilities in reaching safer and more encompassing analyses and increasingly innovative designs. Using ML in design may not only help to process huge amounts of data to reach more useful analyses and better designs but even strengthen human intuition.

A very valuable area of ML and NN – and a quite modern development for general use – is to establish and use personalized digital twins. In this process a digital model (a twin) of an actual complex physical system is established. The model may indeed consist of a number of models in the *hierarchical modeling and analysis* process, see M.L. Bucalem and K.J. Bathe [C].

Considering the evolution of a digital twin, and assuming that we consider the design of a new civil structure, the digital twin would ideally be established right at the beginning of the design efforts and then contain models of the envisaged new structure. A simple beam model may be used to solve for some global overall response, and detailed finite element models may be employed for certain parts of the structure, like critical joints between columns and beams, elevator shafts, emergency exits. The purpose is to use the computer models *initially* in the design of the structure, with guidelines on design rules and material data given in widely used codes and databases. The numerically obtained analysis data for the structure, not yet built, may then be complemented with documented experiences

and available physically measured and numerical test results of similar structures, all available in possibly huge databases.

An important step in the design of the structure can be an optimization of the structure to safely withstand certain loads, to ensure that it operates with (almost) optimal effectiveness throughout its life cycle at a reasonable cost, that it can be easily maintained, and so on. While emphasis on optimization of structures has already been given in the early use of NN, see for example P. Hajela and L. Berke [A], based on recent developments, the optimization of structures can be naturally achieved using ML procedures since techniques of optimization are already available and used in ML (like to optimize the parameters and weights). ML assisted optimization is clearly of much interest in industry, to embed the step of optimization in the design of a structure and hence in the overall use of a digital twin, see T. Rabczuk and K.J. Bathe [A].

Hence it follows that to establish a valuable digital twin, much data is stored and processed which can be achieved with the use of NN and, more generally, ML procedures. For a detailed description of digital twins see T. Tripura, S. Garg and S Chakraborty [A], and for the possible future use of digital twins, see National Academies [A]. Also, for commercial offerings, see e.g. Bentley Systems, Inc. [B]

Once the structure has been built, the digital twin is also very valuable. The structure can be monitored, and physical tests can be performed throughout its life cycle until it is taken out of service, perhaps by demolition. The digital twin would be updated continuously and would thus, at any time, consist of the finite element model(s) used with the corresponding solution results and the data of continuous physical monitoring of the actual physical structure, using strain gages or other, to which specific physical test results may be added. In this way, the digital twin is then a *personalized twin of a specific structure*, and not a generic twin of a certain category of structures (which can, however, also be useful). Since, as time progresses, data pertaining to the personalized digital twin is continuously updated in that continuously new measured results are added and improved finite element models are introduced with new numerical results, we may look at the twin as dynamically evolving. For example, material damage and material fracture might be measured in certain structural components. These two phenomena need then to be represented in the finite element models to continuously have a computational model in the personalized twin that as closely as possible represents the actual physical structure at any point in time.

If the digital twin closely represents the actual physical structure, the engineers can *address questions by interrogating the digital twin* and obtain answers that can be very valuable. A typical such question might be "what if this catastrophic event occurred? ". Once the answers have been obtained, the structure can be sufficiently strengthened prior to the event occurring, like prior to a catastrophic earthquake of a large

magnitude.

In general, the ML techniques would be used to continuously search and compare data, assembled for the personalized digital twin and other similar structures, possibly as given in many databases, with the purpose to arrive at required remedies to apply as the structure ages.

The important point is that the *personalized* digital twin, if properly maintained and used, can be a very useful instrument to have a continuously updated image of the physical structure on the computer, which can be interrogated to keep the actual structure well maintained and safe for use throughout its lifetime. In this process, the use of ML with the needed procedures available is paramount in being able to deal with the massive amount of data used.

The above description considers a physical civil structure, like a building, bridge, or dam. However, personalized digital twins can be widely developed in many fields, like for space vehicles, aircrafts, city and architectural planning, endangered areas of the earth, and in the medical fields for patient care. In each of these applications a personalized digital twin can help to reach, depending on the application, a more effective structure, an improved environment, or a better medical care because the digital twin can be used to continuously monitor and provide insight to the actual physical condition, for “its” safety, strength, health or effectiveness, and make predictions considering future developments, thus enabling appropriate actions to be immediately taken.

With the significant value that a digital twin can offer, it is likely that there will be a rapid growth in their use, first for very important structures and systems, as we see such developments already and then followed by many more applications. The use of “modeling” on the computer in the broadest sense with ML procedures,

including the use of finite element methods, is a very important development. A large area of application is structural health monitoring, which we focus on briefly further below.

6.3.4 Structural health monitoring

Structural health monitoring (SHM) involves the long-term health monitoring of a structure to evaluate the structural integrity, durability and reliability at any time during the life of the structure. This process involves extensive measurements of the performance of the structure, the synthesis of data, comparisons and evaluations, probability aspects, and possible predictions using computational models, see K. Worden, C.R. Farrar, G. Manson, and G. Park [A], C. R. Farrar and K. Worden [A], M. Mehrjoo, N. Khaji, H. Moharrami, A. Bahreininejad [A], L.A. Bull, T.J. Rogers, C. Wickramarachchi, E.J. Cross, K. Worden and N. Dervilis [A], and Y. An,, E. Chatzi, S.H. Sim, S. Laflamme, B. Blachowski and J. Ou [A]. Hence structural health monitoring can involve the effective use of a personalized digital twin, incorporating a NN, which is used in the continuous monitoring of the structure.

Since the objective of SHM is to assess the health of the structure at any time during its lifetime, in essence to withstand events of structural loading, a sophisticated system to achieve the health monitoring may need to be designed to continuously evaluate newly collected data. Structural health monitoring is researched extensively and has already been used for many structures, in particular for major bridge and concrete structures, see J. Ko and Y.Q. Ni [A], W.B. Krätzig and Y.S. Petryna [A], and Y. Q. Ni, Y. Xia, W. Y. Liao and J. M. Ko [A], but at earlier times without the use of Machine Learning. Further applications are in the monitoring of historic buildings, dams, tunnels, tall buildings and cooling towers. A sudden – unforeseen – failure of such a structure could lead to many lives lost and large damage. Hence health monitoring can be very important

and the use of ML can be an effective ingredient, see also M. Domaneschi, A.Z..Noori, M.V. Pietropinto and G.P. Cimellaro [A], A.A. Torky and S. Ohno [A] and T. Rabczuk and K.J. Bathe [A].

There are important differences to the "ideal" use of a digital twin, though. One difference is that the structure might have been built without a digital twin, indeed without the use of computers. This surely is the case for very old, historic structures. Then, depending on the importance of the structure, a digital twin could still be developed, including a finite element model, but maybe costly to establish. The twin may also not be a very accurate representation of the actual physical structure, because the properties of the materials throughout the structure have changed and extensive measurements on the structure would have to be made.

As the name implies, in the process of health monitoring, primary emphasis is placed on assessing the "health" of the structure, that is, the structural integrity and possible failure, for example due to material weakening or cracking. The evaluation in comparison to other structures which are still in existence or have massively failed is then of paramount importance. Then too, extensive physical measurements must be conducted, and databases used for comparisons. In these efforts, ML procedures can be very useful – even without a digital twin -- because a huge amount of data needs to be processed and compared with.

An important application of SHM is the monitoring of wind turbines including their supports, see for example C.C. Ciang, J.R. Lee and H.J. Bang [A] and J. Baquerizo, C. Tutivén, B. Puruncajas, Y. Vidal and J. Sampietro [A]. Here measurements of the performance of the motor, the blades and the support structure can be continuously performed to focus on the safety of the structure, to have minimize damage, maintenance and repair costs. In particular, downtime of the structure should be minimized because it results in loss of energy. Hence the continuous monitoring of the structure, data acquisition and synthesis, based on numerical simulations and NN, and comparisons with data obtained for similar structures are important to ensure an

optimal performance of the system and whether possible failures are incipient. The use of a digital twin with finite element models can be particularly valuable.

Another important application is the health monitoring using ML of bridges -- like steel girder, cable-stayed suspension span, and reinforced concrete bridges, see for example H. Nick, A. Aziminejad, M.H. Hosseini and K Laknejadi [A] , F. Ni, J. Zhang and M.N. Noori [A], and O. Hasancebi and T. Dumlupinar [A]. Here too, the physical monitoring and use of a digital twin with finite element procedures can result in important repairs to be performed timely, and thus a greater safety of the structure.

The above applications show already that the health monitoring of structures with ML procedures will probably significantly increase over the years to come, also because there are the challenges due to new climate conditions affecting structures and surroundings as seen in various areas of the world.

6.4 Exercises

6.4.1

Obtain approximation functions of the following data $(x_i,\ y_i) = (10,5), (14,3),\ (20,12),\ (22,14) for\ i = 1,\ldots,4$. Use a linear regression and a parabolic regression to identify two functions.

6.4.2

Calculate the proper orthogonal decomposition of the simple covariance matrix

$$S = \begin{bmatrix} 6 & -4 & 0 \\ -4 & 7 & -3 \\ 0 & -3 & 3 \end{bmatrix}$$

6.4.3

Consider Fig. 6.1 and identify an additional basic task in ML that would be trained in supervised learning and also a task that would be trained in unsupervised learning. In each case mention some applications.

6.4.4

Consider the deep neural network in Fig. 6.3 and identify the number of variables to optimize for the NN when used with $n = 10$, $m = 6$, $p = 4$.

6.4.5

Show that in Eq. (6.15) the parameter values of the generalized Mooney-Rivlin model calculated from the weights are given correctly.

6.4.6

Consider a generic wind turbine blade, subjected to a given dynamic wind loading dependent on its geometry.

Describe how you would proceed using the process of Section 6.3.2 to optimize the geometry of the blade.

References

[A] D.W. Abueidda, S. Koric, N.A. Sobh and H. Sehitoglu. Deep learning for plasticity and thermo-viscoplasticity. Int. J. Plast. 136, 102852, 102852, 2021

[A] D. W. Abueidda, Z. Kang, S. Koric, K. A. James and I.M. Jasiuk. Topology optimization for three-dimensional elastoplastic architected materials using a path-dependent adjoint method. Int J Numer Methods Eng.;122:1889–1910, 2021.

[A] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold and E. Kuhl. Integrating machine learning and multiscale modeling – perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. Review article, NPJ digital medicine. Nature Publishing Group UK, Vol 2, Issue 1, 2019.

[A] D.J. Allman. A compatible triangular element including vertex rotations for plane elasticity analysis. Comput Struct 1984, 19(1-2), 1-8.

[B] D.J. Allman. A quadrilateral finite element including vertex rotations for plane elasticity analysis. Inter J Numer Meth Engrg 1988, 26(3), 717-730.

[C] D.J. Allman. Analysis of general shells by flat facet finite element approximation. Aeronautical Journal June/July 1991.

[A] Y. An,, E. Chatzi, S.-H. Sim, S. Laflamme,, B. Blachowski and J. Ou. Recent progress and future trends on damage identification methods for bridge structures. Struct Control Health Monit. Review. 2019. DOI: 10.1002/stc. 2416

[A] U. Andelfinger and E. Ramm. EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements. Int. Journal Num. Methods, Vol 36, 1311-1337, 1993

[A] M. Aristodemo. A high-continuity finite element model for two-dimensional elastic problems, Comput. Struct. 21 (5) (1985) 987–993.

[A] R.E. Bank, W.M. Coughran, Jr., W. Fichtner, E.H. Grosse, D.J. Rose and R.K. Smith. Transient simulation of silicon devices and circuits. IEEE Trans. CAD 4(4):436–451, 1985.

[A] J. Baquerizo, C. Tutivén , B. Puruncajas , Y. Vidal and J. Sampietro. Siamese Neural Networks for Damage Detection and Diagnosis of Jacket-Type Offshore Wind Turbine Platforms. Mathematics 10, 1131, 2022.

[A] K. J. Bathe, Finite Element Procedures, Prentice Hall, 1996, 2nd edition KJ Bathe, Watertown, MA, 2014; also published by Higher Education Press China 2016.

[B] K.J. Bathe. The inf–sup condition and its evaluation for mixed finite element methods. Comput Struct 2001;79(2):243–52.

[C] K.J. Bathe. The Finite Element Method with 'Overlapping Finite Elements'. Proceedings Sixth International Conference on Structural Engineering, Mechanics and Computation – SEMC 2016, Cape Town, South Africa (A. Zingoni, ed.), 2016.

[D] K.J. Bathe. The AMORE Paradigm for Finite Element Analysis. Advances in Engineering Software, 130, 1-13, 2019.

[E] K.J. Bathe. Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration Scheme. Computers & Structures, Vol. 85, no. 7–8, pp. 437–445, Apr. 2007.

[F] K.J. Bathe. Advances in the multiphysics analysis of structures. Computational Methods for Engineering Science, B.H.V. Topping (ed), Saxe-Coburg Publications, Stirlingshire, U.K., 2012

[G] K.J. Bathe. To Enrich Life, 2nd edition. 2019, Amazon.com

[H] K. J. Bathe. Convergence of Subspace Iteration. Formulations and Computational Algorithms in Finite Element Analysis, (K.J. Bathe, J.T. Oden and W. Wunderlich, eds.), M.I.T. Press, 1977

[I] K. J. Bathe. Simulation of Structural and Fluid Flow Response in Engineering Practice. Computer Modeling and Simulation in Engineering, Vol 1, 47-77, 1996.

[J] K.J. Bathe. Solution Methods for Large Generalized Eigenvalue Problems in Structural Engineering. Report UCSESM 71-20, Department of Civil Engineering, University of California, Berkeley, November 1971.

[K] K.J. Bathe. The Subspace Iteration Method – Revisited. Computers & Structures, 126, 177-183, 2013.

[L] K.J. Bathe. Remarks on the development of finite element methods and software. Int. J. Computer Applications in Technology, Vol. 7, Nos 3-6, 101-107, 1994.

[M] K.J. Bathe. The Finite Element Method. Encyclopedia of Computer Science and Engineering, B. Wah (ed.), J. Wiley and Sons, 1253-1264, 2009

[N] K.J. Bathe. Modern Finite Element Analysis with AMORE. Proceedings of the 14th International Conference on Computational Structures Technology and the 11th International Conference on Engineering Computational Technology (B.H.V. Topping et al, eds), 2022.

[O] K.J. Bathe. Challenges and Advances in the Analysis of Structures. Proceedings Fourth International Conference on Structural Engineering, Mechanics and Computation -- SEMC 2010, Cape Town, South Africa (A. Zingoni, ed.), 2010.

[P] K.J. Bathe. Video lectures prepared by MIT, https://www.youtube.com/watch?v=oNqSzzycRhw

[A] K.J. Bathe and C. Almeida. A Simple and Effective Pipe Elbow Element—Linear Analysis. ASME, J. Applied Mech., 47, 93-100, 1980.

[B] K.J. Bathe and C. Almeida. A Simple and Effective Pipe Elbow Element—Interaction Effects. J. Appl. Mech., 49, 165-171, 1982

[C] K.J. Bathe and C. Almeida. A Simple and Effective Pipe Elbow Element—Pressure Stiffening Effects. ASME J. Appl. Mech., 49, 914-916, 1982.

[A] K. J. Bathe and M. M. I. Baig. On a composite implicit time integration procedure for nonlinear dynamics. Computers & Structures, Vol. 83, no. 31–32, pp. 2513–2524, Dec. 2005.

[A] K. J. Bathe and F. Brezzi. A Simplified Analysis of Two Plate Bending Elements – the MITC4 and MITC9 Elements. Proceedings, Numerical Methods in Engineering: Theory and Applications, Univ. College Swansea, Wales, July 1987.

[A] K.J. Bathe, F. Brezzi F and L.D. Marini. The MTC9 shell element in plate bending: mathematical analysis of a simplified case. Comput Mech 2011; 47: 617-626.

[A] K.J. Bathe, M.L. Bucalem and F. Brezzi, "Displacement and Stress Convergence of the MITC Plate Bending Elements", J. Eng. Computations, 7, no. 4, 291-302, 1990.

[A] K.J. Bathe, D. Chapelle and P.S. Lee. A shell problem 'highly sensitive' to thickness changes. Int J Numer Methods Eng 2003;57:1039-52.

[A] K.J. Bathe and A. Chaudhary. On the Displacement Formulation of Torsion of Shafts with Rectangular Cross-sections. International Journal for Numerical Methods in Engineering, 18, 1565-1568, 1982.

[A] K.J. Bathe and J. Dong. Component Mode Synthesis with Subspace Iterations for Controlled Accuracy of Frequency and Mode Shape Solutions. Computers & Structures, 139, 28–32, 2014.

[A] K.J. Bathe and E.N. Dvorkin. A formulation of general shell elements - the use of mixed formulation of tensorial components. *Int J Numer Meth Eng* 1986;22(3):697-722.

[B] K.J. Bathe and E.N. Dvorkin, A Four-Node Plate Bending Element Based on Mindlin/Reissner Plate Theory and a Mixed Interpolation. International Journal for Numerical Methods in Engineering, 21, 367-383, 1985.

[A] K. J. Bathe and L. W. Ho. A Simple and Effective Element for Analysis of General Shell Structures. J. Computers & Structures, 13, 673-681, 1981.

[A] K.J. Bathe, A. Iosilevich and D. Chapelle. An inf-sup test for shell finite elements. Comput Struct 2000; 75(5):439-56.

[B] K.J. Bathe, A. Iosilevich and D. Chapelle, "An Evaluation of the MITC Shell Elements", Computers & Structures, 75, 1-30, 2000.

[A] K.J. Bathe, P.S. Lee and J.F. Hiller. Towards improving the MITC9 shell element. Comput Struct 2003;81(8):477-89.

[A] K. J. Bathe and G. Noh. Insight into an implicit time integration scheme for structural dynamics. Computers & Structures, vol. 98–99, pp. 1–6, May 2012.

[A] K.J. Bathe and S. Ramaswamy. On Three-Dimensional Nonlinear Analysis of Concrete Structures. J. Nuclear Engineering and Design, 52, 385-409, 1979.

[A] K. J. Bathe and E. L. Wilson. NONSAP - A Nonlinear Structural Analysis Program. J. Nuclear Engineering and Design, 29, 266-293, 1974.

[B] K. J. Bathe and E. L. Wilson. Stability and accuracy analysis of direct integration methods. International Journal of Earthquake Engineering and Structural Dynamics, 1, 283-291, 1973.

[C] K.J. Bathe and E.L. Wilson. Numerical Methods in Finite Element Analysis. Prentice-Hall 1976.

[A] K. J. Bathe, E. L. Wilson, and F. E. Peterson. SAP IV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems. Earthquake Engineering Research Center Report No. 73-11, College of Engineering, University of California, Berkeley, June 1973, revised April 1974

[A] K.J. Bathe and H. Zhang. A Mesh Adaptivity Procedure for CFD & Fluid-Structure Interactions. Computers & Structures, 87, 604-617, 2009.

[A] K.J. Bathe and L. Zhang, The Finite Element Method with Overlapping Elements –.A New Paradigm for CAD Driven Simulations, Computers & Structures, 182, 526-539, 2017.

[A] J.L. Batoz, K.J. Bathe and L.W. Ho. A study of three-node triangular plate bending elements. *Int J Numer Meth Eng* 1980;15(12):1771-812..

[A] T. Belytschko and C.S.Tsai. A stabilization procedure for the quadrilateral plate element with one-point quadrature. Int J Numer Meth Eng 1983;19(3):405-19.

[A] T. Belytschko, Y.Y. Lu and L. Gu. Element-free Galerkin methods. Int. J. Num Meth. In Eng. 37,

229-256, 1994.

[A] J.A. Benek, P.G. Buning and J.L. Steger. A 3-D Chimera grid embedding technique. 7th Computational Physics Conference, Cincinnati, USA, 1985

[A] J.M. Benítez, F.J. Montáns. The value of numerical amplification matrices in time integration methods. Computers and Structures, 128, 243–250, 2013.

[A] Bentley Systems, Inc. , the ADINA program.

[B] Bentley Systems, Inc. itwin Capture, https://www.bentley.com/software/itwin-capture/

[A] M. Bischoff, E. Ramm and D. Braess, A class of equivalent enhanced assumed strain and hybrid stress finite elements. Computational Mechanics, 22, 443-449, 1999.

[A] K.U. Bletzinger, M. Bischoff and E. Ramm. A unified approach for shear-locking-free triangular and rectangular shell finite elements. Comput Struct 2000;75:321-34.

[A] C. Bonatti, and D. Mohr. On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids. J. Mech. Phys. Solids 158, 104697, 2022.

[A] C. Bonatti, B. Berisha and D. Mohr. From CP-FFT to CP-RNN: Recurrent neural network surrogate model of crystal plasticity. International Journal of Plasticity 158, 103430, 2022.

[A] R. Boussetta, T. Coupez and L. Fourment. Adaptive remeshing based on a posteriori error estimation for forging simulation. Comp. Meth. In Appl. Mech. And Eng., 195 (48-49) , 6626-6645, 2006

[A] M. C. Boyce and E. M. Arruda. Constitutive models of rubber elasticity- A review. Rubber Chemistry and Technology, 73(3), 504-523, 2000.

[A] F. Brezzi and K. J. Bathe. A Discourse on the Stability Conditions for Mixed Finite Element Formulations. J. Computer Methods in Applied Mechanics and Engineering, 82, 27-57, 1990.

[A] F. Brezzi, K. J. Bathe and M. Fortin. Mixed-Interpolated Elements for Reissner/Mindlin Plates. Int. J. Num. Methods in Engineering, 28, 1787-1801, 1989.

[A] M. L. Bucalem and K. J. Bathe. The Mechanics of Solids and Structures – Hierarchical Modeling and the Finite Element Solution. Springer-Verlag, 2011.

[B] M.L. Bucalem and K.J. Bathe. Higher-order MITC general shell elements. Int J Numer Meth Eng 1993;36(21):3729–54.

[C] M.L. Bucalem and K.J. Bathe. Finite element analysis of shell structures. Archiv Comput Meth Engrg 1997;4(1):3-61.

[A] L.A. Bull, T.J. Rogers, C. Wickramarachchi, E.J. Cross, K. Worden, and N. Dervilis. Probabilistic active learning: An online framework for structural health monitoring. Mechanical Systems and Signal Processing 134, 106294, 2019.

[A] J. M. A. César de Sá, R.M. Natal Jorge, R. A. Fontes Valente and P.M. Almeida Areias, Development of shear locking-free shell elements using an enhanced assumed strain formulation, Int. J. Num. Meth. Engng 2002; 53:1721–1750

[A] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. J Comput Phys 1990;90(1):1-64.

[A] A. Chakraborty, T. Wicks, T. Rabczuk, and X Zhuang. Multigoal-oriented dual-weighted residual error estimation using PINNs. Machine Learning for Computational Science and Engineering, Vol. 1, 2025.

[A] Y. Chai and K.J. Bathe. Transient Wave Propagation in Inhomogeneous Media with Enriched Overlapping Triangular Elements. Computers & Structures, 237, 2020.

[A] W. Cao, J. Song and W. Zhang. A solver for subsonic flow around airfoils based on physics-informed neural networks and mesh transformation. Phys. Fluids 36, 027134, 2024.

[A] D. Chapelle and K.J. Bathe. Fundamental considerations for the finite element analysis of shell structures. Comput Struct 1998;66(1):19-36,711-2.

[B] D. Chapelle and K.J. Bathe. The finite element analysis of shells - fundamentals. Berlin: Springer; 2003; 2nd edition, 2011.

[C] D. Chapelle and K.J. Bathe. The mathematical shell model underlying general shell elements. Int J Numer Methods Eng 2000;48:289-313.

[D] D. Chapelle and K. J. Bathe. The Inf-Sup Test. J. Computers & Structures, 47, No. 4/5, 537-545, 1993.

[E] D. Chapelle and K.J. Bathe. On the ellipticity condition for model-parameter dependent mixed formulations. J. Computers & Structures, 88, 581–587, 2010.

[A] L.W. Chen and N. Thuerey. Towards high-accuracy deep learning inference of compressible turbulent flows over aerofoils. Computers & Fluids, Vol. 250, 105707, 2023.

[A] H.G Choi and P.S. Lee. The simplified MITC4+ shell element and its performance in linear and nonlinear analysis. Computers and Structures 290, 107177, 2024

[A] J. Chung, G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-alpha method. Journal of Applied Mechanics (ASME), 60, 371–375, 1993

[A] C.C. Ciang, J.R. Lee and H.J. Bang. Structural health monitoring for a wind turbine system: a review of damage detection methods. Meas. Sci. Technol. 19, 122001, 2008

[A] L. Collatz, The Numerical Treatment of Differential Equations, 3rd edn. Springer-Verlag, New York, 1966

[A] R.D. Cook. On the Allman triangle and a related quadrilateral element. Comput Struct 1986;22(6):1065–7.

[A] S.B. Cooper and J. Van Leeuwen. Alan Turing: his work and impact, Elsevier 2013.

[A] G.R. Cowper. Gaussian quadrature formulas for triangles. Int J Numer Methods Eng 1973;7:405–8.

[A] Y.J. Dai, Y.R. An, Z. Li, J. H. Zhang and C. Yu. Fourier neural operator with boundary conditions for efficient prediction of steady airfoil flows. Applied Mathematics and Mechanics, 44(11), 2019 – 2038, 2023.

[A] W.J.T. Daniel. Performance of overlapping finite elements. Comput. Struct. 31 (1) (1989) 47–53.

[A] S. De and K.J. Bathe. The Method of Finite Spheres. Computational Mechanics, 25, 329-345, 2000.

[B] S. De and K.J. Bathe. Displacement/Pressure Mixed Interpolation in the Method of Finite Spheres. International Journal for Numerical Methods in Engineering, 51, 275-292, 2001.

[A] M. Domaneschi, A. Z. Noori, M.V. Pietropinto and G.P. Cimellaro. Seismic vulnerability assessment of existing school buildings. Comput Struct 248:106522, 2021

[A] C.A. Duarte, I. Babuška and J.T. Oden. Generalized Finite Element Methods for Three-dimensional Structural Mechanics Problems. Computers & Structures 77, 215-232, 2000.

[A] E.N. Dvorkin and K.J. Bathe. A continuum mechanics based four-node shell element for general nonlinear analysis. Eng Comput 1984;1(1):77-88.

[A] E.N. Dvorkin and S.. I. Vassolo. A quadrilateral 2-D finite element based on mixed interpolation of tensorial components. Eng. Comput., 1989, Vol. 6, 1989.

[A] A. P. Engelbrecht. Computational Intelligence: An Introduction. Second Edition, J. Wiley & Sons, 2007.

[A] C. R. Farrar and K. Worden. An introduction to structural health monitoring. Phil. Trans. R. Soc. A 2007 365, 2007.

[A] A.L. Frankel, R.E. Jones, C. Alleman and J.A. Templeton. Predicting the mechanical response of oligocrystals with deep learning. Comput. Mater. Sci. 169, 109099, 2019.

[A] T.P. Fries and T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications. Vol. 84, Issue3, pp. 253-304, 2010.

[A] J.N. Fuhg, C.M. Hamel, K. Johnson, R. Jones and N Bouklas. Modular machine learning-based elastoplasticity: Generalization in the context of limited data. Comp. Methods Applied Mech Engrg, 407, 115930, 2023.

[A[C.L. Gardner, J.W. Jerome and R.S. Eisenberg. Electro-diffusion model simulation of rectangular current pulses in a voltage-biased biological channel. J Theor Biol, 219:291–299, 2002.

[A] F. Ghavamian and A. Simone. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. Comput. Methods Appl. Mech. Eng. 357, 112594, 2019

[A] S. Goswami, C. Anitescu, S. Chakraborty and T. Rabczuk. Transfer learning enhanced-physics informed neural network for phase field modeling of fracture. Theoretical and Applied Fracture Mechanics, 106, 102447, 2020.

[A] T. Grätsch and K.J. Bathe. A posteriori Error Estimation Techniques in Practical Finite Element Analysis. Computers & Structures, 83, 235-265, 2005

[B] T. Grätsch and K.J. Bathe. Influence Functions and Goal-Oriented Error Estimation for Finite Element Analysis of Shell Structures. International Journal for Numerical Methods in Engineering, 63, 709-736, 2005

[A] H. Guo, X. Zhuang, P. Chen, N. Alajlan, and T. Rabczuk. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Engineering with Computers, 38: 5173-5198, 2022.

[A] P. Hajela and L. Berke. Neurobiological computational models in structural analysis and design. Computers & Structures 41, 4, 657-667, 1991.

[A] S. Ham and K. J. Bathe. A finite element method enriched for wave propagation problems. Computers & Structures, vol. 94–95, pp. 1–12, 2012.

[B] S. Ham, B. Lai and K.J. Bathe. The Method of Finite Spheres for Wave Propagation Problems. Computers & Structures, 142, 1-14, 2014.

[A] O. Hasancebi and T. Dumlupinar. Linear and nonlinear model updating of reinforced concrete T-beam bridges using artificial neural networks. Comp. Struct. 119, 1-11, 2013.

[A] J.N. Heidenreich, C. Bonatti and D. Mohr. Transfer learning of recurrent neural network-based plasticity models. *Int J Numer Methods Eng*.;125:e7357, 2024.

[A] J.F. Hiller and K.J. Bathe. Measuring convergence of mixed finite element discretizations: an application to shell structures. Comput Struct 2003;81(8):639-54.

[A] J.W. Hong and K.J. Bathe. Coupling and Enrichment Schemes for Finite Element and Finite Sphere Discretizations. Computers & Structures, 83, 1386-1395, 2005

[A] J. Huang and K.J. Bathe. Quadrilateral overlapping elements and their use in the AMORE paradigm. Computers & Structures, 222, 25-35, 2019

[B] J. Huang and K.J. Bathe. Overlapping Finite Element Meshes in AMORE. Advances in Engineering Software, 144, 2020.

[C] J. Huang and K.J. Bathe. On the Convergence of Overlapping Elements and Overlapping Meshes. Computers & Structures, 244, 2021.

[A] A. Ibrahimbegovic and E.L. Wilson. A unified formulation for triangular and quadrilateral flat shell elements with six nodal degrees of freedom. Comm. in Applied Numer. Methods, 7, 1-9, 1991.

[A] H. Jun, K. Yoon, P.S. Lee, and K.J. Bathe. The MITC3+ Shell Element Enriched in Membrane Displacements by Interpolation Covers. Comput. Methods Appl. Mech. Engrg., 337, 458-480, 2018.

[A] J. Jung, K. Yoon and P.S. Lee. Deep learned finite elements, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113401

[A] G.E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang and L. Yang. Physics-informed machine learning. Nature Reviews Physics, 3, 422- 440. Springer Nature, 2021.

[A] Z. Kazancı and K.J. Bathe. Crushing and crashing of tubes with implicit time integration. Int. J. of Impact Engineering, 42, 80-88, 2012

[A] P. Keast. Moderate-degree tetrahedral quadrature formulas. Comput Methods Appl Mech Eng 1986;55:339–48.

[A] D.N. Kim and K.J. Bathe. A triangular six-node shell element. Comput Struct 2009;87(23), 1451-460.

[A] J. Kim and K.J. Bathe. The Finite Element Method Enriched by Interpolation Covers. Computers & Structures, 116, 35-49, 2013.

[B] J. Kim and K.J. Bathe. Towards a Procedure to Automatically Improve Finite Element Solutions by Interpolation Covers. Computers & Structures, 131, 81-97, 2014.

[A] K.T. Kim and K.J. Bathe. Transient Implicit Wave Propagation Dynamics with the Method of Finite Spheres. Computers & Structures, 173, 50–60, 2016.

[B] K.T. Kim and K.J. Bathe. Accurate Solution of Wave Propagation Problems in Elasticity. Computers & Structures, 249, 2021.

[C] K.T. Kim and K.J. Bathe. The Bathe Subspace Iteration Method Enriched by Turning Vectors. Computers & Structures, 186, 11–21, 2017.

[A] K.T. Kim, L. Zhang and K.J. Bathe. Transient implicit wave propagation dynamics with overlapping finite elements. Comput Struct 2018;199:18-33.

[A] W. Kim and S. Y. Choi. An improved implicit time integration algorithm: The generalized composite time integration algorithm. Computers & Structures, 196, 341-354, 2018.

[A] W. Kim and J. H. Lee. An improved explicit time integration method for linear and nonlinear structural dynamics. Computers and Structures, 206, 42-53, 2018.

[A] J. Ko, and Y.Q. Ni. Technology developments in structural health monitoring of large-scale bridges. Eng Struct. 27, 1715-1725, 2005.

[A] Y. Ko and K.J. Bathe. A new 8-node element for analysis of three-dimensional solids. Computers & Structures, 202, 85-104, 2018

[B] Y. Ko and K.J. Bathe. Inf-sup testing of some three-dimensional low-order finite elements for the analysis of solids. Computers & Structures, 209, 1-13, 2018

[A] Y. Ko, P.S. Lee, and K.J. Bathe. A New MITC4+ Shell Element. Computers & Structures, 182, 404- 418, 2017.

[B] Y. Ko, P.S. Lee and K.J. Bathe. A new 4-node MITC element for analysis of two-dimensional solids and its formulation in a shell element. Computers & Structures, 192, 34-49, 2017

[A] Y. Ko, X. Zhang and K.J. Bathe. Continuum mechanics-based shell elements with six degrees of freedom at each node -- the MITC4 / D and MITC4+ / D elements. Computers & Structures, Vol 308, February 2025, 107622.

[A] M.J. Kochenderfer and T. A. Wheeler, Algorithms for Optimization, MIT Press, 2019.

[A] M Kojic and KJ Bathe. Inelastic Analysis of Solids and Structures. Springer, 2005.

[A] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart and A. Anandkumar. Neural Operator: Learning maps between function spaces with applications to PDEs. J. of Machine Learning Research 24, 1-97, 2023.

[A] W. B. Krätzig and Y.S. Petryna. Structural damage and life-time estimates by nonlinear FE simulation. Engineering Structures 27 (2005) 1726–1740

[A] R. Kroyer, K. Nilsson and K.J. Bathe. Advances in direct time integration schemes for dynamic analysis. Automotive CAE Companion 2016/2017, 32-35, 2016.

[A] B. M. Kulfan, Recent extensions and applications of the 'CST' universal parametric geometry representation method. The Aeronautical journal. Vol.114 (1153), p.157-176, 2010.

[A] G.M. Kulikov and S.V. Plotnikova. A family of ANS four-node exact geometry shell elements in general convected curvilinear coordinates. Int J Numer Meth Eng 2010;83(10):1376-406.

[A] S.B. Kwon, K. J. Bathe and G. Noh. An analysis of implicit time integration schemes for wave propagations. Computers & Structures, vol. 230, p. 106188, 2020.

[B] S.B. Kwon, K.J. Bathe and G. Noh. Selecting the Load at the Intermediate Time Point of the ρ_∞ - Bathe Time Integration Scheme. Computers & Structures 254, 2021.

[A] B. Lai and K.J. Bathe. The Method of Finite Spheres in Three-Dimensional Linear Static Analysis. Computers & Structures, 173, 161–173, 2016.

[A] H. Lamb. On the Propagation of Tremors over the Surface of an Elastic Solid. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, vol. 203, pp. 1–42, 1904,

[A] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. of research of the national bureau of standards, 1950; 45:255-282.

[A] M. Latorre and F.J. Montáns. Extension of the Sussman-Bathe spline-based hyperelastic model to incompressible transversely isotropic materials. Comput and Structures, 122:13-26, 2013.

[B] M. Latorre and F.J. Montáns, WYPiWYG hyperelasticity without inversion formula: Application to passive ventricular myocardium. Comput and Structures 185:47-58, 2017.

[C] M. Latorre and F.J. Montáns. Experimental data reduction for hyperelasticity. Computers& Structures. 232:1-16, 2020.

[A] M. Latorre, E. Peña and F. J. Montáns, Determination and finite element validationof the WYPIWYG strain energy of superficial fascia from experimental data. Annals of Biomedical Engineering 45 (3) (2017) 799–810

[A] C. Lee, K.J. Bathe and G. Noh. Stability of the Bathe implicit time integration methods in the presence of physical damping. Comput. Structures, 295, 107294, 2024.

[A] C. Lee and P.S. Lee. A new strain smoothing method for triangular and tetrahedral finite elements. Comput Methods Appl Mech Eng 2018;341:939–55.

[B] C. Lee and P. S. Lee, The strain-smoothed MITC3+ shell finite element. Computers and Structures 223 106096, 2019.

[A] N. S. Lee and K. J. Bathe. Effects of Element Distortions on the Performance of Isoparametric Elements. ,Int. J. for Numerical Methods in Engineering, 36, 3553-3576, 1993.

[A] P.S. Lee and K.J. Bathe. On the asymptotic behavior of shell structures and the evaluation in finite element solutions. Comput Struct 2002;80(3):235-55.

[B] P.S. Lee and K.J. Bathe. Insight into finite element shell discretizations by use of the "basic shell mathematical model". Comput Struct 2005;83(1):69-90.

[C] P.S. Lee and K.J. Bathe. The quadratic MITC plate and MITC shell elements in plate bending. *Adv Eng Software* 2010;41(5):712–28.

[D] P.S. Lee and K.J. Bathe. Development of MITC isotropic triangular shell finite elements. Comput Struct 2004;82(11):945-62.

[A] S. Lee and K.J. Bathe. An Enhancement of Overlapping Finite Elements. Computers & Structures, 260, 2022.

[B] S. Lee and K.J. Bathe. Additional Overlapping Finite elements – the Pyramid and Prism Elements. Computers & Structures, 268, 106813, 2022.

[C] S. Lee and K.J. Bathe. Solution of the Generalized Eigenvalue Problem using Overlapping Finite Elements. Advances in Engineering Software, 173, 103241, 2022.

[A] Sh. Lee and KJ Bathe. A constitutive neural network for hyperelastic materials. Machine Learning for Computational Science and Engineering, Vol. 1, 2025.

[A] Y. Lee. P.S. Lee and K.J. Bathe. The MITC3+ shell element and its performance. Comput Struct 2014;138:12-23.

[A] Y. Lee, H.M. Jeon, P.S. Lee and K.J. Bathe. The modal behavior of the MITC3+ triangular shell element. Comput Struct 2015;153:148-64.

[A] J.Z. Li and K.P. Yu. An alternative to the Bathe algorithm. Applied Mathematical Modelling, Vol. 69, pp. 255 -272, 2019.

[A] Z. Li, N. Kovachki, C. Choy, B. Li, J. Kossaifi, S. P. Otta, M. A Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli and A. Anandkumar. Geometry-Informed Neural Operator for Large-Scale 3D PDEs, arXiv:2309.00583v, 2023.

[A] K. Linka and E. Kuhl. A new family of Constitutive Artificial Neural Networks towards automated model discovery. Comp Meth in Appl. Mech. and Eng. 403, 115731, 2023.

[A] G.R. Liu, T. Nguyen-Thoi and K.Y. Lam. An edge-based smoothed finite element method (ES-FEM) for static, free and forced vibration analyses of solids, Journal of Sound and Vibration 320, 1100–1130, 2009

[A] R.M MacNeal and R.L. Harder. A proposed standard set of problems to test finite element accuracy. Finite Elem Anal Des 1985;1(1):3-20.

[A] M. M. Malakiyeh, S. Shojaee and K. J. Bathe. The Bathe time integration method revisited for prescribing desired numerical dissipation. Computers & Structures, vol. 212, pp. 289–298, 2019,

[A] M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J.Bathe. The explicit β_1/ β_2-Bathe time integration method. Computers & Structures, 286 (2023) 107092

[B] M.M. Malakiyeh, S. Shojaee, S. Hamzehei-Javaran and K.J. Bathe. New Insights into the beta-1/2 Bathe time integration scheme when L-stable. Computers & Structures, 245, 106433, 2021. .

[A] M. M. Malakiyeh, Z. Anjomshoae, S. Shojaee, S.Hamzehei-Javaran, and K.J. Bathe. Towards optimal use of the explicit β_1/β_2-Bathe time integration method for linear and nonlinear dynamics. Computers & Structures, 298, 107350, 2024.

[A] W. S. McCulloch and W Pitts. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5 (4), 115-133, 1943.

[A] M. Mehrjoo, N. Khaji, H. Moharrami and A. Bahreininejad. Damage detection of truss bridge joints using Artificial Neural Networks. Expert Systems with Applications 35, 1122–1131, 2008.

[A] J. Miklowitz. The theory of elastic waves and waveguides. New York, NY: North Holland, 1978.

[A] M Miñano and F.J. Montáns. WYPiWYG damage mechanics for soft materials: A data-driven approach. Arch Comp Meth in Eng, 25, 165-193, 2018.

[A] F.J. Montáns, E. Cueto and K.J. Bathe. Machine Learning in Computer Aided Engineering.

Machine Learning in Modeling and Simulation – Methods and Applications, by T. Rabczuk and K.J. Bathe (eds), Springer -Verlag, 2023.

[A] J. Moor. The Dartmouth College Artificial Intelligence Conference: The Next Fifty years, AI Magazine, vol 27, No. 4, pp. 87–89, 2006.

[A] NASA TM 2637. National Aeronautics and Space Administration, USA, Technical Memorandum 2637, 1972.

[A] M.Z. Naser. A Look into How Machine Learning is Reshaping Engineering Models: the Rise of Analysis Paralysis, Optimal yet Infeasible Solutions, and the Inevitable Rashomon Paradox. Machine Learning for Computational Science and Engineering, Vol.1, 2025.

[A] National Academies. Opportunities and Challenges for Digital Twins in Engineering: Proceedings of a Workshop—in Brief. (2023)

[A] V.M. Nguyen-Thanh, C. Anitescu, N. Alajlan, T. Rabczuk and X. Zhuang. Parametric deep energy approach for elasticity accounting for strain-gradient effects. Computer Methods in Applied Mechanics and Engineering, 386, 114096, 2021.

[A] F. Ni, J. Zhang and M.N. Noori. Deep learning for data anomality detection and data compression of a long-span suspension bridge. Computer-Aided Civ Infrastructure Eng., 35, 685-700, 2020

[A] Y. Q. Ni, Y. Xia, W. Y. Liao and J. M. Ko. Technology innovation in developing the structural health monitoring system for Guangzhou New TV Tower. Struct. Control Health Monit., 16:73–9, 2009.

[A] H. Nick, A. Aziminejad, M.H. Hosseini and K. Laknejadi. Damage identification in steel girder bridges using model strain energy based damage index method and artificial neural network. Eng Fail Anal. 119: 105010, 2021.

[A] W. L. Nicomedes, K. J. Bathe, F. J. S. Moreira and R C.Mesquita. Meshfree analysis of electromagnetic wave scattering from conducting targets: Formulation and computations. Computers & Structures, Vol. 184, pp. 36–52, 2017.

[B] W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R. C.Mesquita. The Method of Finite Spheres in Acoustic Wave Propagation through Nonhomogeneous Media: Inf-Sup Stability Conditions. Vietnam Journal of Mechanics, VAST, 42, No. 3, 209 – 237, 2020.

[C] W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R C.Mesquita. Acoustic Scattering in Nonhomogeneous Media and the Problem of Discontinuous Gradients: Analysis and inf-sup Stability in the Method of Finite Spheres. Int. J. Numerical Methods in Eng. 122:3141–3170, 2021.

[D] W.L. Nicomedes, K.J. Bathe, F. J. S. Moreira and R C.Mesquita. Overlapping Finite Elements for the Navier-Stokes Equations. Computers & Structures, Computers & Structures, 299, 107343, 2024.

[A] J. Nocedal and S. J. Wright. Numerical Optimization, 2nd edn., Springer Series in Operations Research, Springer, 2006.

[A] G. Noh and K. J. Bathe. An explicit time integration scheme for the analysis of wave propagations. Computers & Structures, Vol. 129, pp. 178–193, 2013.

[B] G. Noh and K.J. Bathe. Imposing Displacements in Implicit Direct Time Integration & a Patch Test. Advances in Engineering Software, 175, 103286, 2023

[C] G. Noh and K.J. Bathe. Further insights into an implicit time integration scheme for structural dynamics. Computers & Structures, 202, 15–24, 2018.

[D] G. Noh and K. J. Bathe. The Bathe time integration method with controllable spectral radius: The ρ_∞-Bathe method. Computers & Structures, vol. 212, pp. 299–310, 2019.

[E] G. Noh and K.J. Bathe. For Direct Time Integrations: A Comparison of the Newmark and ρ_∞-Bathe schemes. Computers & Structures, 225, 2019.

[A] G. Noh, S. Ham and K. J. Bathe. Performance of an implicit time integration scheme in the analysis of wave propagations. Computers & Structures, Vol. 123, pp. 93–105, Jul. 2013

[A] J. A. A. Opschoor, P. C. Petersen and C. Schwab. Deep ReLU Networks and High-Order Finite Element Methods. Report ETH Zürich, Dec. 2019.

[A] P.D. Panagiotopoulos and Z. Waszczyszyn. The Neural Network approach in plasticity and fracture mechanics. Neural Networks in the analysis and design of structures. Springer, Berlin, pp161-195. 1999.

[A] D. J. Payen and K.J. A stress improvement procedure. Comput Struct 2012;112-113:311-26.

[A] T. Rabczuk and KJ Bathe, (eds.) Machine Learning in Modeling and Simulation. Springer 2023.

[A] R Radovitzky and E.N. Dvorkin. A 3D element for non-linear analysis of solids. Communications in numerical methods in Engineering, 10, 183-194, 1994.

[A] M. Raissi and G.E. Karniadakis. Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. 357, 125–141, 2018.

[A] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations, arXiv:1711.10561, 2017.

[B]]M. Raissi, P. Perdikaris and G.E. Karniadakis. Physics informed deep learning (part II): data-driven discovery of nonlinear partial differential equations, arXiv:1711.10566, 2017

[C] M. Raissi, P. Perdikaris and G.E. Karniadakis, Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. J. Comput. Phys., 378, .686-707, 2019.

[A] H.Y. Roh and M. Cho. The application of geometrically exact shell elements to B-spline surfaces, Comput. Methods Appl. Mech. Engrg. 193, 2261–2299, 2004.

[A] X. Romero, M. Latorre and F. J. Montáns. Determination of the WYPiWYG strain energy density of skin through finite element analysis of the experiments oncircular specimens, Finite Elements in Analysis and Design 134 (2017) 1–15.

[A] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6), 386-408, 1958.

[A] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang and T. Rabczuk An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput. Methods Appl. Mech. Engrg. 362, 112790, 2020.

[A] O. Sedehi, A.M. Kosikova, C. Papadimitriou, and L.S. Katafygiotis. On the integration of Physics-Based Machine Learning with hierarchical Bayesian modeling techniques. Mechanical Systems and Signal Processing, 208, 111021, 2024.

[A] Y. Shi, P. Wei, K. Feng , D.C. Feng and M. Beer, A Survey on Machine Learning Approaches for Uncertainty Quantification. Machine Learning for Computational Science and Engineering, Vol. 1, 2025

.[A] J.C. Simo and M.S. Rifai. A class of mixed assumed strain methods and the method of incompatible modes. *Inter J Numer Meth Eng* 1990;29(8):1595-638.

[A] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics 375 (2018) 1339–1364

[A] D. Soares. A novel family of explicit time marching techniques for structural dynamics and wave propagation models. Computer Methods in Applied Mechanics and Engineering, 311, 838-855, 2016.

[A] S. Sra, S. Nowozin and S. J. Wright(eds). Optimization for Machine Learning, MIT Press 2011.

[A] J.L. Steger and J.A. Benek. On the use of composite grid schemes in computational aerodynamics. Comput Methods Appl Mech Eng 1987;64(1-3):301-20.

[A] T. Sussman and K.J. Bathe. Spurious modes in geometrically nonlinear small displacement finite elements with incompatible modes. Comput Struct 2014;140:14-22.

[B] T. Sussman and K.J. Bathe. 3D-shell elements for structures in large strains. *Comput Struct* 2013;122:2-12.

[C] T. Sussman and K. J. Bathe. A Model of Incompressible Isotropic Hyperelastic Material Behavior using Spline Interpolations of Tension-Compression Test Data. Communications in Numerical Methods in Engineering, 25, 53-63, 2009.

[A] P.S. Theocaris and P.D. Panagiotopoulos. Neural networks for computing in fracture mechanics. Methods and prospects of applications, Comp Meth in Appl. Mech and Eng, 106, 213-228, 1993.

[A] R. Tian, G. Yagawa and H. Terasaka. Linear Dependence Problems of Partition of Unity-Based Generalized FEMs. Computer Methods in Applied Mechanics and Engineering, 195, 4768-4782, 2006.

[A] A.A. Torky and S. Ohno. Deep learning techniques for predicting nonlinear multi-component seismic responses of structural buildings. Comput & Struct. 252:106570, 2021.

[A] J. D. Toscano, V. Oommen A J Varghese, Z Zou, N .A. Daryakenari, C. Wu, and G. E . Karniadakis. From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning. Machine Learning for Computaional Science and Engineering, Vol.1, 2025

[A] T. Tripura, S Garg and S Chakraborty. Digital twin for dynamic systems. Machine Learning in Modeling and Simulation, T Rabczuk and KJ Bathe (eds). Springer 2023.

[A] A. M. Turing, Computing Machinery and Intelligence. Mind 49: 433-460, 1950.

[A] W.A. Wall, M. Bischoff and E. Ramm. A deformation dependent stabilization technique, exemplified by EAS elements at large strains. Comp Meth Appl Mech Eng 2000;188(4):859-71

[A] Z. Waszczyszyn and L Ziemianski, Neural Networks in mechanics of structures and materials – new results and prospects of applications. Comput & Struct. 79, 2261-2276, 2001.

[A] P. Wei and M. Beer. Regression models for machine learning. Machine Learning in Modeling and Simulation (T. Rabczuk and K.J. Bathe, eds). Springer 2023.

[A] K.R. Weiss and T M. Khoshgoftaar, Evaluation of Transfer Learning Algorithms using Different Base Learners. IEEE Proceedings - 29th International Conference on Tools with Artificial Intelligence, 2018-07-06.

[A] W. Wen, Y. Tao, S. Duan, J. Yan, K. Wei and D. Fang. A comparative study of three composite implicit schemes on structural dynamic and wave propagation analysis. Computers & Structures, 190, 126-149, 2017

[A] B. Widrow and M.E Hoff. Associative storage and retrieval of digital information in networks of adaptive "neurons". In Biological prototypes and synthetic systems. Springer, Berlin, 1962.

[A] E.L. Wilson and A. Ibrahimbegovic. Use of incompatible displacement modes for the calculation of element stiffness or stresses. Finite Elem Anal Des 1990;7:229-41.

[A] K. Wisniewski and E. Turska. Enhanced Allman quadrilateral for finite drilling rotations.

Comput Meth Appl Mech Engrg 2006;195(44–47):6086–109.

[A] K. Worden, C.R. Farrar, G. Manson and G. Park. The fundamental axioms of structural health monitoring.

Proceedings of the Royal Society A, 463, 1639-1664, 2007.

[A] P. Zakian and K.J. Bathe. Transient Wave Propagations with the Noh-Bathe Scheme and the Spectral Element Method. Computers & Structures, 254, 2021

[A] L. Zhang and K.J. Bathe. Overlapping finite elements for a new paradigm of solution. Comput Struct 2017;187:64-76.

[A] L. Zhang, K.T. Kim, and K.J. Bathe. The New Paradigm of Finite Element Solutions with Overlapping Elements in CAD – Computational Efficiency of the Procedure. Computers & Structures, 199, 1-17, 2018.

[A] J. Zhang, Y. Liu and D. Liu. Accuracy of a composite implicit time integration scheme for structural dynamics, International Journal for Numerical Methods in Engineering, 109, 368–406, 2017.

[A] H. M. Zhang and Y. F. Xing. Optimization of a class of composite method for structural dynamics, Computers & Structures, 202, 60-73, 2018

[A] X. Zhou and K. K. Tamma. Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics. International Journal for Numerical Methods in Engineering, 59, 597-668, 2004.

[A] O.C. Zienkiewicz and Y.K. Cheung. Finite Element Method in Structural & Continuum Mechanics. McGraw-Hill 1967

[A] O.C. Zienkiewicz and P.B. Morice. The finite element method in engineering science. McGraw-Hill, 1971

[A] O.C. Zienkiewicz and J.Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Methods Engrg. 24 (1987) 337–357

INDEX

A

B

C

D

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Zero mass effect