Co-rotational Meshfree Formulation For Large Deformation Inelastic Analysis Of Two-Dimensional Structural Systems

By

Louis Lee Yaw
B.S.E. (Walla Walla College) 1992
M.S. (University of California, Davis) 1996

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Professor Sashi K. Kunnath (Chair)

Professor N. Sukumar

Professor Boris Jeremić

Committee in Charge
2008

Co-rotational Meshfree Formulation For Large Deformation Inelastic Analysis Of Two-Dimensional Structural Systems

Louis Lee Yaw
September 2008
Civil and Environmental Engineering

Co-rotational Meshfree Formulation For Large Deformation Inelastic Analysis Of Two-Dimensional Structural Systems

Abstract

Current computational techniques based on standard finite element methods to simulate large deformations associated with collapse limit states have several limitations. These limitations are primarily due to the nature of collapse which typically involves very large displacements and rotations. The use of modern finite element procedures to solve such problems can lead to inaccurate results or prevent completion of the computer analysis altogether. To overcome these limitations, a meshfree analysis using moving least squares (MLS) basis functions and continuous blending is developed using nodal integration with stabilization for two-dimensional continua. The primary focus of this work is to develop a framework that allows meshfree methodology to be embedded into a finite element-based formulation (or vice-versa) and thereby enabling the simulation of large-deformation structural response to complex loads. Provision is made to consider inelastic material behavior using $J2$ elasto-plasticity with isotropic and or kinematic forms of hardening.

The meshfree analysis is first developed for small strains, small displacements and rotations and applied to representative plane stress problems to validate the methodology. The formulation is then applied to frames comprising beam-columns of I-shaped cross-section and numerically simulated responses are compared to experimental results. Next, with the ultimate goal of improving structural collapse simulation, the meshfree method is extended to large displacements and rotations by incorporating a co-rotational formulation within a small-strain framework. Furthermore, the analysis is implemented by using the recently developed maximum-entropy basis functions which allow for easier imposition of essential boundary conditions. Versions of the implementation allow for either load or displacement control for both geometric and material nonlinear analysis. The meshfree co-rotational formulation is applied to various benchmark problems for validation. Finally, the meshfree co-rotational formulation is applied to collapse type problems that include stiffness softening, cyclic loading and the development of catenary action. Results are compared to advanced finite element or experimental results. Preliminary findings indicate that a meshfree co-rotational analysis is a feasible alternative to finite element simulations for large deformation problems in structural engineering. Further research is warranted to extend the method to more complex material behavior, finite strains and three-dimensional continua.

To my parents,

Gloria Jean Yaw and Louis Fay Yaw,

for all their support and encouragement.

Contents

List of Figures

List of Tables

Acknowledgments

My dissertation committee members have been so very helpful to me during this learning process of research and dissertation preparation. Professor Jeremić helped me several times particularly when I had questions regarding theory and algorithms in plasticity. It was his experience that helped me to realize that global and constitutive state variables should not be updated until the end of iterations within a given increment. He also suggested that I should derive analytical solutions for an elasto-plastic cantilever beam. This turned out to be very helpful during the validation process and the derivations for rectangular and I-shaped cross-sections are given in Appendix B. Professor Sukumar's experience in meshfree methods, the various types of shape functions and computational methods in general has been invaluable. His development of maximum-entropy shape functions in their latest form has greatly simplified my research goals. It was his excellent meshfree class that gave me a good head start on my research. I really appreciate the many helpful conversations he has had with me when I was stuck or trying to learn something new, particularly the implementation of the meshfree algorithms and using LaTeX to prepare journal articles or my dissertation. Professor Kunnath has been very patient and encouraging to me during every phase of my research. His experience in nonlinear analysis has helped me to interpret results when I have been uncertain. He has also provided me with excellent guidance during the research phase and during preparation of this thesis. Many times his suggestions have guided me to compare results with other sources such as OpenSees or LS-DYNA and as a result get past bothersome hurdles. Each time I left his office I have felt hopeful and inspired to work hard and make more progress. I have been very fortunate to have such helpful and encouraging committee members. This has certainly been a team effort and I would not have been successful without their help.

Several pieces of software made freely available on the internet are greatly appreciated. First, the Voronoi diagram generator created by Steve Fortune of AT&T Bell Labs and subsequently modified and made available by Shane O'Sullivan as part of his Master's project at the University of Limerick, Ireland was very helpful. Second, the TNT (Template Numerical Toolkit) of C++ matrix classes and JAMA/C++ linear algebra package made available by Roldan Pozo of NIST (National Institute of Science and Technology) have turned out to be extremely useful for my programming purposes. Third, the reverse Cuthill McGee program `triangulation_rcm.C` provided by John Burkardt significantly improved efficiency of the banded matrix solver that I created to solve the linear algebraic equations in my program. Last, I'd like to thank professor Mark Rashid for providing the simple quadrilateral mesh generator in his finite element class. Without these software packages it would have been much more difficult to complete my research goals.

I appreciate and acknowledge the financial support provided by Walla Walla University during the first two years of my doctoral program. I appreciate the leave of absence that was granted to me so that I could have time to undertake my graduate studies.

Last, I would like to express my gratitude to my savior Jesus Christ for His mercy and many blessings in my life.

Chapter 1

Introduction

Design of structural systems must take into account all feasible design loads that the structure must resist during its lifetime of service. In modern times, however, structural engineers recognize that additional considerations beyond traditional design loads, involving exceptional and extreme events, are becoming a necessary part of design. In particular, an understanding of structural behavior during partial or complete collapse is sometimes needed to assess the overall integrity of the structural design under extreme loading scenarios. Structural behavior during collapse is generally quite complex and much research is still needed in this area. Obviously, it is possible to construct model structures in a laboratory, induce loading that leads to collapse and record the observed behavior. While collapse tests of typical members, connections and partial frames are feasible, the testing of a range of such subassemblies or complete structures is often cost intensive. An alternative is to construct computer models that are sufficiently sophisticated to simulate structural collapse. Inevitably some laboratory testing is necessary to validate and calibrate computer models. Current computational techniques to simulate collapse of structures have several limitations. These limitations are primarily due to the nature of collapse which typically involves very large displacements and rotations and the associated limitations in existing computational approaches to solve such problems.

Modern methods in structural computation have evolved from finite-element based algorithms (Zienkiewicz and Taylor [98]). Most structural engineering problems are readily solved using finite element methods (FEM), which require the discretization of the spatial domain into a collection of elements. However, large-deformation problems, such as those resulting from extreme loads and leading to partial or total collapse, are unwieldy and difficult to solve with traditional mesh-based methods.

These large deformation problems in mesh-based (FE) methods usually require remeshing and mapping state variables to the new mesh - a process that is computationally demanding and prone to numerical errors (Lee and Bathe [51]). Also, since finite element formulations require preservation of the continuum, separation (or breaking) of elements is difficult to simulate unless the entire element is removed or remeshing is done. In the absence of remeshing, large mesh distortions drastically reduce the solution accuracy or impede meaningful computations altogether because the Jacobian in a severely distorted element can become zero or negative. At the limit state of failure, there also arise issues related to material softening that require complex algorithms to avoid mesh sensitivity (Simo et al. [79] and Crisfield and Willis [30]). The advent of meshfree (or element-free) methods can be attributed in large part to the issues and problems encountered in standard finite element formulations. Many of problems discussed above can be averted in meshfree methods since they are formulated to be sufficiently independent of a mesh and large distortions do not adversely affect the construction of the numerical approximation. In particular, collapse evaluation of frame structures is an open problem requiring large deformation analysis and inelastic material modeling. Meshfree methods are well-suited for such problems and are likely to yield new insights into such phenomenon. Over the past decade, meshfree and particle methods (Belytschko et al. [16], Li and Liu [53], Fries and Matthies [36] and Atluri and Shen [7]) have come to the forefront for the solution of fracture (material separation) and large deformation problems. This class of methods has been demonstrated to be in many instances more flexible and superior to finite element methods. For example, the modeling of large inelastic deformations, material softening, and failure phenomena that are prevalent in large-scale structural systems are particularly amenable to meshfree computations. The focus of this research effort is to develop a novel computational approach for the modeling and analysis of structural systems to eventually enable collapse simulation.

1.1 Dissertation objectives and classes of problems considered

Although the current research does not conclude with the complete collapse simulation of a structure, it is intended to explore new technologies to eventually attain that goal. Hence, it is the objective of this dissertation to explore the feasibility of using meshfree methods for simulations of large scale structural systems undergoing large displacements and large rotations, with the intent of advancing future research in structural collapse simulation.

The objectives of the study are as follows:

1. The primary objective of the research is to advance computational simulation methodology in structural engineering so as to improve current capabilities in collapse and failure analysis of structures.

2. Explore meshfree methods as an alternative to finite element methods.

3. Explore the feasibility of using meshfree methods for simulations of large scale frame

structures for small strains, displacements and rotations.

4. Use meshfree methods in a co-rotational formulation and by so doing extend analysis capabilities to large displacements and rotations but with small strains.

5. The previous two objectives shall include consideration of material nonlinearity. While the present work will be limited to simple J2 elasto-plasticity with kinematic and isotropic hardening, extending the methodology to more general classes of materially nonlinear problems should be facilitated.

6. The developed methodology will be implemented in a special purpose computer program and computer simulations will be compared to benchmark analytical solutions, to results obtained from other finite element based software, and to experimental results available in the literature.

   The following classes of problems will be considered:

   (a) Simulations with both load and displacement control.

   (b) Small strains, displacements and rotations for linear elastic materials.

   (c) Small strains, displacements and rotations for J2 elasto-plasticity with linear hardening and/or exponential hardening.

   (d) Small strains, large displacements and rotations with linear elastic materials

   (e) Small strains, large displacements and rotations with J2 elasto-plasticity.

   (f) Cyclic loading of structures for some of the above classes of problems.

   (g) Loading of structures with constant axial load applied first, followed by a nonlinear analysis using displacement control.

7. Discuss the results obtained and identify advantages and issues/concerns with the methodology.

8. Draw conclusions on the current research effort and provide future research directions.

1.2 Original Contributions

As part of this dissertation a variety of items are considered original work. A search of the literature did not provide any indication that the items listed in this section are found elsewhere. The items listed below are given starting with the most significant original work.

1. Although co-rotational formulations using finite elements are quite common, a co-rotational formulation in a meshfree setting for a continuum does not appear in the literature. Specifically, a variationally consistent meshfree co-rotational formulation for 2D continuum type problems is developed and implemented as part of this dissertation. The formulation is similar to that developed by Crisfield and Moita [28]. The primary differences being the use of meshfree maximum-entropy basis functions, the generalization of the approach to n node neighbors, rather than just 4 nodes of a typical quadrilateral element, the choice of using each node's Voronoi cell centroid to enforce zero spin in order to determine the co-rotating frame angle of rotation, the consideration of both elastic and inelastic problems, the use of nodal integration by Chen et al. [20] and the inclusion of the stabilization technique of Puso et al. [72] in order to avoid locking and spurious modes. The co-rotational formulation as applied allows solution of problems with large displacements and rotations but with restriction to small strains.

2. A new methodology is presented for modeling beams for the case of plane stress with consideration of non-uniform thickness in the two-dimensional domain. Specifically, steel frames comprised of beams with I-shaped cross-section are modeled. The methodology is implemented for small strain small displacement problems using mov-

ing least squares (MLS) basis functions with the method of continuous blending to allow the enforcement of essential boundary conditions. The methodology is also implemented for small strain large displacement problems using the meshfree co-rotational formulation using maximum-entropy basis functions. In both cases, application of these techniques and basis functions to planar frame type problems is new.

3. An analytical solution for elasto-plastic cantilever beams of rectangular and I-shaped cross-section loaded at their free end is derived in Appendix B. The presented derivation is independent of work by others. However, some similarities of approach are found in the works by Yu and Zhang [97] and Phillips [71].

4. An analytical solution, for large displacements of an elastic cantilever with point load at its free end, is given in Appendix C. The solution is given in the work by Khosravi et al. [48]. The derivation is based on Euler-Bernoulli beam theory and does not consider axial strains. Khosravi et al. do not provide a reference for the solution given. In this dissertation the solution is extended to include axial deformations. As indicated, a similar approach can be used to include shear deformations.

1.3 Organization of remaining chapters

The remainder of this dissertation is organized as follows. In Chapter 2, a review of the literature discusses the ingredients generally necessary to model collapse simulations such as geometric and material nonlinearity and the increasingly popular use of co-rotational formulations. The chapter continues with a review of the current state of the art in collapse simulation. A discussion of the advantages and disadvantages of the current techniques leads to the idea that an alternative method of analysis may hold promise in advancing structural computation to overcome the drawbacks of existing techniques. Meshfree methods are thereby introduced as a potential alternative to finite element approaches. Chapter 3 continues with a literature review of meshfree methods, the advantages of meshfree methods, how meshfree methods have been used in the past and how the current research intends to take advantage of the current meshfree technology. Two variants of meshfree basis functions and their implementation are also introduced. Chapter 4 is devoted to the implementation and extension of meshfree to the analysis of planar frames for small strains, small displacements and rotations. In Chapter 5 the previous implementation is extended to large displacements and rotations by introducing a co-rotational formulation for a 2D continuum. Applications and validation of the proposed meshfree methodology is provided in Chapter 6 through a variety of example problems that are compared to analytical solutions, results from FE simulations and to experimental results when possible. Lastly, Chapter 7 concludes with general observations on the current research and a discussion on means to extend the proposed methodology in the area of collapse simulation. Finally, recommendations for future research are suggested.

Chapter 2

Literature Review

Collapse of large-scale civil engineering structures involves large displacements, large rotations and relatively smaller strains. Under such conditions the material response is inelastic in regions of extreme stress. At the limit state near collapse, large strains and fracture are also likely. To simulate such behavior computational models need to consider both geometric and material nonlinearities. Having this in mind a review of literature pertinent to the issues related to the development of a large-displacement analysis framework is provided in this section. The review starts with methods that consider geometric nonlinearities.

2.1 Geometric nonlinearities

In essence, when material non-linearities are excluded, geometric nonlinearities result when the forces required to cause structural deformation are a nonlinear function of the displacements (see Figure 2.1). Except for very simple problems, closed form solutions are very difficult if not impossible to obtain. For this reason it is necessary to resort to incremental iterative methods using computer-based simulations. In most cases this is done using the finite element method (FEM), although in some cases it is possible to use numerical

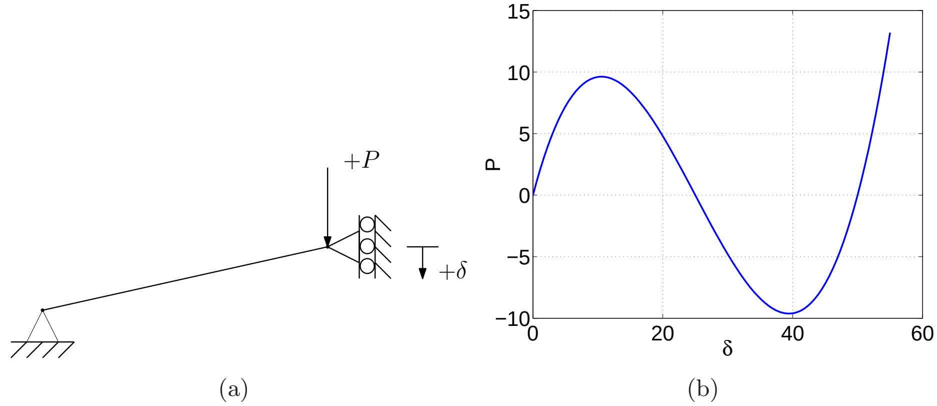


Figure 2.1: Geometric nonlinearity: (a) shallow truss example; (b) load P as a nonlinear function of displacement δ.

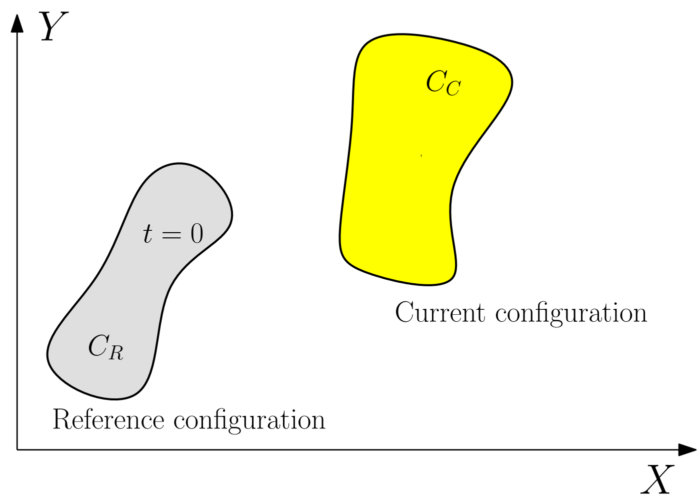


Figure 2.2: Reference and current configurations of an arbitrary deformable structure.

methods such as finite differences or boundary element methods (BEM). Geometric nonlinearities are essential in collapse simulation because they capture the effects of buckling, large changes in structure shape and the changes in internal forces necessary to keep the structure in static equilibrium.

In the conventional finite element method, geometric nonlinearities are accounted for by considering finite strains. It is possible to show that if small strains are used, with large rotations that often result with large displacements, rigid body motions cause erroneous strains to develop within the structure. This obviously should not be the case because valid rigid body motions are strain-free. It is for this reason that a valid measure of finite strain is required for such problems. Consideration of finite strains is usually implemented within either a Total Lagrangian (TL) or Updated Lagrangian (UL) approach.

To explain the different approaches consider Figure 2.2. The structure is in the *reference* configuration initially, usually at time $t = 0$. Prescribed loads and displacements cause the structure to deform, so that at some time $t > 0$ the structure reaches the *current* configuration. In statics, the application of such prescribed loads and displacements takes place over a sufficient period of time so that it is valid to neglect dynamic effects. In this dissertation it is assumed that all problems are time independent and hence the assumptions of statics are valid.

In a TL approach all static and kinematic variables are expressed in terms of the reference configuration. Conversely, in a UL approach all static and kinematic variables are expressed in terms of the most recent current configuration of the displaced structure. A rectangular coordinate system is usually attached to each finite element of the discretized structure in a UL formulation. Updating of these coordinates takes place in every step of the incremental analysis. The interested reader is referred to Belytschko et al. [17] for further details on general TL and UL formulations. Analytical, TL, UL and co-rotational derivations for the shallow truss problem of Figure 2.1 are presented by Mattiasson [63]. The TL and UL approaches are commonly used in solid mechanics type problems, however, they are also used for frame structures which constitute the primary structural type considered in this dissertation. Hence, the literature review in this section discusses how the finite element Lagrangian approaches have been used for large displacement analysis in frames. The literature review by Schulz and Fillipou [74] provides an excellent overview of developments leading to large-deformation FE analyses.

It appears that the first TL and UL formulations for 3D beam elements in incremental form were given by Bathe and Bolourchi [11]. In their paper, they demonstrate that the TL and UL formulations yield identical stiffness matrices and nodal point force vectors. They also note that the UL formulation is computationally more efficient. These incremental forms of TL and UL, due to their computational effectiveness, became very popular and were then applied to many structural problems, such as thin-walled beams in the work by Conci [24], in which a UL approach is employed.

It is evident from the literature that 3D frame problems with large displacements require special treatment due to the complex nature of finite rotations as discussed by Argyris [3]. By recognizing these difficulties a general treatment of 3D beams with large rotations, using a TL approach, is given by Simo and Vu-Quoc [77, 81].

A variety of beam theories have been implemented for large displacement finite element analysis. For example, the Bernoulli hypothesis is used in the work of Jelenić [42]. Timoshenko type beam elements are implemented in a TL formulation by Crivelli and Felippa [31]. The paper by Simo and Vu-Quoc [81] makes use of the Kirchhoff-Love model. The choice of a particular beam theory is often based on problem specific requirements or sometimes to simply demonstrate the validity or appropriateness of a new computational or analytical approach.

In addition to TL and UL formulations, an extremely popular method for large displacement and large rotation analysis of framed structures is based on the co-rotational formulation. In fact, one might argue that this has become the method of choice for such problems when using beam elements. More discussion of the co-rotational formulation is given later on in this dissertation, however, the following basic ideas are noted. A co-rotational analysis separates rigid body motion from strain producing deformations. This is done in finite element formulations by attaching a local co-rotating coordinate frame to each element in the structure. With respect to the local co-rotating frame the rigid body deformations are negligible and hence strains and subsequently stresses are calculated based on the local element displacements. Relations between local variables and global variables allow the determination of the large deflections and rotations that take place at the global level. Co-rotational formulations are a relatively easy way to introduce the geometrically nonlinear effects of large deflections and large rotations, while using small strain assumptions at the local level.

Large rotations were first treated in a *variationally consistent* co-rotational formulation for 3D beams by Crisfield [25]. Crisfield's paper notes that researchers had previously used a co-rotational formulation for beam elements, but that they had not used a consistent derivation of the internal forces and tangent stiffness matrix. In fact, some authors had only applied transformation matrices within the co-rotational formulation to the standard linear stiffness matrix. This approach does not account for the geometric stiffness matrix which arises by taking the variation of the transformation matrices. A variationally consistent formulation is necessary to achieve quadratic convergence during Newton-Raphson iterations for global equilibrium.

Some noteworthy developments in co-rotational formulations for beam elements have been reported by several researchers. Urthaler and Reddy [91] develop a 2D co-rotational beam formulation for Euler-Bernoulli, Timoshenko, and simplified Reddy theories. Hence, they demonstrate a method to include shear deformations and the necessary details to avoid shear locking. It bears mentioning that for the 2D case element tangent stiffness matrices in a co-rotational formulation are symmetric. However, this has traditionally not been the case when formulations are extended to 3D due to the non-commutativity of spatial rotations. The resulting non-symmetric element stiffness matrices require more storage and appropriate solvers must be used when solving the resulting global system of equations. It appears that Li [54] is the first to construct a formulation for 3D beams using vectorial rotation variables and by so doing the stiffness matrices at the element and global levels are symmetric. This has important computational advantages over previous formulations. Li [55] also gives an interesting co-rotational formulation for 2D beam elements. This formulation is noteworthy because of the use of a mixed formulation using the Hellinger-Reissner functional, the use of vectorial rotation variables, adoption of Green strains and successful avoidance of shear locking. Lastly, mention is made of the developments given by Crisfield [26, 27] in which various large displacement formulations, including co-rotational formulations, for beam elements in 2D and 3D are given.

Finally, it is noted that co-rotational formulations have much in common with the *natural* approach introduced by Argyris et al. [4]. The natural approach has been used for geometrically nonlinear problems and recognizes the advantages of separating rigid body motions from strain producing deformations just like in co-rotational formulations.

2.2 Material nonlinearities

To simulate large deformations that lead to post-elastic response in structures, it is generally necessary to account for material nonlinearities. For the purposes of this study, only materials related to large-scale civil engineering structures are addressed. In particular, the most common material models attempt to reproduce the behavior of metals or reinforced concrete.

The easiest way to consider material nonlinearities is by means of plastic analysis. Many texts, particularly in steel design, devote chapters to plastic design, which is the study of the development of concentrated plastic hinges and the determination of the collapse load at which the structure forms an unstable mechanism due to plastic hinges, see for example

Li and Li [52]. McGuire et al. [66] explain that a plastic analysis assumes that the beam cross-section has only two possible states, namely, (i) the cross-section is completely elastic if the maximum stress is less than or equal to the yield stress, or (ii) perfectly plastic across the depth of the beam with tensile and compressive stresses constant at the specified yield stress. This type of analysis is easy to implement in a computer analysis. However, a fine discretization along beam elements is necessary to allow the location of hinges to develop accurately. A more accurate analysis arises by allowing for the gradual development of inelasticity across the beam depth. This is referred to as a *distributed plasticity* approach as explained by Liew et al. [56]. Much research is devoted to this topic and only a representative list and some relevant details are given here. Liew et al. [56, 57] use a second-order refined concentrated plastic hinge method to obtain many of the benefits usually obtained by the distributed plasticity approach. By so doing they obtain comparable results but with greater computational efficiency. Thus far the approaches mentioned are displacement-based approaches. As indicated by Alemdar and White [2] these approaches have some serious drawbacks due to the inability of the simple displacement polynomials to correctly represent the highly nonlinear curvature that results along the beam length in a distributed plasticity formulation. In fact, Alemdar and White [2] provide a comparative study of different approaches, displacement-based, flexibility-based and mixed-based formulations. In particular, the flexibility based formulations do not have the same drawbacks as the displacement-based approaches. For this reason researchers have recently devoted much effort to this type of formulation, see for example Sivaselvan and Reinhorn [82] who apply a flexibility-based approach to the collapse analysis of plane frames. Alemdar [1] provides both flexibility and mixed-based approaches for distributed plasticity in steel frames. The primary benefit of a flexibility-based approach is the ability to use one element for a given frame member rather than the required multiple element discretization used in the displacement-based approach [75]. As indicated in the excellent review of Scott and Fenves [75] a discussion of the benefits of the flexibility approach is provided by Neuenhofer and Filippou [69]. Lastly, the reader is referred to the fiber beam element using a flexibility approach by Taucer et al., which has been implemented in the open source software package OpenSees [65]. Fiber-based beam elements allow a precise definition of cross-section shape and, during the global incremental analysis, integration of the fibers across the beam depth keeps track of the state of distributed plasticity. Various other researchers have also used fiber-based beam elements such as Torkamani and Somnez [90].

The research cited above considers material nonlinearities and often also includes geometric nonlinearities. It is possible to include material nonlinearities in TL, UL, natural and co-rotational approaches. It seems however from the literature that this is most easily done at the local level in co-rotational formulations. By combining both types of nonlinearity it is possible to model plastic and geometric instabilities, which is demonstrated in many of the previously cited works. In particular, the work by Battini and Pacoste [12] specifically is formulated to account for plastic instability of 3D beams using a co-rotational formulation.

In the reserach above, the focus is on general schemes for including material nonlinearity in beam type elements, however, when a structure is modeled as a continuum, more advanced material models in 2D or 3D are commonly used. Kojić and Bathe [49] present a variety of material models that consider metal plasticity, creep, viscoplasticity, soil plasticity and even large strain plasticity. In this dissertation, the simplest continuum material model of $J2$ plasticity is used since it is an effective model for metals. It is for this reason that the chapters on applications mostly focus on steel framed structures that exhibit inelastic material behavior. The reader is also referred to the works of Simo and Taylor [80] and the monograph by Simo and Hughes [78], which provide details on $J2$ plasticity in 2D and 3D and inelastic material behavior in general.

2.3 State of the art collapse analysis

By taking into account geometric and material nonlinearities it is possible to numerically solve certain types of collapse analysis problems. In fact, from an engineering standpoint many of the works cited above are constructed specifically to create the capability to study structural collapse. Hence, a review of literature is provided in this section which focuses specifically on collapse research.

A common approach to collapse analysis is the alternate path method. In this method one removes an individual column or individual beam and determines the resulting loads on the rest of the structure caused by the structural element removal. The loads once supported by the removed element, find an alternate path through the remaining structural system. If the capacities of the remaining structural elements are adequate, then collapse will not occur. However, if the remaining structural elements are not adequate to resist the additional loads, collapse may take place and/or there will be local deficiencies. When considering global progressive collapse Ettouney et al. [34] point out that the commonly used alternate path method is not necessarily adequate and that consideration of the global response of the structural system should be made. Some simple methods to ascertain global system stability are provided in their paper. It is also recognized that when members fail due to overloading due to some extreme event, these members may fall, with impact, upon other members. This obviously is a repeatedly occurring situation during progressive collapse. Hence, Kaewkulchai and Williamson [45] provide a modeling technique to account for the rigid body impact of one structural member upon another. They also provide an example which demonstrates the importance of accounting for impact during progressive collapse.

In particular, they note that impact velocity is the crucial factor in determining if in fact structural elements will be damaged.

Modeling of collapse due to blast loads is illustrated by Luccioni et al. [60]. In their work very advanced modeling is undertaken to simulate the behavior of the structure during the pressure wave of the blast. They point out that this type of modeling is much more complicated than wind or seismic loading type conditions, since architectural elements also play a crucial role in the transfer of pressure wave loads to the building structural system. The results of their analysis of a reinforced concrete structure due to simulated blast loading are compared to the same structure that was destroyed by a terrorist bomb detonation in real life.

Villaverde [92] gives an excellent review of literature and summarizes methods to assess collapse potential of structures. In particular, he describes collapse analysis techniques that fall into the following categories: single-degree-of-freedom models, nonlinear static procedure, step-by-step finite-element analyses and detection of abrupt response increase, incremental dynamic analyses and shake table collapse experiments. Similar information is provided by Marjanishvili and Agnew [61] who review four techniques to assess collapse vulnerability: linear-elastic static, nonlinear static, linear-elastic dynamic, and nonlinear dynamic methodologies. Another excellent review of literature on progressive collapse and comparison of codes and standards is provided by Mohamed [67].

The work mentioned previously, by Sivaselvan and Reinhorn [82], provides the results of both static and dynamic collapse analyses for some simple structural examples. Their approach is for 2D structures using a co-rotational and flexibility-based formulation which also incorporates nonlinear material behavior.

As mentioned by Mohamed [67], literature on collapse greatly increased following 2001. It is also evident from the literature that much research is still needed. For example, although many analysis methods are ready to be implemented to study collapse, it seems that focus on systematic studies of collapse behavior using current analysis techniques is still lacking. Part of the reason for this seems to be that analysis techniques are only recently becoming capable of performing meaningful simulations. For this reason the present research is timely and relevant in that it aims to improve the capability for accurate collapse simulation.

Recent research by Khandelwal et al. [47] and Bao et al. [10] provides further impetus for creating computational methods for the advanced simulation of structures. Specifically, Khandelwal and co-workers provide a study which uses high-fidelity finite element methods on steel structure subassemblies, such as beam-column joints. Bao and co-workers similarly use high-fidelity finite element methods on reinforced concrete beam-column joints. The results from both of these advanced numerical studies are then used to calibrate simpler macromodels that are computationally more efficient for the solution of global structural collapse. It is envisioned that the meshfree methods developed in this dissertation will eventually lead to another tool with which to calibrate macromodels for global structural analysis.

2.4 Observations and conclusions

The literature review suggests that computer-based analyses to model collapse should have the following characteristics.

1. The methodology should incorporate both geometric and material nonlinearities.
2. The formulation should consider distributed plasticity, thereby allowing the spread of inelasticity to occur both across the depth of the section and along the length of the member.

3. Beam models should be based on exact theories, that include for example large strains, shear deformations and allow for arbitrary beam cross-section distortions.

4. Special attention must be given to avoid shear locking.

5. The modeling should account for highly nonlinear curvature along beams due to the formation of plastic hinges.

6. Mesh distortions should not affect the computational simulation.

7. Incremental iterative techniques should be used to capture material and geometric softening. Arc length control is the most robust way to handle this requirement.

8. The approach should be able to handle composite materials to enable fiber modeling.

9. The methodology is computationally efficient and should be able to model global structural behavior so that progressive collapse can be observed.

10. Consideration of dynamic effects must be included since contact and impact effects should be accounted for.

The preceding list is the motivation for the work contained in this dissertation. It is the intent of the research presented herein to address many of the shortcomings of previous methods for collapse simulation. Although many of the items in the list have been addressed there are two issues in particular that tend to cause erroneous results or cause simulations to stop altogether. Items 5 and 6 tend to limit displacement-based methods and item 6 can affect flexibility-based methods. The direction that this research attempts to take is expected to eliminate or alleviate these issues. The items that are not yet addressed in this research include large strains (identified in item 3), the implementation of arc length control (listed under item 7), and consideration of composite materials (listed in item 8).

Although arc length control is not implemented, displacement control is implemented and is adequate for the given objectives. Regarding item 9 much of the research here is quite new and it is certainly the case that computational efficiency needs to be improved. Item 10, is not addressed which effectively removes the ability to consider problems of contact or impact. Aside from these items all other characteristics are addressed as discussed below.

Geometric nonlinearities are addressed by use of a co-rotational formulation. A simple $J2$ flow plasticity material model is implemented which includes linear and exponential forms of isotropic hardening as well as kinematic hardening. This material model is appropriate for steel. The approach taken here uses a 2D continuum for plane stress, which automatically attains the distributed plasticity requirement. The meshfree continuum approach is based on elasticity theory (for small strains) and hence does take into account shear deformations and places no limitations on beam cross-section distortions. As is demonstrated later in the dissertation, shear locking is avoided. As long as the discretization of the domain is sufficient, nonlinear curvature along beams is properly accounted for. The model created herein uses meshfree technology and hence is unaffected by mesh distortions. Both load control and displacement control methods are implemented, which are sufficient to demonstrate the effectiveness of the approach. Theoretically, global structural behavior during progressive collapse is possible with the current implementation and careful study of member forces during the analysis can lend insight into the evolution of alternate paths during collapse.

The summary in the preceding paragraph forms the basis of the developments described in the remaining chapters. Meshfree literature and theory is presented next.

Chapter 3

Meshfree Methods

Most structural engineering problems are readily solved using finite element (FE) methods, which require the discretization of the spatial domain into a collection of elements. However, FE methods encounter a host of issues in nonlinear structural analysis in applications involving cyclic and extreme loads at the limit state near collapse. Continuing research efforts to address these problems remain in the realm of FE methodology with the result that strategies applied to one class of problems may not be valid for another. The elements which make up the mesh in FE simulations must be predefined. By contrast, the discretization of a domain without resorting to a predefined mesh forms the basis of meshfree (or element-free) methods. A meshfree method typically requires only the specification of nodes (both within the domain and on the boundary) to define the domain without the need for any specific connectivity information between the nodes. Since the first formal introduction of a meshfree Galerkin method, the so-called diffuse element method by Nayroles et al. [68], many variants of element-free approaches have been proposed by Belytschko et al. [18], Liu et al. [59] and Atluri and Zhu [8] among others.

The literature on meshfree methods is vast and comprehensive. The reader is referred to overview papers by Belytschko et al. [16], Li and Liu [53] and Fries and Matthies [36] for additional details on theory and applications. Most of the structural applications to date have been limited to problems in solid mechanics. With the possible exception of Weitzmann [94], who applied meshfree methods to concrete shear walls, which are then coupled to FE beam and column line elements of a building frame structure, very little effort has been devoted toward extending meshfree methods to applications in large-scale structural engineering. In particular, collapse evaluation of frame structures is an open problem requiring large deformation analysis and inelastic material modeling. Meshfree methods are well-suited for such problems and are likely to yield new insights into such phenomenon.

Meshfree methods are now routinely used for many specialized applications in computational mechanics. Besides the fact that the task of accurate mesh generation in finite element methods can be time-consuming and computationally demanding (particularly for problems requiring remeshing), the growing popularity of element-free methods stems from its ability to solve certain classes of problems that are unwieldy and difficult to solve with traditional mesh-based methods. For example, large deformation problems in mesh-based (FE) methods usually require remeshing and mapping state variables to the new mesh—a process that is prone to numerical errors. In the absence of remeshing, large mesh distortions drastically reduce the solution accuracy or impede meaningful computations altogether because the Jacobian in a severely distorted element can become zero or negative. This problem is averted in meshfree methods since they are formulated to be sufficiently independent of a mesh and large distortions do not adversely affect the construction of the numerical approximation.

In this section two variants of meshfree shape (basis) functions are introduced. First, moving least squares shape functions are derived using standard arguments. Second, a recently developed type of basis function, based on the maximum-entropy (max-ent) principle is presented. In each case, examples are given for one-dimensional and two-dimensional basis functions.

3.1 Moving least squares basis functions

Shape functions in meshfree methods are constructed independent of an underlying mesh structure. This is the main distinction of meshfree methods as opposed to finite element interpolants. Moving least squares (MLS) approximants as given in Lancaster and Salkauskas [50] are widely used in meshfree Galerkin methods (see Belytschko et al. [16]), and a variant of MLS shape functions is used in this study. For a review of the most commonly used meshfree approximation schemes, the interested reader can refer to Sukumar and Wright [85].

3.1.1 MLS shape function derivation

Lancaster and Salkauskas [50] use a weighted least squares approach to derive the MLS shape functions (see Appendix A for common meshfree terminology and a weighted least squares derivation of meshfree shape functions). The shape functions are also obtained by imposing the polynomial consistency (reproducing) conditions as given by Belytschko et al. [16], which is the approach presented here.

In two dimensions, the moving least squares approximant for a vector-valued function $\boldsymbol{u}(\boldsymbol{x})$ is written as

$$\mathbf{u}^h(\boldsymbol{x}) = \sum_{a=1}^{n} \phi_a(\boldsymbol{x})\mathbf{d}_a \equiv \boldsymbol{\phi}^T\mathbf{d}, \tag{3.1}$$

where $\phi_a(\boldsymbol{x})$ are the nodal shape functions, $\mathbf{d}_a$ is the nodal coefficient vector for node a, and n is the number of nodes in the neighborhood of $\boldsymbol{x}$ such that $\phi_a(\boldsymbol{x}) \neq 0$. In Belytschko et al. [16], the MLS shape function $\phi_a(\boldsymbol{x})$ is assumed to be of the form

$$\phi_a(\boldsymbol{x}) = \mathbf{p}^T(\boldsymbol{x}_a)\boldsymbol{\alpha}(\boldsymbol{x})w(\boldsymbol{x}_a), \tag{3.2}$$

where $\mathbf{p}(\boldsymbol{x}) = \{1\ x\ y\}^T$ is a linear basis in two dimensions, $\boldsymbol{\alpha}(\boldsymbol{x})$ is a vector of unknowns to be determined and $w(\boldsymbol{x}) \geq 0$ is a weighting function.

The vector of unknowns, $\boldsymbol{\alpha}(\boldsymbol{x})$, is determined by imposing the consistency (reproducing) condition, i.e., the shape function must exactly reproduce $\mathbf{p}(\boldsymbol{x})$. Hence, ϕ_a must satisfy

$$\mathbf{p}(\boldsymbol{x}) = \sum_{a=1}^{n} \mathbf{p}(\boldsymbol{x}_a)\phi_a(\boldsymbol{x}). \tag{3.3}$$

Now, substituting Eq. (3.2) into Eq. (3.3) yields

$$\mathbf{p}(\boldsymbol{x}) = \Big[\sum_{a=1}^{n} \mathbf{p}(\boldsymbol{x}_a)\mathbf{p}^T(\boldsymbol{x}_a)w(\boldsymbol{x}_a)\Big]\boldsymbol{\alpha}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{x}), \tag{3.4}$$

which gives

$$\boldsymbol{\alpha}(\boldsymbol{x}) = \boldsymbol{A}^{-1}(\boldsymbol{x})\mathbf{p}(\boldsymbol{x}). \tag{3.5}$$

Upon substitution of $\boldsymbol{\alpha}(\boldsymbol{x})$ into Eq. (3.2) the final shape function expression is

$$\phi_a(\boldsymbol{x}) = \mathbf{p}^T(\boldsymbol{x}_a)\boldsymbol{A}^{-1}(\boldsymbol{x})\mathbf{p}(\boldsymbol{x})w(\boldsymbol{x}_a). \tag{3.6}$$

The weight function provides the local character of the shape function. For example, the shape function ϕ_a has a radius of support, ρ_a, within which it is nonzero. This is best illustrated in one dimension (see Figure 3.1), where the following quartic weight function

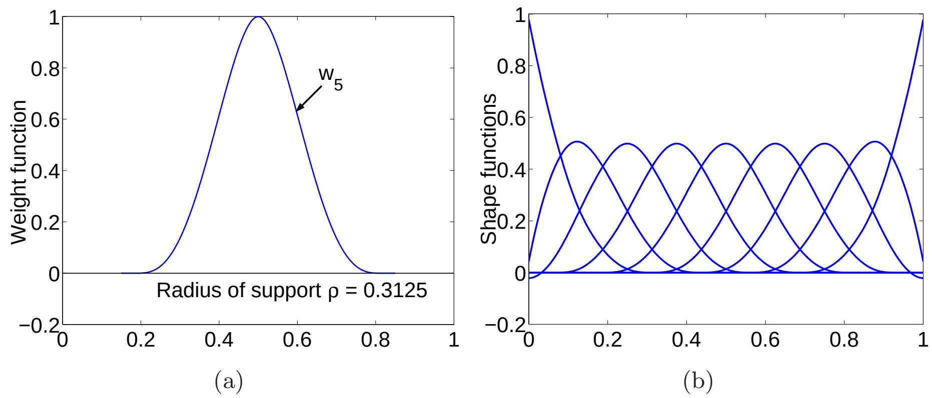


Figure 3.1: MLS construction (9 equi-spaced nodes): (a) weight function, (b) shape functions.

is used to generate the shape functions:

$$w(q) = \begin{cases} 1 - 6q^2 + 8q^3 - 3q^4 & q \leq 1 \\ 0 & q > 1 \end{cases}, \tag{3.7}$$

and $q = \|\boldsymbol{x} - \boldsymbol{x}_a\|/\rho_a$. Note that the shape functions do not interpolate on the boundary ($\phi_a(\boldsymbol{x}_b) \neq \delta_{ab}$). This characteristic makes it difficult to impose essential boundary conditions. In the following chapter, in Section 4.3, a discussion is given regarding how to solve this problem of correctly imposing boundary conditions. An alternative to MLS basis functions that allows easier imposition of essential boundary conditions is presented next.

3.2 Maximum-entropy basis functions

In meshfree Galerkin methods, moving least squares (MLS) approximants [50] and natural neighbor interpolation schemes [22, 76] have been widely used, whereas maximum-entropy basis functions are of more recent origin [5, 84]. For general overviews of meshfree methods and meshfree approximants, the interested reader is referred to Belytschko et al. [16], Li and Liu [53], and Sukumar and Wright [85]. In this dissertation, starting with Chapter 5, maximum-entropy basis functions are used to construct the trial and test approximations that appear in the weak form. Maximum-entropy basis functions satisfy a weak Kronecker-delta property on the boundary, which greatly simplifies the imposition of essential boundary conditions [5].

In two dimensions, the constant and linear reproducing conditions

$$\sum_{a=1}^{n} \phi_a(\boldsymbol{x}) = 1 \tag{3.8}$$

and

$$\sum_{a=1}^{n} \phi_a(\boldsymbol{x}) \boldsymbol{x}_a = \boldsymbol{x}, \tag{3.9}$$

do not prescribe unique basis functions if $n > 3$. The Shannon entropy in Reference [84] and a modified entropy functional in Reference [5] are used to regularize the problem to obtain unique basis functions for any n. The entropy functional of Arroyo and Ortiz [5] is generalized in Sukumar and Wright [85] on using the notion of a *prior* (or weight function) within the Shannon-Jaynes entropy functional.

The variational formulation for maximum-entropy basis functions using the Shannon-Jaynes entropy functional is: find $\phi_a(\boldsymbol{x}) \geq 0$ as the solution of the following constrained optimization problem:

$$\max_{\boldsymbol{\phi} \in \mathbb{R}_+^n} \left[-\sum_{a=1}^{n} \phi_a(\boldsymbol{x}) \ln \left(\frac{\phi_a(\boldsymbol{x})}{w_a(\boldsymbol{x})} \right) \right], \tag{3.10a}$$

subject to the linear reproducing conditions:

$$\sum_{a=1}^{n} \phi_a(\boldsymbol{x}) = 1, \tag{3.10b}$$

$$\sum_{a=1}^{n} \phi_a(\boldsymbol{x})(\boldsymbol{x}_a - \boldsymbol{x}) = \boldsymbol{0}, \tag{3.10c}$$

where $w_a(\boldsymbol{x})$ is a prior estimate (weight function), and $\mathbb{R}^n_+$ is the non-negative orthant. The prior weight, $w_a(\boldsymbol{x})$, is the initial estimate of the basis function $\phi_a(\boldsymbol{x})$. If $w_a(\boldsymbol{x}) = 1$ for all a, then the Shannon entropy functional, $-\sum_a \phi_a \ln \phi_a$, is obtained. On using the method of Lagrange multipliers, the solution of the variational problem is [85]:

$$\phi_a(\boldsymbol{x}) = \frac{Z_a(\boldsymbol{x};\boldsymbol{\lambda})}{Z(\boldsymbol{x};\boldsymbol{\lambda})}, \quad Z_a(\boldsymbol{x};\boldsymbol{\lambda}) = w_a(\boldsymbol{x}) \exp(-\boldsymbol{\lambda} \cdot \tilde{\boldsymbol{x}}_a), \tag{3.11}$$

where $\tilde{\boldsymbol{x}}_a = \boldsymbol{x}_a - \boldsymbol{x}$ ($\boldsymbol{x}, \boldsymbol{x}_a \in \mathbb{R}^d$) are shifted nodal coordinates, $\boldsymbol{\lambda}$ are the d Lagrange multipliers associated with the constraints in (3.10c), and $Z(\boldsymbol{x}) = \sum_b Z_b(\boldsymbol{x};\boldsymbol{\lambda})$. A Newton method is used to solve the dual optimization problem ($\min \ln Z$) to obtain $\boldsymbol{\lambda}$; details on the computation of ϕ_a and $\nabla\phi_a$ are provided in References [5] and [85] for a uniform prior and a Gaussian prior, respectively.

The expressions for the derivatives of the maximum-entropy basis functions for any choice of a prior weight function are presented below. The notations and approach presented in Arroyo and Ortiz [5] are adopted. In what follows, it is assumed that $\boldsymbol{\lambda}$ is the converged solution for the Lagrange multipliers and $\nabla\phi_a$ is the gradient of the basis function. Equation (3.11) is written as

$$\phi_a(\boldsymbol{x};\boldsymbol{\lambda}) = \frac{\exp\left[f_a(\boldsymbol{x};\boldsymbol{\lambda})\right]}{\sum_{b=1}^{n} \exp\left[f_b(\boldsymbol{x};\boldsymbol{\lambda})\right]}, \quad f_a(\boldsymbol{x};\boldsymbol{\lambda}) = \ln w_a(\boldsymbol{x}) - \boldsymbol{\lambda} \cdot \tilde{\boldsymbol{x}}_a \tag{3.12}$$

where $\boldsymbol{\lambda}$ is implicitly dependent on $\boldsymbol{x}$. Using Eq. (3.12) yields

$$\nabla\phi_a = \phi_a \left(\nabla f_a - \sum_{b=1}^{n} \phi_b \nabla f_b \right). \tag{3.13}$$

Taking the gradient of f_a in Eq. (3.12) and simplifying results in

$$\nabla f_a = \frac{\nabla w_a}{w_a} + \boldsymbol{\lambda} - \tilde{\boldsymbol{x}}_a \cdot \nabla\boldsymbol{\lambda}, \tag{3.14}$$

where $\nabla\boldsymbol{\lambda}$ remains to be determined. To this end, on taking the total derivative of both sides of the equality $\mathbf{r}(\boldsymbol{x};\boldsymbol{\lambda}) = -\sum_a \phi_a(\boldsymbol{x};\boldsymbol{\lambda})\tilde{\boldsymbol{x}}_a = \mathbf{0}$, the following is obtained

$$D\mathbf{r} = \nabla\mathbf{r} + \nabla_\lambda \mathbf{r} \cdot \nabla\boldsymbol{\lambda} = \mathbf{0},$$

where $\nabla\mathbf{r}$ is the gradient of $\mathbf{r}$ (keeping $\boldsymbol{\lambda}$ fixed) and ∇_λ is used to denote the gradient operator with respect to $\boldsymbol{\lambda}$. On using Eq. (3.12) and noting that the Hessian of $\ln Z$ is $\mathbf{H} = \nabla_\lambda \mathbf{r}$, yields

$$\nabla\boldsymbol{\lambda} = -\mathbf{H}^{-1}(\mathbf{A} - \mathbf{I}), \quad \mathbf{H} = \sum_{b=1}^{n} \phi_b \tilde{\boldsymbol{x}}_b \otimes \tilde{\boldsymbol{x}}_b, \quad \mathbf{A} = \sum_{b=1}^{n} \phi_b \tilde{\boldsymbol{x}}_b \otimes \frac{\nabla w_b}{w_b},$$

and therefore ∇f_a in Eq. (3.14) becomes

$$\nabla f_a = \frac{\nabla w_a}{w_a} + \boldsymbol{\lambda} + \tilde{\boldsymbol{x}}_a \cdot \left[(\mathbf{H})^{-1} - (\mathbf{H})^{-1} \cdot \mathbf{A} \right]. \tag{3.15}$$

Using the above expression for ∇f_a in Eq. (3.13), the gradient of ϕ_a is

$$\nabla\phi_a = \phi_a \left\{ \tilde{\boldsymbol{x}}_a \cdot \left[(\mathbf{H})^{-1} - (\mathbf{H})^{-1} \cdot \mathbf{A} \right] + \frac{\nabla w_a}{w_a} - \sum_{b=1}^{n} \phi_b \frac{\nabla w_b}{w_b} \right\}. \tag{3.16}$$

For the numerical results in this dissertation, the quartic prior weight function of Eq. (3.7) is used and the radius of support, ρ_a, is taken as 0.9 times the distance to the fifth nearest neighbor.

The advantages of using maximum-entropy basis functions are revealed in Figure 3.2. Quartic weight functions, max-ent basis functions and moving least squares (MLS) basis functions are depicted on a unit square covered by a 3×3 nodal grid. For this example, the support size of the nodal weight function is taken as 1.25 times the distance to the fifth nearest neighbor. It is evident from Fig. 3.2f that the interior MLS basis function is not zero on the boundary in contrast to the max-ent basis function (Fig. 3.2d), which is zero on the boundary of the domain. Furthermore, boundary basis functions using maximum-entropy are interpolatory (Fig. 3.2c), whereas MLS basis functions are not (Fig. 3.2e). Due to these properties of max-ent basis functions, the imposition of essential boundary conditions in max-ent meshfree methods is performed as in finite element methods.

One-dimensional max-ent shape functions, using a quartic weight function, are illustrated in Figure 3.3b for comparison with the MLS shape functions previously shown in Figure 3.1b. As is seen in the 2D case the 1D max-ent shape functions are zero on the boundary whereas the MLS shape functions are not.

3.3 Consistency requirements

During numerical implementation it is useful to verify that the basis functions constructed are correct. As indicated by Belytschko et al. [16], if basis functions are constructed properly they satisfy certain consistency conditions. Furthermore, it is best if the conditions are satisfied to machine precision during numerical computations. The consistency conditions, for the case of two-dimensions, are as follows for arbitrary evaluation point $\boldsymbol{x}$:

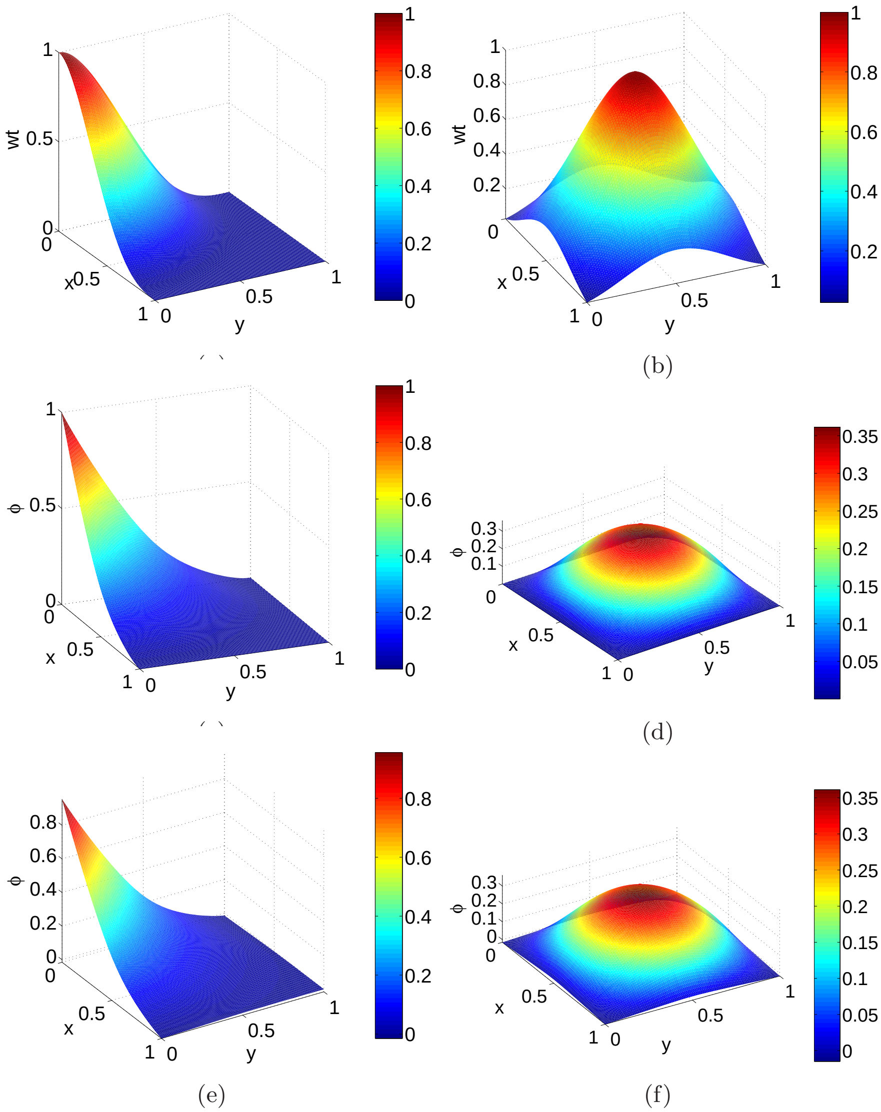


Figure 3.2: Max-ent versus MLS basis functions on unit square (3 x 3 grid). Quartic weight, max-ent and MLS basis functions for corner node in (a),(c),(e) and for center node in (b),(d),(f).

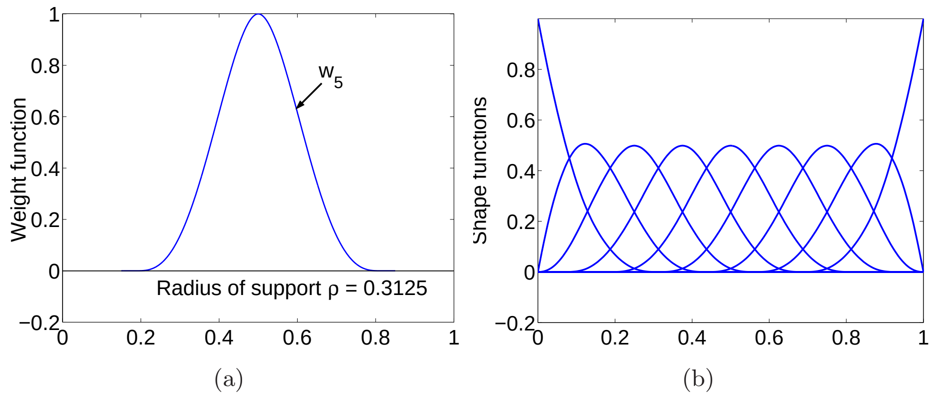


Figure 3.3: Max-ent construction (9 equi-spaced nodes): (a) weight function, (b) shape functions.

1. Partition of unity

$$\sum_{a}^{n} \phi_a(\boldsymbol{x}) = 1 \tag{3.17}$$

2. Linear consistency

$$\sum_{a}^{n} \phi_a(\boldsymbol{x}) x_a = x, \quad \sum_{a}^{n} \phi_a(\boldsymbol{x}) y_a = y \tag{3.18}$$

3. Partition of nullity

$$\sum_{a}^{n} \phi_{a,x}(\boldsymbol{x}) = 0, \quad \sum_{a}^{n} \phi_{a,y}(\boldsymbol{x}) = 0 \tag{3.19}$$

4. Derivative consistency

$$\sum_{a}^{n} \phi_{a,x}(\boldsymbol{x}) x_a = 1, \tag{3.20}$$

$$\sum_{a}^{n} \phi_{a,y}(\boldsymbol{x}) x_a = 0, \tag{3.21}$$

$$\sum_{a}^{n} \phi_{a,x}(\boldsymbol{x}) y_a = 0, \tag{3.22}$$

$$\sum_{a}^{n} \phi_{a,y}(\boldsymbol{x}) y_a = 1 \tag{3.23}$$

Chapter 4

Analysis of Plane Frames

4.1 Introduction

This chapter is an initial attempt to establish a new paradigm in structural engineering computation that offers a novel approach to analyzing structural engineering problems. With the goal of designing and protecting the civil infrastructure from unconventional loads, there arises the need to explore and develop new tools to analyze and predict the performance of structures. Great strides have been achieved in the exploration of meshfree technology in metal forming and crashworthiness simulations, but its application in structural engineering has yet to be initiated in a decisive manner. In this chapter a preliminary effort is presented to develop a framework that allows meshfree methodology to be embedded into a finite element-based formulation (or vice-versa) and thereby enabling the simulation of large-deformation structural response to complex loads. However, prior to embarking on the ultimate challenge of tackling large-deformation structural analysis that enables modeling of complex phenomena such as fracture and separation, it is essential to demonstrate the feasibility of the method by extending well-established theories in meshfree methodology to known concepts in computational structural analysis. The following phases are envisioned to accomplish the overall goals of this research endeavor: (i) development of a meshfree methodology for a class of structural elements and validation of the approach for nonlinear problems; (ii) extension of the developed methodology to incorporate co-rotational transformations; and (iii) incorporation of features to model material damage, separation, etc. This chapter addresses only the first step in this larger effort.

Therefore, with the eventual goal of investigating the feasibility of utilizing meshfree methods in such applications, a blended finite element and meshfree Galerkin method is formulated for nonlinear analysis of planar frames. Frame bending is modeled as a 2D continuum problem under plane stress conditions. This was considered more suitable than formulating a 1D beam (as employed by Atluri et al. [6], Donning and Liu [32], and Suetake [83]) because MLS shape functions would need to have cubic consistency in order to approximate both the displacement and rotation deformation fields. This causes increased difficulties in the meshfree formulation when trying to enforce displacement and slope boundary conditions. Furthermore, higher-order derivatives of the shape functions are required when solving the typical fourth-order differential equation necessary to model beam bending. Therefore, the plane stress approximation was considered more suitable for the proposed formulation and future research objectives. Small strain $J2$ elasto-plasticity is used to characterize material behavior and a stabilized nodal integration scheme is employed to obtain the discrete equations. An approach to model general sections with non-uniform thickness is developed, though the particular case of steel frames composed of wide flange sections is investigated in this dissertation. The proposed analytical scheme is applied to several examples involving beam and frame subassemblies undergoing post-elastic behavior. Results of numerical simulations are compared with analytical solutions, FE simulations and available experimental data to validate the proposed formulation.

4.2 Integrating the weak form

The variational (weak) form arises by taking the first variation of the potential energy and setting it to zero. Using the strain-displacement relation ($\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{d}$, where in a meshfree context the $\boldsymbol{d}$ values are *nodal coefficients*, and not nodal displacements) and the displacement approximation Eq. (3.1) in the weak form leads to

$$\boldsymbol{f}^{ext} - \boldsymbol{f}^{int} = \boldsymbol{0}, \tag{4.1a}$$

$$\boldsymbol{f}^{ext} = \int_S \boldsymbol{\phi}^T \bar{\boldsymbol{t}}\, dS, \quad \boldsymbol{f}^{int} = \int_V \boldsymbol{B}^T \boldsymbol{\sigma}\, dV, \tag{4.1b}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress and $\bar{\boldsymbol{t}}$ is the prescribed traction vector. For a linear elastic material, constitutive relations ($\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} = \mathbf{C}\boldsymbol{B}\boldsymbol{d}$) are substituted in Eq. (4.1b) to give

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}^{ext}, \tag{4.2}$$

where the stiffness matrix is

$$\boldsymbol{K} = \int_V \boldsymbol{B}^T \mathbf{C} \boldsymbol{B}\, dV. \tag{4.3}$$

For plane stress, the elastic modulus matrix, $\mathbf{C}$, is

$$\mathbf{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{4.4}$$

where E is the modulus of elasticity and ν is Poisson's ratio.

In an effort to depart from using elements for the purpose of numerical integration, a node-based integration technique is used to compute $\boldsymbol{K}$ in Eq. (4.3). For node-based

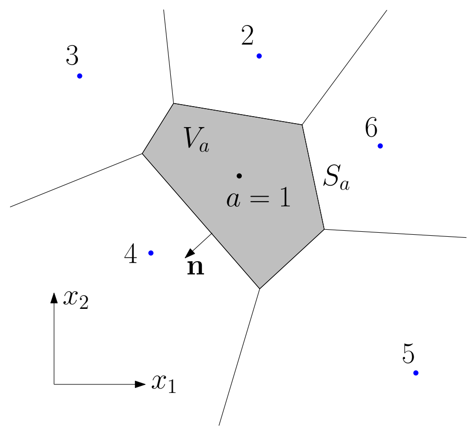


Figure 4.1: Voronoi cell of node a.

integration, a background geometric structure, such as a Voronoi diagram, is still required. This geometric structure is preferable since it is node-based rather than element-based and no Jacobian is required. A further advantage of nodal integration is that state variables, such as material properties, are associated with nodes rather than elements. The nodal integration procedure adopted here closely follows the integration scheme proposed by Chen et al. [20].

Consider the Voronoi cell domain V_a and boundary of segments S_a enclosing node a as shown in Figure 4.1. Over the domain V_a, the components of the smoothed strain tensor (finite volume averaging) are

$$\varepsilon_{ij}(\boldsymbol{x}_a) = \frac{1}{2A_a}\int_{V_a}(u_{i,j} + u_{j,i})\,dV = \frac{1}{2A_a}\int_{S_a}(u_i n_j + u_j n_i)\,dS, \tag{4.5}$$

where the last expression is found by using the divergence theorem, A_a is the Voronoi cell area associated with node a, and n_i is the ith component of a unit vector normal to the Voronoi cell boundary S_a.

Now, similar to FEM, $\boldsymbol{\varepsilon}$ of Eq. (4.5) is written as a strain-displacement relation. Using

Eq. (3.1) in Eq. (4.5) and defining some new variables the strain-displacement relations are

$$\boldsymbol{\varepsilon}(\boldsymbol{x}_a) = \sum_{b=1}^{6} \boldsymbol{B}_b(\boldsymbol{x}_a)\boldsymbol{d}_b = [\boldsymbol{B}_1\ \boldsymbol{B}_2 \cdots \boldsymbol{B}_6] \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_6 \end{bmatrix} \equiv \boldsymbol{B}(\boldsymbol{x}_a)\boldsymbol{d}, \tag{4.6}$$

where the index b ranges over the nodes whose associated shape function supports cover any vertex of the Voronoi cell a (i.e., nodes 1 to 6 for the example of Figure 4.1) and the following definitions apply:

$$\boldsymbol{\varepsilon} = [\varepsilon_{11}\ \varepsilon_{22}\ 2\varepsilon_{12}]^T \quad \text{and} \quad \boldsymbol{d}_a = [d_{a1}\ d_{a2}]^T \tag{4.7}$$

$$\boldsymbol{B}_b(\boldsymbol{x}_a) = \begin{bmatrix} b_{b1}(\boldsymbol{x}_a) & 0 \\ 0 & b_{b2}(\boldsymbol{x}_a) \\ b_{b2}(\boldsymbol{x}_a) & b_{b1}(\boldsymbol{x}_a) \end{bmatrix} \tag{4.8}$$

$$b_{bi}(\boldsymbol{x}_a) = \frac{1}{A_a} \int_{S_a} \phi_b(\boldsymbol{x}) n_i(\boldsymbol{x})\, dS. \tag{4.9}$$

On using the strain-displacement relation Eq. (4.6) in Eq. (4.3) gives $\boldsymbol{K}$ associated with node a as

$$\boldsymbol{K}(\boldsymbol{x}_a) = \boldsymbol{B}^T(\boldsymbol{x}_a)\mathbf{C}\boldsymbol{B}(\boldsymbol{x}_a)A_a t. \tag{4.10}$$

The thickness of the two dimensional domain, t, is generally taken as unity. The external force vector $\boldsymbol{f}^{ext}$ of Eq. (4.1b) is found similarly (see Chen et al. [20]).

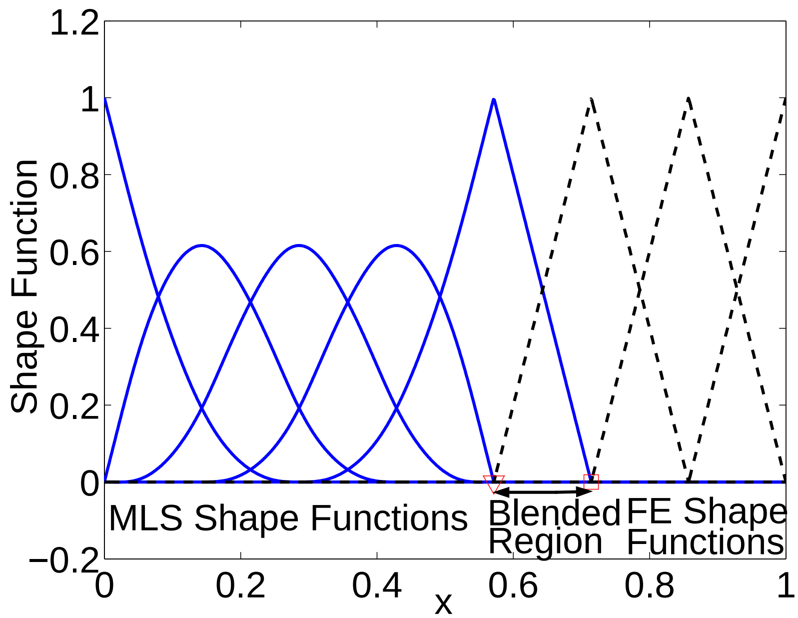


Figure 4.2: Blending of linear MLS and FE shape functions.

4.3 Enforcement of essential boundary conditions

In general MLS shape functions do not possess the Kronecker-delta property. Hence it is difficult to enforce essential boundary conditions when using MLS shape functions. To overcome this problem a variety of techniques have been devised to enforce essential boundary conditions such as Lagrange multiplier method, penalty method, Nitsche's method and continuous blending method (Fernández-Méndez and Huerta [35]). In this dissertation, continuous blending is used because it allows the MLS shape functions to blend into FE shape function regions. Hence, the MLS shape functions are used everywhere except at nodes where essential boundary conditions need to be enforced. At such nodes FE shape functions are used, and enforcement of essential boundary conditions is imposed on the finite element nodes in the standard way.

Continuous blending as proposed by Huerta and Fernández-Méndez [40] is accomplished by recognizing three distinct regions possible in the discretized domain when the two types of shape functions are used. These regions are (i) MLS regions, (ii) blended regions (transition between MLS and FE shape functions) and (iii) FE regions. For region (i) the MLS shape functions are as given in Eq. (3.6). In this dissertation, a linear polynomial basis is used to construct the MLS shape functions. The finite element shape functions perform best in the method of continuous blending if they are also linear as indicated in Huerta and Fernández-Méndez [40]. Therefore in region (ii) the MLS shape functions are blended into linear quadrilateral finite element shape functions. Lastly, in region (iii) the transition is complete and typical linear quadrilateral finite elements are solely used to construct the approximate solution.

The meshfree approximation in a blended region is represented as

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{a=1}^{n_{MLS}} \tilde{\phi}_a(\boldsymbol{x})\boldsymbol{d}_a + \sum_{b=1}^{n_{FE}} N_b(\boldsymbol{x})\boldsymbol{u}_b, \tag{4.11}$$

where the tilde is used to denote the blended approximation. If a node needs enforcement of an essential boundary condition there is a two dimensional linear finite element shape function, N_b, associated with the node. Note that a blended region does not have a complete set of finite element shape functions. Hence, in Eq. (4.11) the sum over a is for all MLS shape functions that are nonzero in the given blended region and the sum over b is for all nonzero FE shape functions. The MLS shape functions in a blended region are constructed the same as explained previously by enforcing the consistency condition:

$$\mathbf{p}(\boldsymbol{x}) = \sum_{a=1}^{n_{MLS}} \tilde{\phi}_a(\boldsymbol{x})\mathbf{p}(\boldsymbol{x}_a) + \sum_{b=1}^{n_{FE}} N_b(\boldsymbol{x})\mathbf{p}(\boldsymbol{x}_b). \tag{4.12}$$

Equation (4.12) states that in the blended region the combined FE and MLS approximation Eq. (4.11) is consistent with the polynomial that it is trying to approximate. Then, following a procedure similar to the MLS shape function derivation, the MLS approximant in the blended region is

$$\tilde{\phi}_a(\boldsymbol{x}) = \phi_a(\boldsymbol{x}) - \mathbf{p}^T(\boldsymbol{x}_a)\mathbf{A}^{-1}\left(\sum_{b=1}^{n_{FE}} N_b(\boldsymbol{x})\mathbf{p}(\boldsymbol{x}_b)\right)w(\boldsymbol{x}_a). \tag{4.13}$$

The first term on the right hand side of Eq. (4.13) is the MLS shape function of node a in the meshfree region. The second term on the right hand side of Eq. (4.13) is the correction to the MLS shape function of node a if it is nonzero in the blended region. An example of one-dimensional linear MLS shape functions blended into linear finite elements is shown in Figure 4.2.

4.4 Numerical implementation

Several issues in the numerical implementation require further attention. First, it is shown how to calculate the individual components of the smoothed strain-displacement matrices. Second, nodal integration is unstable and requires some form of numerical stabilization, which is addressed. Here, the terms stable and stabilization are not a mathematically precise usage; however, they are often used in this context in the meshfree literature.

4.4.1 Numerical evaluation of strain-displacement matrix components

To carry out the integration, by numerically evaluating the components Eq. (4.9) of the strain-displacement matrix, a two-node trapezoidal rule is employed. As indicated in the example of Figure 4.3(a), $\boldsymbol{x}_a^M$ and $\boldsymbol{x}_a^{M+1}$ are the end nodes of segment S_a^M. The length of the segment is ℓ_a^M and surface normal of the segment is $\boldsymbol{n}_a^M$. Using these definitions Eq. (4.9) is rewritten as a summation over the number of Voronoi cell segments, N_s,

$$b_{bi}(\boldsymbol{x}_a) = \frac{1}{A_a}\sum_{M=1}^{N_s}\left[\phi_b(\boldsymbol{x}_a^M)n_{ai}^M\frac{\ell_a^M}{2} + \phi_b(\boldsymbol{x}_a^{M+1})n_{ai}^M\frac{\ell_a^M}{2}\right]. \tag{4.14}$$

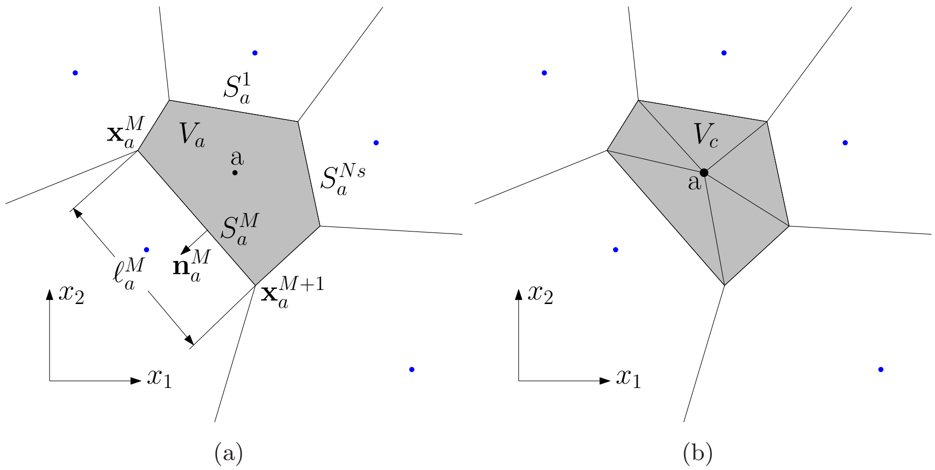


Figure 4.3: Node a: (a) integration over Voronoi cell; (b) triangular subcells.

When the last segment in the summation is reached define $M+1 = N_s + 1 \equiv 1$. Next, noting that Eq. (4.14) only involves evaluation of $\phi_b n_{ai}$ at the vertices of the Voronoi cell for node a, Eq. (4.14) is now written as

$$b_{bi}(\boldsymbol{x}_a) = \frac{1}{A_a} \sum_{M=1}^{N_s} \left[\frac{1}{2} (n_{ai}^M \ell_a^M + n_{ai}^{M+1} \ell_a^{M+1}) \phi_b(\boldsymbol{x}_a^{M+1}) \right]. \tag{4.15}$$

This last equation involves no derivatives of the MLS shape functions. The technique of nodal integration is used in linear problems in Chen et al. [20] and also in nonlinear problems involving large displacements in Chen et al. [21].

4.4.2 Stabilization of stiffness matrix

Nodal integration instabilities are often manifested by hourglass modes in the calculated deflected shape, by spurious low-energy modes in an eigen analysis and by locking in nearly or totally incompressible materials. Hence, some form of stabilization is needed for the stiffness matrix given in Eq. (4.10). Puso et al. [72] proposed the following stabilization scheme:

$$\boldsymbol{K}^s(\boldsymbol{x}_a) = \boldsymbol{K}(\boldsymbol{x}_a) + \left[\alpha_s \sum_{c\in T_a} (\boldsymbol{B}(\boldsymbol{x}_a) - \boldsymbol{B}^c(\boldsymbol{x}_a))^T \mathbf{C}_s (\boldsymbol{B}(\boldsymbol{x}_a) - \boldsymbol{B}^c(\boldsymbol{x}_a)) A_c t\right], \tag{4.16}$$

where $\boldsymbol{K}^s(\boldsymbol{x}_a)$ is the stabilized matrix, $\alpha_s = 1.0$ is the stabilization factor and $\mathbf{C}_s$ is the stabilization modulus matrix. The first term on the right hand side of Eq. (4.16) is equivalent to Eq. (4.10) and the second (stabilization) term in brackets is a summation over the set of triangular subcells, T_a, for Voronoi cell a (see Figure 4.3(b)). Over each triangular subcell c the $\boldsymbol{B}^c$ matrix is constructed in the same way that $\boldsymbol{B}$ matrices are constructed over a Voronoi cell.

When constructing $\mathbf{C}_s$ for plastic materials with Lamé parameters μ and λ, the recommendation of Puso et al. [72] is adopted such that the effective moduli are

$$\tilde{\mu} = \bar{H}/2 \quad \text{and} \quad \tilde{\lambda} = \max(\lambda, 12.5\bar{H}), \tag{4.17}$$

where for linear hardening, $\bar{H}$ is the hardening modulus and for exponential hardening, $\bar{H}$ is taken as the slope of the tangent to the exponential hardening curve at zero plastic strain. The effective elastic modulus $\tilde{E}$ and Poisson's ratio $\tilde{\nu}$ in terms of $\tilde{\mu}$ and $\tilde{\lambda}$ are given by

$$\tilde{E} = \frac{\tilde{\mu}(3\tilde{\lambda} + 2\tilde{\mu})}{\tilde{\lambda} + \tilde{\mu}} \quad \text{and} \quad \tilde{\nu} = \frac{\tilde{\lambda}}{2(\tilde{\lambda} + \tilde{\mu})}. \tag{4.18}$$

4.4.3 Summary of discrete equations

In general Eq. (4.1a) is nonlinear since the unknown stress field at time $n+1$ is a nonlinear function of strain. Hence, following an approach similar to that presented in finite element monographs such as Gosz [37], linearization of $\boldsymbol{\sigma}$ gives

$$\boldsymbol{\sigma}_{n+1} \approx \boldsymbol{\sigma}_n + \mathbf{C}_n^{ep} \Delta \boldsymbol{\varepsilon}_n = \boldsymbol{\sigma}_n + \mathbf{C}_n^{ep} \boldsymbol{B} \Delta \boldsymbol{d}_n, \tag{4.19}$$

where $\Delta \boldsymbol{d}_n = \boldsymbol{d}_{n+1} - \boldsymbol{d}_n$, $\mathbf{C}_n^{ep}$ is the plane stress elasto-plastic tangent modulus matrix (Simo and Taylor [80], see also Appendix E) and use has been made of the strain-displacement relations. Substitution of Eq. (4.19) into Eq. (4.1a) gives

$$\int_V \boldsymbol{B}^T \mathbf{C}_n^{ep} \boldsymbol{B} \, dV \Delta \boldsymbol{d}_n = \left\{ \int_S \boldsymbol{\phi}^T \bar{\boldsymbol{t}} \, dS \right\}_{n+1} - \int_V \boldsymbol{B}^T \boldsymbol{\sigma}_n \, dV \quad \Rightarrow \quad \boldsymbol{K}_n^t \Delta \boldsymbol{d}_n = \boldsymbol{f}_{n+1}^{ext} - \boldsymbol{f}_n^{int}, \tag{4.20}$$

where $\boldsymbol{K}_n^t$ is the tangent stiffness matrix. A Newton-Raphson scheme is used to iterate the linearized (see Gosz [37]) system of Eqs. (4.20) until convergence is achieved. The iterated equation is written as

$$\boldsymbol{K}_{n+1}^{t(\nu)} \Delta \boldsymbol{d}_n^{(\nu)} = \boldsymbol{f}_{n+1}^{ext} - \boldsymbol{f}_{n+1}^{int(\nu)}, \tag{4.21}$$

where ν is the iteration counter. It is understood that when the iteration counter is zero the matrices and vectors are evaluated at time n, i.e., $\boldsymbol{f}_{n+1}^{int(0)} = \boldsymbol{f}_n^{int}$, etc.

On the basis of the preceding developments, the discrete equations are obtained as follows:

$$\boldsymbol{K}(\boldsymbol{x}_a) = \boldsymbol{B}^T(\boldsymbol{x}_a) \tilde{\mathbf{C}} \boldsymbol{B}(\boldsymbol{x}_a) A_a t \tag{4.22a}$$

$$\boldsymbol{K}^s(\boldsymbol{x}_a) = \boldsymbol{K}(\boldsymbol{x}_a) + \left[\alpha_s \sum_{c \in T_a} (\boldsymbol{B}(\boldsymbol{x}_a) - \boldsymbol{B}^c(\boldsymbol{x}_a))^T \mathbf{C}_s (\boldsymbol{B}(\boldsymbol{x}_a) - \boldsymbol{B}^c(\boldsymbol{x}_a)) A_c t \right] \tag{4.22b}$$

$$\boldsymbol{f}_b^{ext} = \sum_{a=1}^{n_b} \phi_b(\boldsymbol{x}_a) \bar{\boldsymbol{t}}(\boldsymbol{x}_a) S_a. \tag{4.22c}$$

In the above equations, for nonlinear problems, such as elasto-plasticity, $\tilde{\mathbf{C}} = \mathbf{C}^{ep}$ and

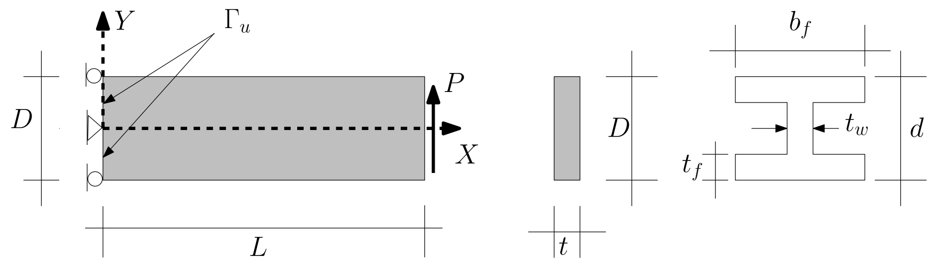
$\boldsymbol{K}^s(\boldsymbol{x}_a)$ is the stabilized tangent stiffness matrix to be used in Eq. (4.21). For linear problems $\tilde{\mathbf{C}}$ is replaced with Eq. (4.4) and the stabilized stiffness matrix, $\boldsymbol{K}^s(\boldsymbol{x}_a)$, replaces $\boldsymbol{K}$ in Eq. (4.2). In Eq. (4.22c) S_a is the length along the boundary of the Voronoi cell of node a along which the traction $\bar{\boldsymbol{t}}$ acts, and n_b is the number of boundary points. Once the discrete equations are solved for the nodal coefficients, $\boldsymbol{d}$, the displacements at each node are found by using Eq. (3.1). The strains at each node a are found by using Eq. (4.6) and stresses are found by using the appropriate constitutive relations.

4.5 Formulation for sections with non-uniform thickness

In order to allow for non-uniform thickness Eq. (4.22a) is modified to allow a unique thickness, t_a, for each node a:

$$\boldsymbol{K}(\boldsymbol{x}_a) = \boldsymbol{B}^T(\boldsymbol{x}_a)\tilde{\mathbf{C}}\boldsymbol{B}(\boldsymbol{x}_a)A_a t_a. \tag{4.23}$$

A similar modification is required for Eq. (4.22b). By setting the thickness for different regions of the 2D continuum it is possible to model a variety of common beam cross-sections. For example, I-beams and channels in the case of steel cross-sections or T-beams and I-shaped girders in the case of concrete. Of course the thicknesses are set to obtain a moment of inertia which matches the beam being modeled. The validation problems included below are for beams of I-shaped cross-section. Therefore, in Appendix D an example is provided which illustrates the process of deriving the thickness of a section associated with a particular meshfree node to obtain the correct moment of inertia for an I-beam.



(a) Cantilever Beam (b) Rectangular Cross-section (c) I-beam

Figure 4.4: Cantilever beam notation.

4.6 Validation of methodology

The linear and nonlinear response of several realistic frame subassemblies are evaluated using the proposed blended FEM and meshfree method. For all example problems only constrained nodes have FE shape functions for enforcement of essential boundary conditions by the blending method. The remaining domain is modeled with MLS shape functions. The results are compared to analytical solutions in the case of the cantilever beam and to experimental data for a frame corner connection and a portal frame. Also included are comparisons with simulations using one-dimensional fiber-section beam elements since they are commonly employed in nonlinear frame analysis. The open-source structural analysis software OpenSees, Mazzoni et al. [65], is used for both the 1D fiber beam (*dispBeamColumn* element) simulations and the FE simulations with enhanced strain quadrilateral elements.

4.6.1 Cantilever beam

First, the results for an I-beam cantilever using a linear elastic material are presented. In Table 1, with grid refinement, the normalized tip displacement and bending stress values asymptotically approach 1.0 where $\delta_{\text{theor.}} = 0.0308$ in. and $\sigma_{\text{theor.}} = 25.0$ ksi. For these results the variables used are $E = 29,000$ ksi, $\nu = 0.3$, $P = 5$ kips, $L = 10$ in., $I_{xx} = 2.0$ in.^{4},

$d = 2$ in., $t_w = 1$ in. with t_f and b_f dependent on the grid as explained in Appendix D.

Grid	$\delta/\delta_{\text{theor.}}$ (in)	$\sigma_{xx}/\sigma_{\text{theor.}}$ (ksi)
11×3	1.045	0.77
21×5	1.025	0.89
31×7	1.016	0.94
41×9	1.012	0.95
51×11	1.010	0.96
61×13	1.009	0.97

Table 4.1: Cantilever I-beam tip displacement and maximum bending stress.

Second, the results for an I-beam cantilever using an elasto-plastic material model are shown in Figure 4.5. This analysis is performed with small strain plane stress $J2$ elasto-plasticity as outlined in Simo and Taylor [80]. The solution procedure uses Newton-Raphson iterations at the global level to enforce equilibrium between internal and external forces (see Gosz [37]), and at the constitutive level an implicit integration scheme with radial return is employed (Simo and Hughes [78], see Appendix E for details). In Figure 4.5b, the load versus displacement response is compared to an analytical solution for an elasto-plastic cantilever based on Euler-Bernoulli beam theory with an elastic shear deformation term included similar to Eq. (D.6). For the results of Figure 4.5b, $I_{xx} = 1.313$ in.^{4}, however, all remaining geometry and material properties are the same as the linear analysis. In addition, the hardening modulus is $\bar{H} = 500$ ksi, the yield stress is 36 ksi, and the maximum applied load is 8 kips. The I-beam is modeled with a 51 by 11 grid of nodes (similar to 500 elements). Figure 4.5b also shows results for a 1D fibersection beam model with a discretization of 10 finite beam elements and a 2D continuum model using 500 enhanced strain quadrilateral finite elements.

The analytical solution (see Appendix B for the derivation) uses the bilinear model shown in Figure 4.5a and is developed independently of previous work. However, the initial steps to find the analytical solution are similar to Yu and Zhang [97] who set out

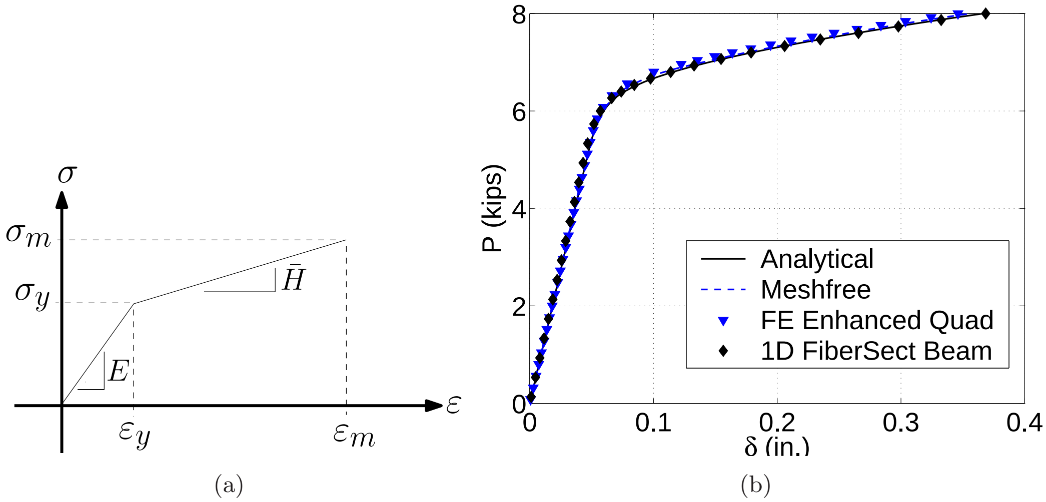


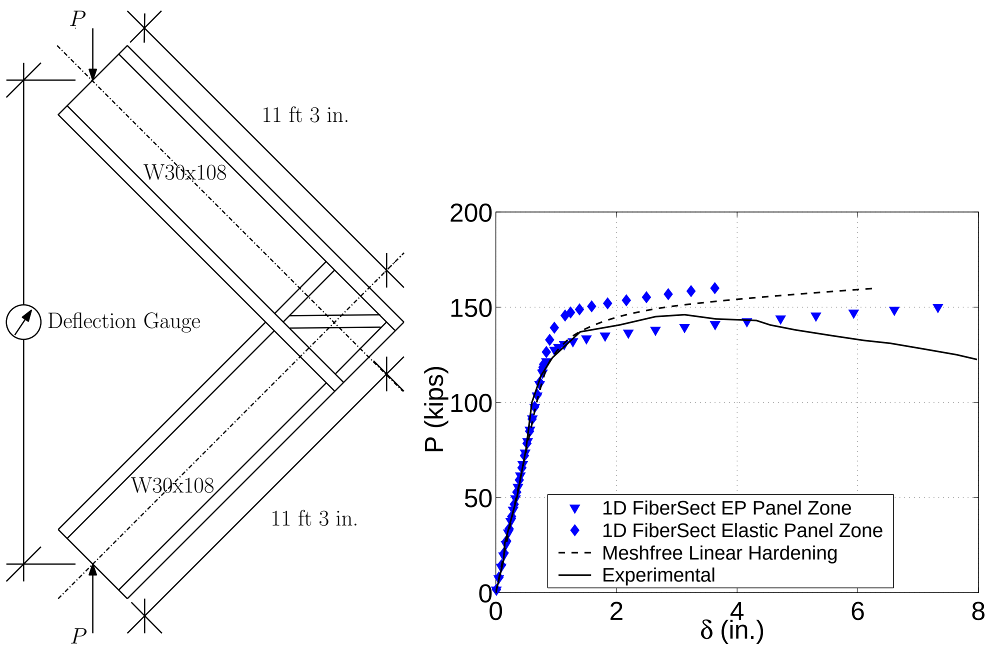
Figure 4.5: Inelastic cantilever I-beam: (a) bilinear model; (b) load versus displacement.

preliminary formulas for an elasto-plastic beam of rectangular cross-section with a linear hardening material. The interested reader is also referred to Phillips [71], where a method to express the curvature as a function of applied moment is presented for beams of various cross-sections, including a plastically deforming I-beam. Once curvature is known at every cross-section deflection is calculated using the Second Area Moment theorem.

The numerical solution slightly differs from the analytical solution for a variety of reasons. First, the discretization of the cantilever across the beam depth cannot exactly represent the bilinear stress profile. Second, the analytical solution assumes plane sections remain plane. However, the numerical solution is based on the elasticity solution which does not restrict plane sections to remain planar. Third, the exact displacement boundary conditions for an I-beam are not known. Hence, all nodes at the support are pinned. Last, the analytical solution is based on the bilinear material model within Euler-Bernoulli beam theory. This differs from the numerical solution where a $J2$ elasto-plastic material model in plane stress is assumed. Despite these differences, the agreement between the numerical and analytical results is excellent.

4.6.2 Frame corner connection

In the second example, the nonlinear response of a frame corner connection tested to failure by Beedle and Christopher [13] is investigated and the computed response is compared with experimental results. The frame connection is made of W30x108 members and stiffeners as shown in the test setup of Figure 4.6a. In the numerical model larger thicknesses are specified along straight and diagonal stiffener lines to properly simulate the effect of the stiffeners on the response of the connection. The resulting experimental versus simulated load displacement results are shown in Figure 4.6b. Load displacement results are also shown for 1D fibersection beam models with a discretization of 22 finite beam elements. For the beam elements, in one case the panel zone elements were allowed to have an elasto-plastic response, whereas the other case was forced to have an elastic panel zone. The numerical results vary from the experimental results for several reasons. First, Beedle and Christopher [13] do not provide the material properties for the corner connection material. Theoretical predictions for the load displacement curve based on an elastic perfectly plastic material provided by Beedle and Christopher [13] indicate their assumed yield stress value of 36 ksi. However, following the recommendation of Johnston et al. [44], for a more accurate plastic analysis, a yield plateau value of 33 ksi is used in the material model herein. Secondly, the hardening used for the numerical results is based on an assumed ultimate value of 55 ksi. Beedle and Christopher [13] mention that the frame connection did develop its full plastic moment but failed by flange local buckling and that this accounts for some hardening followed by softening as shown by the experimental curve. Hence, it is not reasonable to expect that the estimated hardening behavior provided in the numerical results (which does not consider flange local buckling) will exactly match the experimental results. Despite these differences, the numerical results are in general agreement with the experimental results.



(a) (b)

Figure 4.6: Frame corner connection: (a) test set up; (b) load versus displacement response.

Figure 4.7 shows an example of the final (160 kip load) stresses for the corner connection. The σ_{xx} stresses are oriented along the axis of the top W30x108 beam. It is evident from the stress plot that a plastic hinge has formed near the corner connection and that the location of the neutral axis has shifted from the beam centerline toward the tension flange. This shift is due to the combined stress state of bending and axial stresses. Each of these observations are expected and lend confidence to the validity of the results. Lastly, it is found that grid refinement from 357 nodes to 621 nodes does not change the simulation results significantly. Figure 4.8b shows the magnified final (160 kip load) deflected shape for the corner connection. Deflection results without stabilization (Figure 4.8a) exhibit hourglass modes in the deformed shape.

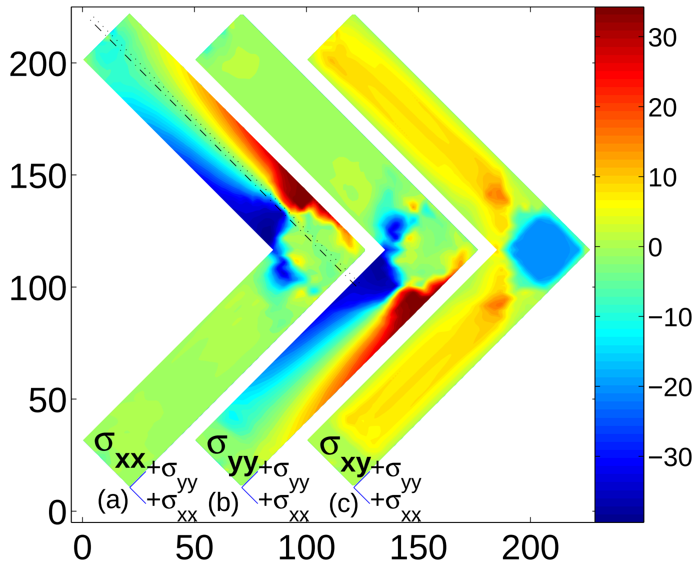


Figure 4.7: Frame corner connection stresses (ksi): (a) σ_{xx}; (b) σ_{yy}; and (c) σ_{xy}.

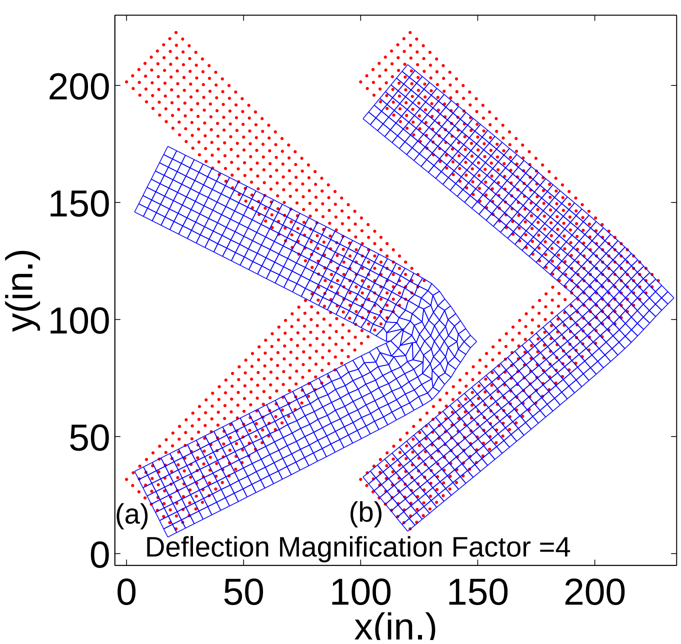


Figure 4.8: Frame corner connection displacements: (a) without stabilization; (b) with stabilization.

4.6.3 Portal frame

Consider now the response of a portal frame loaded by equal vertical and lateral forces. In this case an 8 inch deep I-beam with 4 inch flanges tested by Baker and Roderick [9] is utilized in the simulation. Based on the specified web thickness of 1/4 inch, a given plastic moment capacity of 576 k-in and given upper yield stress of 36 ksi provided in Baker and Roderick [9], the flange thickness is calculated as 0.422 inches. These properties give $Z_x =$ 16 in.^{3} and $I_{xx} = 56.18$ in.^{4}, which are used in the frame analysis. The frame dimensions, support conditions and loading are shown in Figure 4.9a. Linear hardening is assumed with an assumed ultimate steel stress of 55 ksi. Although the given material was assumed to be A36 steel, the lower yield plateau stress of 33 ksi is used. Load deflection results for a numerical model with 2023 nodes are compared in Figure 4.9b to the experimental results provided by Baker and Roderick [9]. The load deflection results are also shown for a 1D fibersection beam model with a discretization of 5 beam elements per column and 10 beam elements for the girder. The load deflection results are in good agreement with the experimental results. As the loading progresses hinges develop simultaneously in the beam and column near the top right corner joint of the frame. Following this condition, a constant moment results in the beam segment between the vertical load and the left column. A uniform elasto-plastic stress profile (see Figure 4.9d) progresses along the beam segment until the frame resists no more load. These observations are consistent with those presented in Baker and Roderick [9], and the numerical results given here closely match the expected behavior.

Inevitably there are differences between the numerical and experimental results. In addition to this there is always uncertainty in the true material properties of the tested frame as well as uniformity of the quality of welded connections. In fact, Baker and Roderick [9] state only an assumed yield of 36 ksi and do not provide an estimate for the ultimate

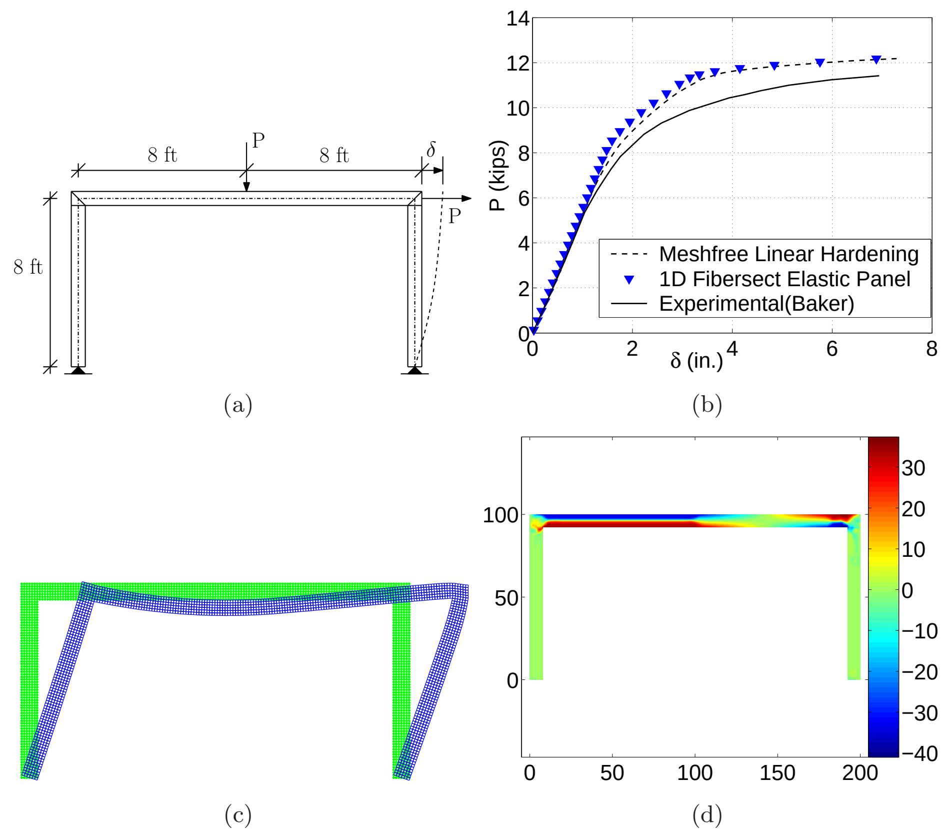


Figure 4.9: Frame example: (a) test set up; (b) load versus displacement plot; (c) frame deflected shape; and (d) stresses σ_{xx} (ksi).

stress. Having said this, it is noted that the most important value seems to be the yield stress used in the numerical study. Adjusting ultimate stress and other hardening variables has relatively little affect on the numerical curve shown in Figure 4.9b. However, using a yield stress of 33 ksi gives numerical results similar to the experiment.

4.7 Conclusions

The capabilities of the element-free Galerkin method have been extended in this chapter to the inelastic analysis of steel frames. A blended finite element and meshfree method was developed for beam bending approximated as a plane stress problem. The treatment of sections with non-uniform thickness is presented though the particular case of wide-flange sections is considered in the sample simulations. The methodology was applied to solve a variety of frame subassemblies undergoing inelastic deformations. It is shown that the results from the numerical simulations match theory, experimental observations and other finite element based solutions with considerable accuracy. This work has established the feasibility of meshfree methods for the simulation of nonlinear frame response. However, as pointed out in the introduction, the successful application of the meshfree formulation to plane frame analysis is only a first necessary step toward the eventual goal of extending the technology to more complex problems. In this phase of work, only material nonlinearities in a small strain framework were considered. Incorporation of large displacements based on a co-rotational formulation is presented in the next chapter along with the use of maximum-entropy shape functions which allow easier enforcement of essential boundary conditions and numerical implementation.

Chapter 5

Meshfree Co-rotational Formulation

5.1 Introduction

Co-rotational formulations are commonly used in finite element formulations for the analysis of structures. Wempner [95] and Belytschko and Hsieh [15] pioneered the introduction of co-rotational formulations in finite element analysis. Such a formulation has many commonalities with the 'natural approach' of Argyris et al. [4]. The co-rotational formulation is very popular for beams and shell elements and it has been extended to include finite strains with continuum elements in a consistent formulation by Crisfield and Moita [28, 29]. One of the primary motivations of a co-rotational formulation is the ability to use linear elements in a non-linear context. Thus far, the co-rotational formulation has only been implemented using finite element shape functions. In this dissertation, meshfree basis functions are introduced within the framework of a co-rotational formulation for continua. It appears that this has not been previously introduced in the literature.

This chapter is part of the larger effort to advance collapse simulation technology for large-scale civil engineering structures. Collapse simulation is by its nature a problem that is highly nonlinear, subject to large displacements, rotations and inelastic material behavior.

While finite-element based simulations of structural collapse and failure have met with some success for limited applications [39, 41, 60], much of the effort using finite elements to simulate large displacements have been subject to difficulties due to mesh distortions that cause a need for remeshing, loss of accuracy, and at times unsuccessful completion of the simulation altogether. These difficulties are observed for both continuum elements as well as beam elements.

Significant work has gone into the development of beam elements for limit state analysis of large-scale engineering structures. However, as noted by Torkamani [90], these methods tend to have two principal deficiencies: inaccurate descriptions of material nonlinearity, and an inability to properly capture large distortions across the length of the element. Fiber-based beam elements have been used to improve modeling of material nonlinearity [75, 86, 90] and Lagrangian or co-rotational formulations are employed to include large deflections. Despite these advances, the ability to simulate collapse is still inadequate.

The objective of this chapter is to examine the ability of meshfree methods to alleviate some of these difficulties and explore the potential for an alternative approach to large-displacement analysis of structural systems.

On combining the advantages of meshfree methods with those of a co-rotational formulation, it is expected that the ability to simulate large displacements and large rotations can be better facilitated.

Furthermore, by using a continuum approach, material behavior is modeled more accurately through the cross-section of beam-type structural elements. As a first step toward the eventual goal of advancing collapse simulations, the present chapter focuses on two-dimensional continua in the presence of small strains with elastic and elasto-plastic material behavior.

The remainder of this chapter is organized as follows. In Section 5.2, the co-rotational formulation is derived to give (i) the relationship between global and local variables, (ii) the angle of rotation of a typical co-rotating coordinate system, and (iii) a variationally consistent tangent stiffness matrix. Including inelastic material behavior is also discussed followed by an algorithm for a co-rotational formulation in a meshfree setting. The section concludes with a brief discussion of numerical implementation details specific to a co-rotational formulation. Section 5.3 presents numerical examples for validation of the proposed formulation, which is followed by some concluding remarks in Section 5.4.

5.2 Co-rotational formulation

In general, the motion of a body is composed of rigid body translation, rigid body rotation and strain producing deformations. Consider a sufficiently small region $\Omega \subset \mathbb{R}^2$ of a body. To this small region attach a local coordinate frame that rotates and translates with the material points of the region. With respect to this local coordinate frame, the rigid body rotations and translations, of the small region's overall motion, are negligible and only local strain-producing deformations remain. This is the key idea behind a co-rotational formulation. It is the objective of a co-rotational formulation to perform a nonlinear analysis of a structure and determine the global displacement behavior as well as the stress and strain causing local deformations. Some of the advantages of a co-rotational formulation are as follows. First, for small strain/large rotation problems Mattiasson [64] indicates that the co-rotational formulation is more accurate and has better convergence properties than finite strain total Lagrangian or updated Lagrangian formulations. Second, co-rotational formulations satisfy the principle of material frame indifference (Belytschko and Bindermen [14]). As a result of the material frame invariance damage constitutive equations are not limited to isotropic elastic response (Masud *et al* [62]). Third, inelastic type constitutive equations take the same form as in the case of a small deformation theory since stresses and strain tensors are objective (W.K. Liu *et al* [58]). This greatly simplifies integration of inelastic constitutive equations. Lastly, geometric nonlinearities due to large displacements and rotations are taken into account without the requirement of a finite strain formulation and alternative stress definitions.

For a co-rotational formulation several key ingredients are necessary. These ingredients are: (i) the relationship between global and local variables, (ii) a method for determining the angle of rotation of a typical co-rotating coordinate system, and (iii) the expression for a variationally consistent tangent stiffness matrix. These ingredients are described within the following subsections, where for the sake of clarity and completeness, intermediate steps in the derivation are also indicated. The ensuing presentation closely follows Crisfield and Moita [28].

5.2.1 Relationship between global and local variables

Referring to Figure 5.1, the relationship between overall global deformations and the local strain producing deformations is illustrated. In Figure 5.1, node L and its neighboring nodes are shown. In general, there are n nodes (node L is included in the set of n nodes) to which a local co-rotating coordinate frame is associated (for finite elements the coordinate frame is usually attached to each element). For simplicity only four nodes are shown and the local co-rotating frame origin is placed at node L. In the reference configuration, the local co-rotating frame axes are parallel to the global axes. Due to displacement of the overall structure the n nodes translate, rotate and deform to some current configuration as

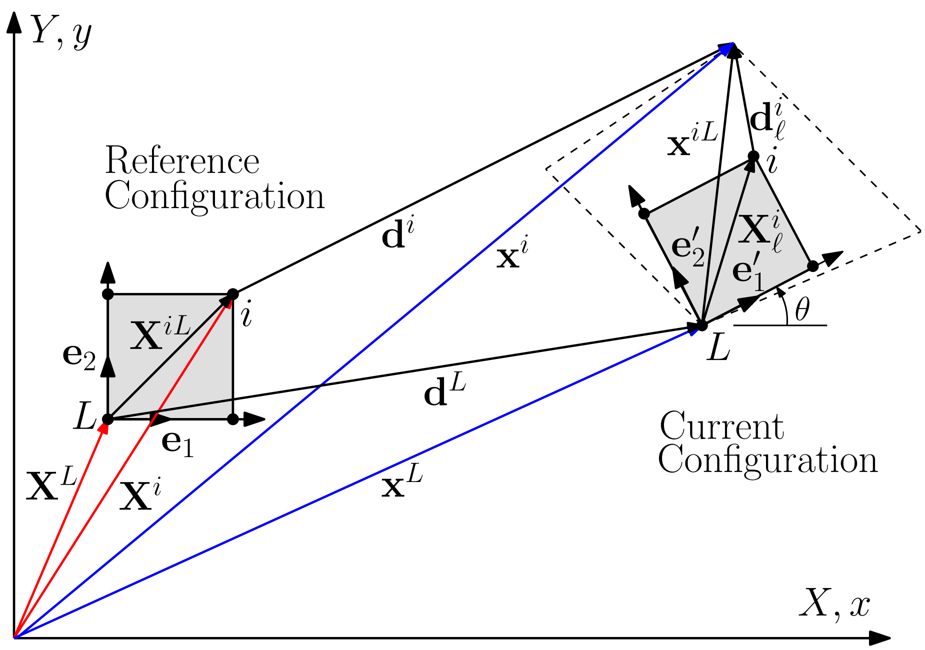


Figure 5.1: Reference and current configurations in co-rotational formulation.

shown.

From the figure the local nodal coefficients (nodal displacements for the case of finite elements) for node i in the local coordinate frame are expressed as

$$\mathbf{d}_\ell^i = \mathbf{Q}^T \mathbf{x}^{iL} - \mathbf{X}_\ell^i, \tag{5.1}$$

where a subscript ℓ is attached to vectors with components in local coordinates (with some exceptions such as stress, $\boldsymbol{\sigma}$, and strain-displacement matrices, $\mathbf{B}$, which are clearly understood to be in the local coordinates of a co-rotational formulation) and $\mathbf{x}^{iL} = \mathbf{x}^i - \mathbf{x}^L = \mathbf{X}^{iL} + \mathbf{d}^i - \mathbf{d}^L$ indicates the difference between the spatial coordinates of nodes i and L in the current configuration with components in the global coordinate system. The orthogonal matrix $\mathbf{Q} = [\mathbf{e}_1' \; \mathbf{e}_2']$ is a rotation matrix, so that $\mathbf{Q}^T$ transforms global vector components to local vector components. The unit basis vectors $\mathbf{e}_1'$ and $\mathbf{e}_2'$ which define the local co-rotating coordinate frame are defined in terms of θ with global components as follows:

$$\mathbf{e}'_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \quad \mathbf{e}'_2 = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}. \tag{5.2}$$

Lastly, $\mathbf{X}^i_\ell$ represents the material coordinates of node i in the local coordinate frame. It is noted that $\mathbf{X}^i_\ell = \mathbf{Q}^T\mathbf{X}^{iL} = \mathbf{Q}^T\left(\mathbf{X}^i - \mathbf{X}^L\right)$.

Based on the reference and current configurations Eq. (5.1) expresses the local nodal coefficient components for node i in terms of known quantities $\mathbf{x}^{iL}$, $\mathbf{X}^i_\ell$ and as yet unknown quantity θ, the angle of rotation of the local co-rotating coordinate frame. This unknown quantity is determined in the next section.

5.2.2 Co-rotating frame angle of rotation

The angle of rotation of the co-rotated coordinate frame is found by assuming that the local spin, due to local nodal displacements in the current configuration, is equal to zero. The local spin is evaluated at the centroid of the Voronoi cell for node L in the reference configuration (see Jetteur and Cescotto [43])

$$\Omega_\ell = \frac{\partial u_{1\ell}}{\partial Y_\ell} - \frac{\partial u_{2\ell}}{\partial X_\ell} = 0. \tag{5.3}$$

The meshfree approximation for the displacement field in terms of the local nodal coefficients, $\mathbf{d}_\ell$, is written as

$$u_{j\ell} = \boldsymbol{\phi}^T\mathbf{d}_{j\ell}, \quad (j = 1, 2) \tag{5.4}$$

where $\boldsymbol{\phi}$ is the vector of nodal basis functions and $\mathbf{d}_{j\ell}$ denotes the vector of local nodal coefficients associated with degree of freedom j.

Substituting Eq. (5.4) into Eq. (5.3) gives

$$\Omega_\ell = \left(\frac{\partial \boldsymbol{\phi}}{\partial Y_\ell}\right)^T \mathbf{d}_{1\ell} - \left(\frac{\partial \boldsymbol{\phi}}{\partial X_\ell}\right)^T \mathbf{d}_{2\ell} = \mathbf{a}_\ell^T \mathbf{d}_\ell, \tag{5.5}$$

where

$$\mathbf{a}_\ell = \begin{bmatrix} \frac{\partial \phi_1}{\partial Y_\ell} \\ -\frac{\partial \phi_1}{\partial X_\ell} \\ \vdots \\ \frac{\partial \phi_n}{\partial Y_\ell} \\ -\frac{\partial \phi_n}{\partial X_\ell} \end{bmatrix} \quad \text{and} \quad \mathbf{d}_\ell = \begin{bmatrix} d^1_{1\ell} \\ d^1_{2\ell} \\ d^2_{1\ell} \\ d^2_{2\ell} \\ \vdots \\ d^n_{1\ell} \\ d^n_{2\ell} \end{bmatrix}. \tag{5.6}$$

Note that $\mathbf{a}_\ell$ is evaluated at the centroid of the Voronoi cell for node L in local material coordinates $\mathbf{X}_\ell$ (which is equivalent to evaluation in global material coordinates $\mathbf{X}$) and hence is a fixed vector. Next substitute Eq. (5.1) into Eq. (5.5) to get

$$\Omega_\ell = \sum (\mathbf{a}^i_\ell)^T \mathbf{d}^i_\ell = \sum (\mathbf{a}^i_\ell)^T (\mathbf{Q}^T \mathbf{x}^{iL} - \mathbf{X}^i_\ell) = 0 \tag{5.7}$$

or

$$\Omega_\ell = \sum (\mathbf{a}^i_\ell)^T (\mathbf{Q}^T \mathbf{x}^{iL}) - \sum (\mathbf{a}^i_\ell)^T (\mathbf{X}^i_\ell) = 0. \tag{5.8}$$

Noting that the last term of Eq. (5.8) is zero and expanding the first term yields

$$\Omega_\ell = \sum (\mathbf{a}^i_\ell)^T \left(\cos\theta \begin{pmatrix} x^{iL} \\ y^{iL} \end{pmatrix} + \sin\theta \begin{pmatrix} y^{iL} \\ -x^{iL} \end{pmatrix} \right) = 0. \tag{5.9}$$

Finally, from Eq. (5.9)

$$\Omega_\ell = a \sin\theta + b\cos\theta = 0, \tag{5.10}$$

where

$$a = \sum (\mathbf{a}_\ell^i)^T \begin{pmatrix} y^{iL} \\ -x^{iL} \end{pmatrix} \quad \text{and} \quad b = \sum (\mathbf{a}_\ell^i)^T \begin{pmatrix} x^{iL} \\ y^{iL} \end{pmatrix}. \tag{5.11}$$

It is convenient to rewrite a and b as follows:

$$a = \mathbf{c}^T \bar{\mathbf{x}}, \tag{5.12}$$

where

$$\mathbf{c} = \begin{bmatrix} 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & -1 & & & \vdots \\ 0 & 0 & 1 & 0 & & & \\ \vdots & & & & \ddots & & \\ 0 & 0 & & & & 0 & -1 \\ 0 & 0 & \dots & & & 1 & 0 \end{bmatrix} \mathbf{a}_\ell, \quad \bar{\mathbf{x}} = \begin{bmatrix} x^{1L} \\ y^{1L} \\ x^{2L} \\ y^{2L} \\ \vdots \\ x^{nL} \\ y^{nL} \end{bmatrix} \tag{5.13}$$

and $b = \mathbf{a}_\ell^T \bar{\mathbf{x}}$. Note that $\mathbf{c}$ in Eq. (5.13) is a $2n$ by $2n$ matrix depending on the number of neighbors n and similarly $\bar{\mathbf{x}}$ is a $2n$ by 1 vector. With these expressions in hand it is possible to solve for the angle of rotation θ, which from Eq. (5.10) is

$$\theta = \tan^{-1}\left(\frac{-b}{a}\right). \tag{5.14}$$

5.2.3 Derivation of the tangent stiffness matrix

To derive the tangent stiffness matrix first consider the local internal force vector, $\mathbf{q}_{L\ell}$, for node L and its neighboring nodes, which is written as

$$\mathbf{q}_{L\ell} = \int_{\Omega} \mathbf{B}^T \boldsymbol{\sigma} \, dV = \mathbf{K}_{t\ell} \mathbf{d}_{\ell}, \tag{5.15}$$

where $\mathbf{B}$ is the local strain-displacement matrix, $\boldsymbol{\sigma}$ are the local Cauchy stresses and $\mathbf{K}_{t\ell}$ (t is used to denote a tangent stiffness matrix) represents the local tangent stiffness matrix, which, as part of the iterative process for global equilibrium, is possibly constructed by considering inelastic material behavior.

Next, note that the local nodal coefficients, $\mathbf{d}_{\ell}$, are related to the global nodal coefficients, $\mathbf{d}$, via some function, f, i.e.,

$$\mathbf{d}_{\ell} = f(\mathbf{d}, \mathbf{e}'_1, \mathbf{e}'_2), \tag{5.16}$$

and the variation of Eq. (5.16) leads to the relationship

$$\delta \mathbf{d}_{\ell} = \mathbf{T} \delta \mathbf{d}, \tag{5.17}$$

where $\mathbf{T}$ is some as yet to be determined transformation matrix. Virtual work at the local and global level are equivalent so that

$$(\delta \mathbf{d}_{\ell})^T \mathbf{q}_{L\ell} = (\delta \mathbf{d})^T \mathbf{q}_{L}. \tag{5.18}$$

The global internal forces in terms of the local internal forces are found by making use of

Eq. (5.17) and Eq. (5.18), which yields

$$(\mathbf{T}\delta\mathbf{d})^T\mathbf{q}_{L\ell} = (\delta\mathbf{d})^T\mathbf{T}^T\mathbf{q}_{L\ell} = (\delta\mathbf{d})^T\mathbf{q}_L, \tag{5.19}$$

$$\mathbf{q}_L = \mathbf{T}^T\mathbf{q}_{L\ell} = \mathbf{T}^T\mathbf{K}_{t\ell}\mathbf{d}_\ell \tag{5.20}$$

where the last equality comes from the use of Eq. (5.15).

To obtain the global stiffness matrix the variation of Eq. (5.20) is taken, which gives

$$\delta\mathbf{q}_L = \mathbf{T}^T\delta\mathbf{q}_{L\ell} + \delta\mathbf{T}^T\mathbf{q}_{L\ell} = \mathbf{T}^T\mathbf{K}_{t\ell}\delta\mathbf{d}_\ell + \mathbf{K}_{t\sigma}\delta\mathbf{d} = \mathbf{T}^T\mathbf{K}_{t\ell}\mathbf{T}\delta\mathbf{d} + \mathbf{K}_{t\sigma}\delta\mathbf{d}, \tag{5.21}$$

where $\delta\mathbf{T}^T\mathbf{q}_{L\ell}$ is represented as shown by $\mathbf{K}_{t\sigma}\delta\mathbf{d}$. The matrix $\mathbf{K}_{t\sigma}$ is the initial stiffness matrix and the last equality in Eq. (5.21) is found by making use of Eq. (5.17). Equation (5.21) yields

$$\delta\mathbf{q}_L = \left[\mathbf{T}^T\mathbf{K}_{t\ell}\mathbf{T} + \mathbf{K}_{t\sigma}\right]\delta\mathbf{d} = \mathbf{K}_T\delta\mathbf{d}, \tag{5.22}$$

where $\mathbf{K}_T$ represents the tangent stiffness matrix at the global level.

To find the transformation matrix in Eq. (5.22), the variation of Eq. (5.1) is taken to give

$$\delta\mathbf{d}^i_\ell = \mathbf{Q}^T\delta\mathbf{x}^{iL} + \delta\mathbf{Q}^T\mathbf{x}^{iL}. \tag{5.23}$$

From Figure 5.1, note that

$$\mathbf{x}^{iL} = \mathbf{X}^{iL} + \mathbf{d}^i - \mathbf{d}^L = \mathbf{X}^{iL} + \mathbf{d}^{iL}. \tag{5.24}$$

Taking the variation of Eq. (5.24) gives

$$\delta\mathbf{x}^{iL} = \delta\mathbf{X}^{iL} + \delta\mathbf{d}^{iL} = \delta\mathbf{d}^{iL}, \tag{5.25}$$

where the last step results since $\delta\mathbf{X}^{iL}$ is zero. Substituting Eq. (5.25) into Eq. (5.23) yields

$$\delta\mathbf{d}_\ell^i = \mathbf{Q}^T \delta\mathbf{d}^{iL} + \delta\mathbf{Q}^T \mathbf{x}^{iL}. \tag{5.26}$$

Taking the variation of $\mathbf{Q}^T$ gives

$$\delta\mathbf{Q}^T = \delta \begin{bmatrix} \mathbf{e}_1' & \mathbf{e}_2' \end{bmatrix}^T = \begin{bmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix}^T \delta\theta. \tag{5.27}$$

Consequently,

$$\delta\mathbf{Q}^T \mathbf{x}^{iL} = \begin{bmatrix} -s & c \\ -c & -s \end{bmatrix} \begin{pmatrix} x^{iL} \\ y^{iL} \end{pmatrix} \delta\theta = \begin{bmatrix} -sx^{iL} + cy^{iL} \\ -cx^{iL} - sy^{iL} \end{bmatrix} \delta\theta = \mathbf{Q}^T \begin{pmatrix} y^{iL} \\ -x^{iL} \end{pmatrix} \delta\theta. \tag{5.28}$$

Now substituting Eq. (5.28) into Eq. (5.26) yields

$$\delta\mathbf{d}_\ell^i = \mathbf{Q}^T \delta\mathbf{d}^{iL} + \mathbf{Q}^T \begin{pmatrix} y^{iL} \\ -x^{iL} \end{pmatrix} \delta\theta. \tag{5.29}$$

If $\mathbf{Q}^T \delta\mathbf{d}^L$ is added to Eq. (5.29) it should have no effect if the local coordinate system computations correctly satisfy the infinitesimal strain-free rigid body requirements. This addition to Eq. (5.29) gives

$$\delta\mathbf{d}_\ell^i = \mathbf{Q}^T \delta\mathbf{d}^i + \mathbf{Q}^T \begin{pmatrix} y^{iL} \\ -x^{iL} \end{pmatrix} \delta\theta. \tag{5.30}$$

To obtain $\delta\theta$ differentiate Eq. (5.14) by recalling that $\frac{d(\tan^{-1} u)}{dx} = \frac{1}{1+u^2}\frac{du}{dx}$. This gives

$$\delta\theta = \frac{1}{1+\frac{b^2}{a^2}}\delta(-ba^{-1}) = \frac{a^2}{a^2+b^2}(-\delta b a^{-1} + a^{-2} b\delta a) = \frac{a^2}{a^2+b^2}\left(\frac{b\delta a}{a^2} - \frac{a\delta b}{a^2}\right). \qquad (5.31)$$

Rearranging and simplifying Eq. (5.31) yields

$$\delta\theta = \frac{b\delta a - a\delta b}{a^2+b^2} = \frac{1}{a^2+b^2}(b\mathbf{c}^T - a\mathbf{a}_\ell^T)\delta\mathbf{d} = \mathbf{v}^T\delta\mathbf{d}. \qquad (5.32)$$

Substituting $\delta\theta = \mathbf{v}^T\delta\mathbf{d}$ into Eq. (5.30) gives

$$\delta\mathbf{d}_\ell^i = \mathbf{Q}^T\delta\mathbf{d}^i + \mathbf{Q}^T\begin{pmatrix} y^{iL} \\ -x^{iL}\end{pmatrix}\mathbf{v}^T\delta\mathbf{d}. \qquad (5.33)$$

Next, realizing that $\mathbf{Q}^T\begin{pmatrix} y^{iL} \\ -x^{iL}\end{pmatrix} = \begin{pmatrix} y_\ell^i \\ -x_\ell^i\end{pmatrix}$, Eq. (5.33) becomes

$$\delta\mathbf{d}_\ell^i = \mathbf{Q}^T\delta\mathbf{d}^i + \begin{pmatrix} y_\ell^i \\ -x_\ell^i\end{pmatrix}\mathbf{v}^T\delta\mathbf{d}. \qquad (5.34)$$

Using Eq. (5.34), an alternative form is written for all neighbors and the current point L as

$$\delta\mathbf{d}_\ell = (\bar{\mathbf{Q}} + \bar{\mathbf{x}}_\ell\mathbf{v}^T)\delta\mathbf{d}, \qquad (5.35)$$

where

$$\bar{\mathbf{Q}} = \begin{bmatrix} [\mathbf{Q}^T] & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & [\mathbf{Q}^T] & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & [\mathbf{Q}^T] \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.36}$$

and

$$\bar{\mathbf{x}}_\ell^T = [\ y_\ell^1 \quad -x_\ell^1 \quad y_\ell^2 \quad -x_\ell^2 \quad \dots \quad y_\ell^n \quad -x_\ell^n\ ]. \tag{5.37}$$

Note that $\bar{\mathbf{Q}}$ is a $2n$ by $2n$ matrix. Then, comparing Eq. (5.35) with Eq. (5.17) it is evident that

$$\mathbf{T} = \bar{\mathbf{Q}} + \bar{\mathbf{x}}_\ell \mathbf{v}^T. \tag{5.38}$$

All that remains to construct the tangent stiffness matrix (see Eq. (5.22)) is the initial stiffness matrix $\mathbf{K}_{t\sigma}$. The initial stiffness matrix arises from (see Eq. (5.21))

$$\delta \mathbf{T}^T \mathbf{q}_{L\ell} = \mathbf{K}_{t\sigma} \delta \mathbf{d}. \tag{5.39}$$

The variation of $\mathbf{T}^T$ is found by representing the first part of Eq. (5.39) as

$$\delta \mathbf{T}^T \mathbf{q}_{L\ell} = \delta \mathbf{T}^1 \mathbf{q}_{L\ell}^1 + \delta \mathbf{T}^2 \mathbf{q}_{L\ell}^2 + \ldots = \sum_{j=1}^{2n} \delta \mathbf{T}^j \mathbf{q}_{L\ell}^j, \tag{5.40}$$

where $\mathbf{T}^j$ is the jth column of $\mathbf{T}^T$ and $\mathbf{q}_{L\ell}^j$ is the jth component of $\mathbf{q}_{L\ell}$ (which is a scalar). Working now only with the first term in the summation Eq. (5.40) and using the transpose of Eq. (5.38) gives

$$\delta \mathbf{T}^1 \mathbf{q}^1_{L\ell} = \mathbf{q}^1_{L\ell} \delta \left\{ \left\{ \begin{array}{c} \mathbf{e}'_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} + y^1_\ell \mathbf{v} \right\} = \mathbf{q}^1_{L\ell} \mathbf{G}^1 \delta \mathbf{d}, \tag{5.41}$$

where $\mathbf{0}^T = [\ 0\ \ 0\ ]$. From Eq. (5.41), $\mathbf{G}^1 \delta \mathbf{d}$ must be determined. This is given by

$$\mathbf{G}^1 \delta \mathbf{d} = \delta \left\{ \left\{ \begin{array}{c} \mathbf{e}'_1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} + y^1_\ell \mathbf{v} \right\} = \left\{ \begin{array}{c} \mathbf{e}'_2 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} \delta\theta + \delta y^1_\ell \mathbf{v} + y^1_\ell \delta \mathbf{v}. \tag{5.42}$$

Now note that δy^1_ℓ comes from Eq. (5.35), i.e.,

$$\delta y^1_\ell = \left\{ \left[\begin{array}{ccccc} \mathbf{e}'^T_2 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \end{array} \right] - x^1_\ell \mathbf{v}^T \right\} \delta \mathbf{d}. \tag{5.43}$$

To see this, consider for a moment the generic variable $\mathbf{w}$. The variation of this variable in local coordinates is related to the variation of itself in global coordinates as (see Eq. (5.35))

$$\delta \mathbf{w}_\ell = (\bar{\mathbf{Q}} + \bar{\mathbf{x}}_\ell \mathbf{v}^T) \delta \mathbf{w}, \tag{5.44}$$

where $\mathbf{w} = \mathbf{X} + \mathbf{d}$ and $\mathbf{w}_\ell = \mathbf{X}_\ell + \mathbf{d}_\ell = \mathbf{x}_\ell$. Specifically, $\mathbf{w}^T_\ell = \left\{ \begin{array}{ccccc} x^1_\ell & y^1_\ell & \ldots & x^n_\ell & y^n_\ell \end{array} \right\}$. Then observe that $\delta \mathbf{w} = \delta \mathbf{d}$ since $\delta \mathbf{X} = 0$. By taking only the row of Eq. (5.44) associated with δy^1_ℓ Eq. (5.43) is obtained.

If the last term of Eq. (5.42) is not included (just yet), using Eqs. (5.32), (5.42) and

(5.43) yields

$$\mathbf{G}^{1,a} = \left\{ \begin{array}{c} \mathbf{e}_2' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} \mathbf{v}^T + \mathbf{v} \left\{ \begin{array}{c} \mathbf{e}_2' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\}^T - x_\ell^1 \mathbf{v}\mathbf{v}^T, \tag{5.45}$$

which is symmetric. In order to obtain the complete form of $\mathbf{G}$ it is necessary to determine $\delta\mathbf{v}$. To this end, $\mathbf{v}^T$ of Eq. (5.32) is rewritten as (in the following $\mathbf{g} = \mathbf{a}_\ell$)

$$\mathbf{v} = \frac{1}{a^2 + b^2}(b\mathbf{c} - a\mathbf{g}). \tag{5.46}$$

Then, taking the variation of Eq. (5.46) gives, by use of the product rule,

$$\delta\mathbf{v} = \delta\left(\frac{1}{a^2 + b^2}\right)(b\mathbf{c} - a\mathbf{g}) + \frac{1}{a^2 + b^2}\delta(b\mathbf{c} - a\mathbf{g}). \tag{5.47}$$

Now observe that

$$\delta\left(\frac{1}{a^2 + b^2}\right) = \delta\left((a^2 + b^2)^{-1}\right) = -(a^2 + b^2)^{-2}(2a\mathbf{c}^T + 2b\mathbf{g}^T)\delta\mathbf{d} = \frac{-2(a\mathbf{c}^T + b\mathbf{g}^T)}{(a^2 + b^2)^2}\delta\mathbf{d}, \tag{5.48}$$

and

$$\delta(b\mathbf{c} - a\mathbf{g}) = \delta b\mathbf{c} - \delta a\mathbf{g} = (\mathbf{c}\mathbf{g}^T - \mathbf{g}\mathbf{c}^T)\delta\mathbf{d}. \tag{5.49}$$

Substituting Eqs. (5.48) and (5.49) into Eq. (5.47) yields

$$\delta\mathbf{v} = \frac{-2(a\mathbf{c}^T + b\mathbf{g}^T)}{(a^2 + b^2)^2}(b\mathbf{c} - a\mathbf{g})\delta\mathbf{d} + \frac{(\mathbf{c}\mathbf{g}^T - \mathbf{g}\mathbf{c}^T)}{a^2 + b^2}\delta\mathbf{d}$$

$$= \frac{-2(ab\mathbf{c}\mathbf{c}^T - a^2\mathbf{g}\mathbf{c}^T + b^2\mathbf{c}\mathbf{g}^T - ab\mathbf{g}\mathbf{g}^T)}{(a^2 + b^2)^2}\delta\mathbf{d} + \frac{(a^2 + b^2)(\mathbf{c}\mathbf{g}^T - \mathbf{g}\mathbf{c}^T)}{(a^2 + b^2)^2}\delta\mathbf{d}$$

$$= \frac{-2(ab\mathbf{c}\mathbf{c}^T - a^2\mathbf{g}\mathbf{c}^T + b^2\mathbf{c}\mathbf{g}^T - ab\mathbf{g}\mathbf{g}^T)}{(a^2+b^2)^2}\delta\mathbf{d} + \frac{(a^2\mathbf{c}\mathbf{g}^T - a^2\mathbf{g}\mathbf{c}^T + b^2\mathbf{c}\mathbf{g}^T - b^2\mathbf{g}\mathbf{c}^T)}{(a^2+b^2)^2}\delta\mathbf{d}$$

$$= \frac{(-2ab\mathbf{c}\mathbf{c}^T + a^2\mathbf{g}\mathbf{c}^T - b^2\mathbf{c}\mathbf{g}^T + 2ab\mathbf{g}\mathbf{g}^T + a^2\mathbf{c}\mathbf{g}^T - b^2\mathbf{g}\mathbf{c}^T)}{(a^2+b^2)^2}\delta\mathbf{d}$$

$$= \left[\frac{2ab(\mathbf{g}\mathbf{g}^T - \mathbf{c}\mathbf{c}^T) + (a^2-b^2)(\mathbf{c}\mathbf{g}^T + \mathbf{g}\mathbf{c}^T)}{(a^2+b^2)^2}\right]\delta\mathbf{d}$$

$$= \mathbf{V}^T\delta\mathbf{d}, \tag{5.50}$$

which is symmetric. The expression in brackets above for $\mathbf{V}^T$ is identical to that found by Crisfield and Moita [28] with the exception that their denominator is not squared (a likely typographical error). Note also, that the last term of $\mathbf{G}^1$, which includes the variation of $\mathbf{v}$, may be neglected (see Crisfield [27]). However, for completeness it is kept here. Hence, having

$$\mathbf{G}^{1,b} = y_\ell^1\mathbf{V}^T, \tag{5.51}$$

the final expression for $\mathbf{G}^1$ is found as

$$\mathbf{G}^1 = \mathbf{G}^{1,a} + \mathbf{G}^{1,b}. \tag{5.52}$$

However, the matrix $\mathbf{G}^1$ is only sufficient to construct the first term in the summation Eq. (5.40). The other $\mathbf{G}^j$ matrices are found similarly. Hence, the initial stiffness matrix is calculated as $\mathbf{K}_{t\sigma} = \sum_{j=1}^{2n}\mathbf{q}_{L\ell}^j\mathbf{G}^j$and subsequently the entire tangent stiffness matrix as given in Eq. (5.22).

As additional information the calculation of $\mathbf{G}^2$ is demonstrated next. Starting with the second term in the summation of Eq. (5.40) yields

$$\delta \mathbf{T}^2 \mathbf{q}_{L\ell}^2 = \mathbf{q}_{L\ell}^2 \delta \left\{ \left\{ \begin{array}{c} \mathbf{e}_2' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} - x_\ell^1 \mathbf{v} \right\} = \mathbf{q}_{L\ell}^2 \mathbf{G}^2 \delta \mathbf{d}. \tag{5.53}$$

From Eq. (5.53), $\mathbf{G}^2 \delta \mathbf{d}$ is determined, which is given by

$$\mathbf{G}^2 \delta \mathbf{d} = \delta \left\{ \left\{ \begin{array}{c} \mathbf{e}_2' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} - x_\ell^1 \mathbf{v} \right\} = \left\{ \begin{array}{c} -\mathbf{e}_1' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} \delta\theta + \delta(-x_\ell^1)\mathbf{v} + (-x_\ell^1)\delta \mathbf{v}. \tag{5.54}$$

Now note that δx_ℓ^1 arises in a similar fashion as that described after Eq. (5.43), i.e.,

$$\delta x_\ell^1 = \left\{ \left[\begin{array}{ccccc} \mathbf{e}_1'^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{array} \right] + y_\ell^1 \mathbf{v}^T \right\} \delta \mathbf{d}. \tag{5.55}$$

Then taking Eq. (5.54) and using Eqs. (5.32), (5.55) and (5.50) gives

$$\mathbf{G}^2 = \left\{ \begin{array}{c} -\mathbf{e}_1' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\} \mathbf{v}^T + \mathbf{v} \left\{ \begin{array}{c} -\mathbf{e}_1' \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right\}^T - y_\ell^1 \mathbf{v}\mathbf{v}^T - x_\ell^1 \mathbf{V}^T. \tag{5.56}$$

Last, expressions for the generic cases of $\mathbf{G}^{2i-1}$ and $\mathbf{G}^{2i}$ are given below. In general, for $i = 1$ to n

$$\mathbf{G}^{2i-1} = \begin{array}{c} 1 \\ \vdots \\ i \\ \vdots \\ n \end{array} \begin{Bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{e}'_2 \\ \vdots \\ \mathbf{0} \end{Bmatrix} \mathbf{v}^T + \mathbf{v} \begin{Bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{e}'_2 \\ \vdots \\ \mathbf{0} \end{Bmatrix}^T - x^i_\ell \mathbf{v}\mathbf{v}^T + y^i_\ell \mathbf{V}^T, \tag{5.57}$$

$$\mathbf{G}^{2i} = \begin{Bmatrix} \mathbf{0} \\ \vdots \\ -\mathbf{e}'_1 \\ \vdots \\ \mathbf{0} \end{Bmatrix} \mathbf{v}^T + \mathbf{v} \begin{Bmatrix} \mathbf{0} \\ \vdots \\ -\mathbf{e}'_1 \\ \vdots \\ \mathbf{0} \end{Bmatrix}^T - y^i_\ell \mathbf{v}\mathbf{v}^T - x^i_\ell \mathbf{V}^T. \tag{5.58}$$

5.2.4 Nonlinear material stiffness

If plasticity is included in the co-rotational formulation then it is necessary to update the material properties during each load step of the analysis. Hence, the local tangent stiffness matrix $\mathbf{K}_{t\ell}$ takes the following form:

$$\mathbf{K}_{t\ell} = \int_\Omega \mathbf{B}^T \mathbf{C}_{ep} \mathbf{B} \, dV, \tag{5.59}$$

where $\mathbf{C}_{ep}$ is the elasto-plastic modulus matrix that evolves during each load step if the local trial stresses fall outside the yield surface such as in a plane stress $J2$ plasticity formulation with radial return (see Simo and Taylor [80] and Simo and Hughes [78]). All other formulas remain the same.

5.2.5 Load control algorithm for a meshfree co-rotational formulation

An algorithm for the co-rotational formulation in a meshfree setting is given below. The given algorithm is for a linear elastic or elasto-plastic material. In the following, the vectors $\mathbf{d}$ represent meshfree nodal coefficients whereas the vectors $\mathbf{u}$ represent displacements.

1. Set up storage variables

2. Loop over load increments

   (a) Create $\Delta \boldsymbol{f}^{n+1}$

   (b) Construct $\boldsymbol{f}_{int}^{n}$ for each node L and its neighbors based on current stresses, $\boldsymbol{\sigma}^{n}$

   (c) Construct $\mathbf{K}^{n}$ based on current $\boldsymbol{f}_{int}^{n}$ with current $\Delta \mathbf{d}_{I\ell}$ values and $\mathbf{u}^{n}$

   (d) Modify $\Delta \boldsymbol{f}^{n+1}$ and $\mathbf{K}^{n}$ to account for supports

   (e) Solve for $\Delta \mathbf{d}_{I} = (\mathbf{K}^{n})^{-1} \Delta \boldsymbol{f}^{n+1}$

   (f) Calculate displacements $\Delta \mathbf{u}^{n+1}$ based on $\Delta \mathbf{d}_{I}$

   (g) Calculate $\Delta \mathbf{d}_{I\ell}$ based on $\mathbf{u}^{n} + \Delta \mathbf{u}^{n+1}$

   (h) Calculate the incremental nodal strains based on the latest $\Delta \mathbf{d}_{I\ell}$

   (i) Calculate current stresses $\boldsymbol{\sigma}^{n+1}$ (based on elastic or elasto-plastic constitutive relations)

   (j) Construct $\boldsymbol{f}_{int}^{n+1}$ for each node L and its neighbors based on current stresses, $\boldsymbol{\sigma}^{n+1}$

   (k) Update global stiffness to get $\mathbf{K}^{n+1}$ based on current $\boldsymbol{f}_{int}^{n+1}$ with current $\Delta \mathbf{d}_{I\ell}$ values and $\mathbf{u}^{n} + \Delta \mathbf{u}^{n+1}$

   (l) Modify $\mathbf{K}^{n+1}$ to account for supports

(m) Initialize variables for Newton-Raphson iterations, $k = 0$, $tol = 10^{-2}$ and $maxiter = 100$

(n) Calculate the residual $\mathbf{g}_{(k)}^{n+1} = \boldsymbol{f}_{int(k)}^{n+1} - \boldsymbol{f}^{n+1}$

(o) Begin Newton-Raphson Iterations, while $\left|\mathbf{g}_{(k)}^{n+1}\right| > tol$ and $k <= maxiter$

i. $\delta\mathbf{d}_I^{(k)} = -(\mathbf{K}_{(k)}^{n+1})^{-1}\mathbf{g}_{(k)}^{n+1}$

ii. $\Delta\mathbf{d}_I^{(k+1)} = \Delta\mathbf{d}_I^{(k)} + \delta\mathbf{d}_I^{(k)}$

iii. Calculate displacements $\Delta\mathbf{u}_{(k+1)}^{n+1}$ based on $\Delta\mathbf{d}_I^{(k+1)}$

iv. Calculate $\Delta\mathbf{d}_{I\ell}^{(k+1)}$ based on $\mathbf{u}^n + \Delta\mathbf{u}_{(k+1)}^{n+1}$

v. Calculate incremental nodal strains based on the latest $\Delta\mathbf{d}_{I\ell}^{(k+1)}$

vi. Calculate current stresses $\boldsymbol{\sigma}_{(k+1)}^{n+1}$

vii. Construct $\boldsymbol{f}_{int(k+1)}^{n+1}$ for each node L and its neighbors based on current stresses, $\boldsymbol{\sigma}_{(k+1)}^{n+1}$

viii. Update global stiffness to get $\mathbf{K}_{(k+1)}^{n+1}$ based on current $\boldsymbol{f}_{int(k+1)}^{n+1}$ with current $\Delta\mathbf{d}_{I\ell}^{(k+1)}$ values and $\mathbf{u}^n + \Delta\mathbf{u}_{(k+1)}^{n+1}$

ix. Modify $\mathbf{K}_{(k+1)}^{n+1}$ to account for supports

x. Calculate the residual $\mathbf{g}_{(k+1)}^{n+1}$

xi. Update iteration variable $k = k + 1$

xii. If $k = maxiter$ and $\mathbf{g}_{(k)}^{n+1} > tol$, provide warning that equilibrium tolerance not met

(p) End while loop of Newton-Raphson iterations

(q) Update strain $\boldsymbol{\varepsilon}^{n+1} = \boldsymbol{\varepsilon}^n + \Delta\boldsymbol{\varepsilon}^{n+1}$

(r) Update displacements $\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta\mathbf{u}^{n+1}$

(s) Update stresses

3. End loop over prescribed load increments

5.2.6 Numerical implementation details

In this chapter the numerical examples are created using the max-ent shape functions given in Section 3.2. These shape functions are used in the weak form with nodal integration in just the same way as that described previously for MLS shape functions in Section 4.2 (of course continuous blending is not necessary when using the max-ent shape functions). Furthermore, the numerical implementation details follow closely those given in Section 4.4 with the following observations. The stiffness matrices as constructed in Section 4.4 are for a first order small strain analysis in local coordinates. These stabilized local stiffness matrices, as part of the co-rotational formulation described by Eq. (5.22) are transformed to the global level, and along with the additional initial stiffness matrix, are assembled into a global stiffness matrix as is commonly done in standard finite element methods.

In the first order analysis described in Section 4.4 it is possible to incrementally update the internal force vector, construct a residual and iterate for equilibrium at the global level. However, within the co-rotational framework it becomes essential instead to construct the local internal force vectors nodewise at the local level from the current total stresses. This is necessary because local internal forces are required to construct the initial stiffness matrices. Therefore, consistent with the stabilization scheme described in Section 4.4.2 and as given in Eq. (4.16), the local internal forces take the following form:

$$\mathbf{q}_{L\ell} = \boldsymbol{f}_{\ell}^{int} = \int_{\Omega} \mathbf{B}^T \boldsymbol{\sigma} dV + \alpha_s \sum_{c \in T_a}^{n} \left[\int_{\Omega} \left(\mathbf{B} - \mathbf{B}^c \right)^T \mathbf{C}_s \left(\mathbf{B} - \mathbf{B}^c \right) \mathbf{d}_{\ell} \; dV_c \right]. \tag{5.60}$$

After construction of the initial stiffness matrices these local internal forces are transformed to the global level according to the first part of Eq. (5.20) and assembled into a global internal force vector as part of the residual calculation process. The residual is then used in the Newton-Raphson scheme to enforce global equilibrium as indicated in the algorithm of Section 5.2.5. By use of the consistent internal forces an optimum rate of convergence is maintained in the iterations for global equilibrium.

The creation of $\mathbf{C}_s$ for plastic materials remains the same as that described previously in Section 4.4.2. However, for elastic materials $\mathbf{C}_s = \mathbf{C}_{elast}$.

5.3 Numerical examples

Numerical results for plane stress are presented using an implicit Newton-Raphson iteration scheme at the global level. At the constitutive level, for inelastic materials, $J2$ plasticity with an implicit Newton-Raphson iteration scheme using radial return is employed [80].

5.3.1 Linear elastic cantilever beam

A linear elastic cantilever beam with $\nu = 0.0$, $E = 100.0$ ksi and uniform thickness $t = 2.0$ inches is loaded with a uniform load along the vertical free end. Deflected shapes are shown for a regular and irregular grid of nodes in Figures 5.2a and 5.2b, respectively. A load displacement plot of a meshfree co-rotational cantilever beam is compared to a 1D co-rotational beam finite element in Figure 5.2c. The software OpenSees [65] is used to obtain the results for the 1D beam element. The 1D beam element model uses ten beam elements. An analytical solution based on Euler-Bernoulli beam theory with consideration of axial deformations is also shown in the load displacement plot. For both regular and irregular grids, the agreement of the current method with the other solutions is excellent. The final deflected shape of the cantilever corresponds to a load of 10 kips, and the plot of stress (Figure 5.2d) with increasing displacement is shown for model node A indicated.

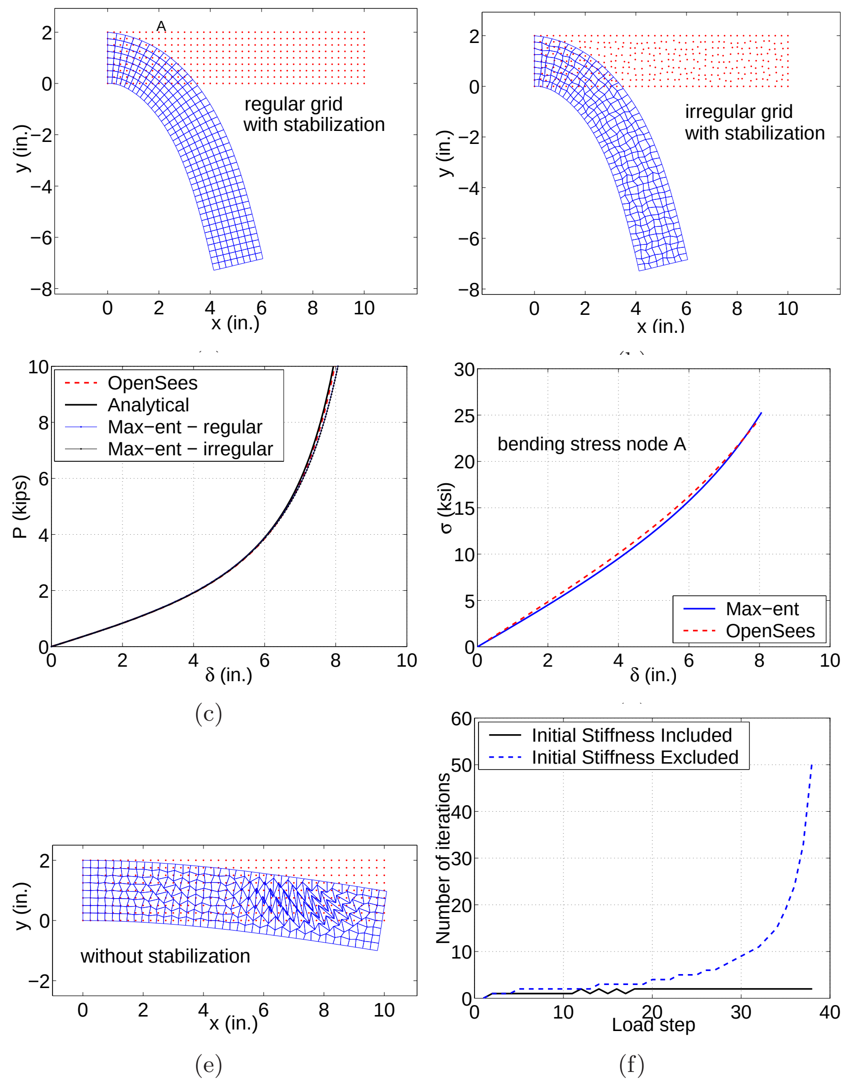


Figure 5.2: 2D co-rotational meshfree cantilever beam solution compared to 1D co-rotational beam element: (a) final deflected shape (regular grid); (b) final deflected shape (irregular grid); (c) load displacement plot; (d) bending stress; (e) spurious deflected shape without stabilization; and (f) iterations per load step with and without the initial stiffness matrix included.

The loading takes the strains of the small strain formulation higher than is recommended (25 percent bending strain at node A), however, the results illustrate robust and smooth results and the effectiveness of the stabilization in suppressing hourglass modes. Figure 5.2e illustrates the hourglass modes that result when no stabilization is used. In fact, without stabilization, the analysis crashes and fails even to converge at loads of about 0.4 kips.

For the above stabilized solution a series of 100 load increments are applied with at most two Newton-Raphson iterations per load step required to reach equilibrium for a residual tolerance of 10^{-2}. It is interesting to note that in the past some researchers applied co-rotational formulations without including the variation of the transformation matrix which leads to the initial stiffness matrix. Although for this cantilever beam problem comparable results are obtained for lower load levels by excluding the initial stiffness matrix the number of iterations required for equilibrium increases dramatically. Figure 5.2f illustrates the number of iterations required without the initial stiffness matrix for the first 38 load steps in an analysis identical to the one described above. The analysis was terminated after the 38th load step since iterations required began to exceed 100. This demonstrates the value of a consistent formulation and the loss of the quadratic rate of convergence when the initial stiffness matrix is excluded.

5.3.2 Linear elastic circular shallow arch

A pin supported linear elastic circular shallow arch is loaded with a concentrated force at its central point as shown in Figure 5.3a. For the arch, $\nu = 0.0$, $E = 68.948$ kN/mm^2, radius is 10581.6 mm, cross-section radial depth is 79.2 mm, and the width of the cross-section is 25.4 mm. The span of the arch from pin to pin is 2540 mm. The arch is modeled with 2761 meshfree nodes, which is similar to 2500 quadrilateral elements. In Figure 5.3b, the load displacement response, exhibiting snap-through behavior, is compared

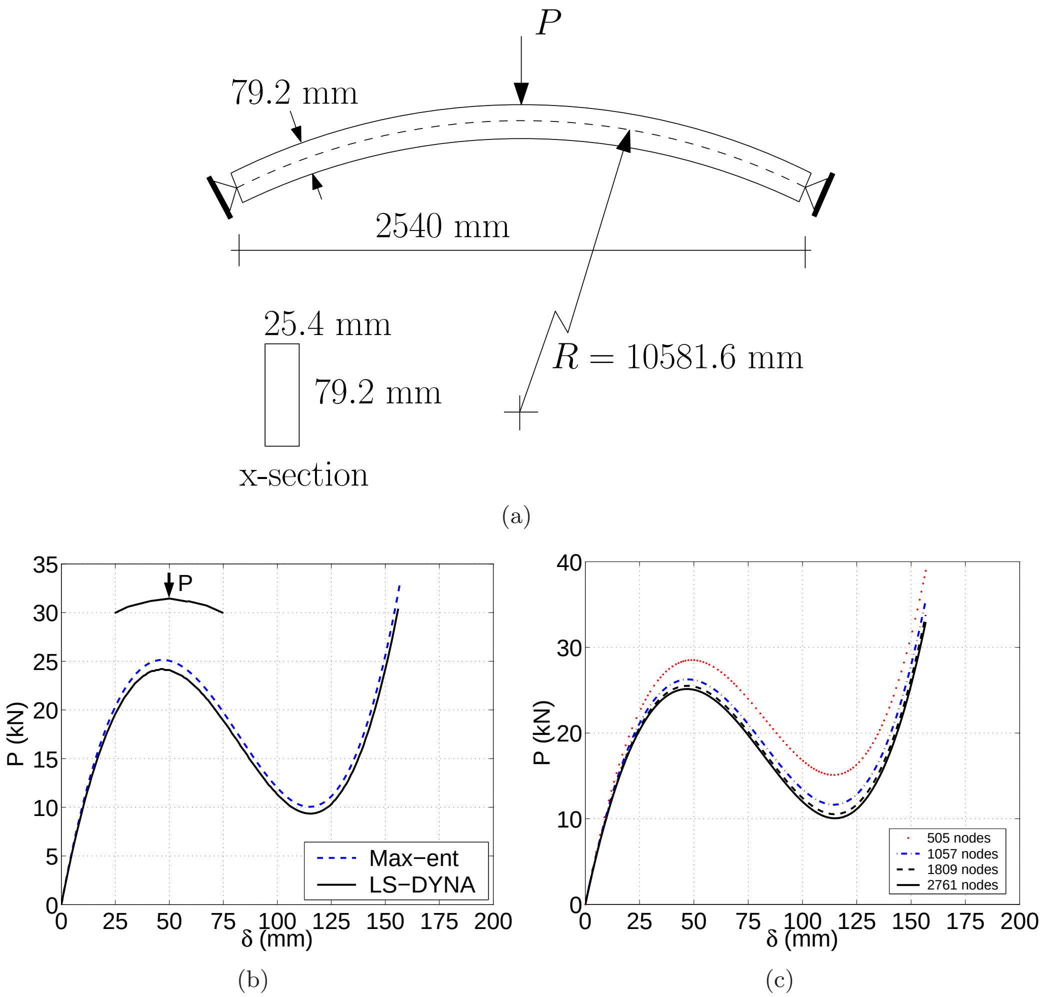
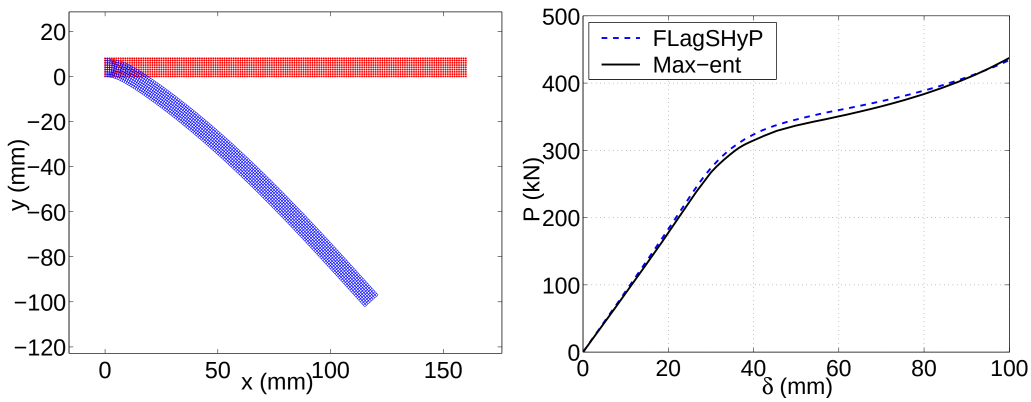


Figure 5.3: Results for pin-supported linear elastic circular shallow arch. (a) initial arch configuration; (b) load displacement plot; (c) max-ent model convergence.



(a) (b)

Figure 5.4: Results for cantilever with inelastic material behavior included. (a) deflected shape (450 kN load); and (b) load displacement plot.

to results found by using 2500 quadrilateral membrane elements in LS-DYNA [38]. The load displacement results are obtained by using a single node displacement control scheme using 115 displacement increments (see Clarke et al. [23]). The agreement with LS-DYNA is very good. Numerical results are also shown in Figure 5.3c illustrating the convergence of the meshfree method with grid refinement. The analysis does not correctly capture the snap through behavior when the initial stiffness matrix is excluded, which further illustrates the importance of a variationally consistent co-rotational formulation.

5.3.3 Elasto-plastic cantilever

As mentioned previously, once a co-rotational formulation is constructed, it is relatively easy to include traditional small strain inelastic material behavior. To demonstrate this, in Figure 5.4a, a cantilever beam is loaded at its free end with a load of 450 kN, which is well beyond first yield. A plastic hinge develops and large rotations of the cantilever beam result. The maximum bending strain is 25 percent. The maximum-entropy model has 1449 nodes. The cantilever is 2 mm thick, 8 mm in depth, and is 160 mm long. A plane stress elasto-plastic material ($J2$ plasticity with radial return [80]) is used with

$E = 140930$ kN/mm^2, $\nu = 0.3$, a linear hardening modulus of 284 kN/mm^2, and a yield stress of 1550 kN/mm^2. Finite element results, obtained by using the finite element large strain hyperelasto-plastic program (FLagShyP) by Bonet and Wood [19], are included for comparison. For the FLagShyP model the same material properties are used, along with 1280 hexahedral elements. Although the FLagShyP model is for a hyperelasto-plastic material, for relatively small strains this is comparable to the elasto-plastic material used in the maximum-entropy model. It is evident from the load displacement plot of Figure 5.4b that the finite element and maximum-entropy results are in very good agreement. The analysis is completed by using a displacement control scheme of 42 increments at the free end of the cantilever.

5.3.4 Elastic and elasto-plastic T-frame

A T-frame is loaded with a point load as shown in Figure 5.5a. Figure 5.5b shows the vertical displacement of node A versus load for elastic and elasto-plastic ($J2$ plasticity) materials. The deflected shapes, for the load levels labeled in Figure 5.5b, are illustrated in Figures 5.5c–5.5e. The maximum bending strain is 10 percent and 21 percent for the elastic and elasto-plastic cases, respectively. The results are intended to demonstrate the ability of the co-rotational formulation to capture large displacements and rotations for elastic and elasto-plastic cases. The material properties are as follows: $E = 29000$ ksi, $\nu = 0.3$, linear hardening modulus $\bar{H} = 100$ ksi and yield stress $f_y = 550$ ksi. The beams and columns of the frame are 4 inch in depth and 1 inch thick. For the elastic case an artificially high yield stress is used so that yielding is avoided during the entire simulation. The analysis is completed using 70 equal (0.3 inch) steps of displacement control at node A.

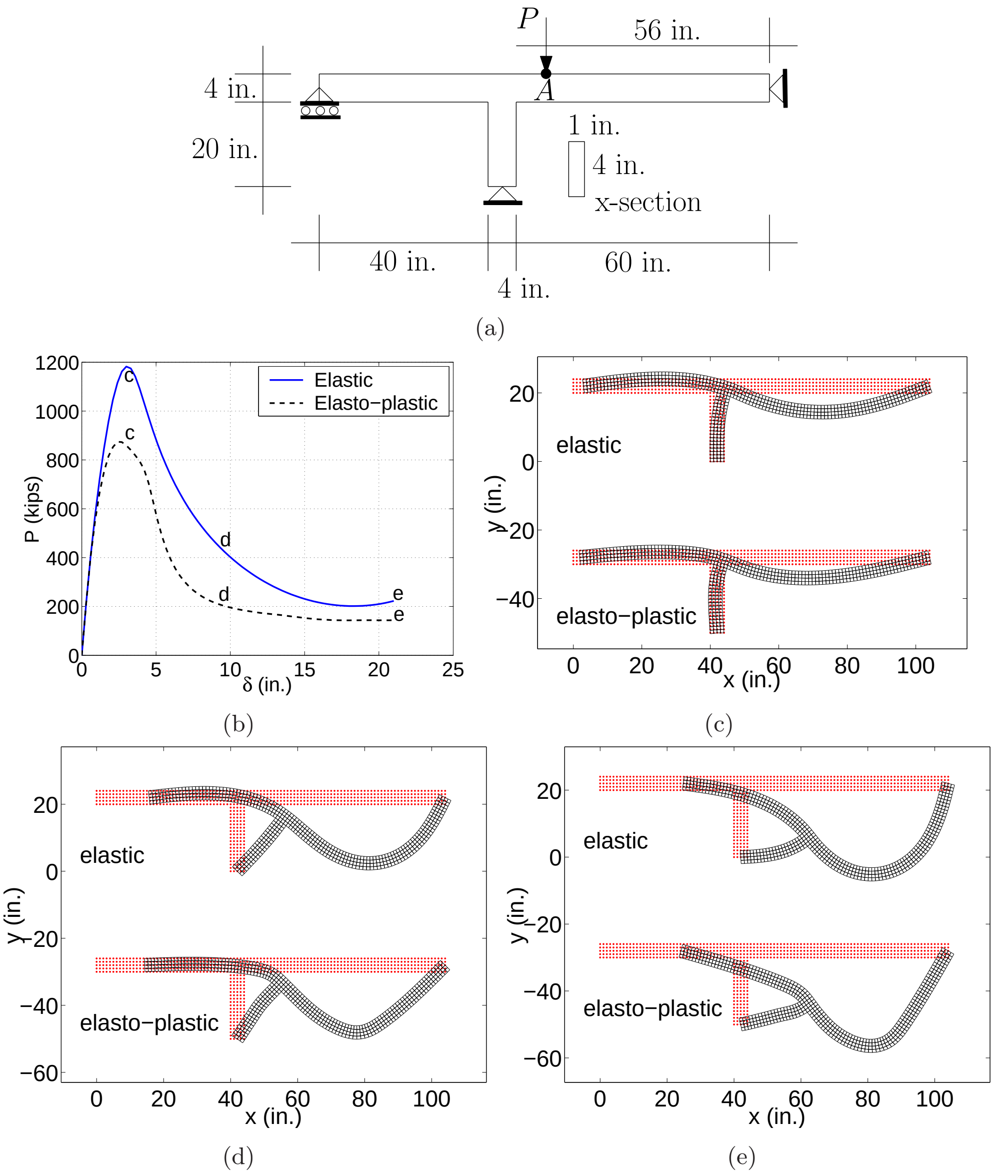


Figure 5.5: Results for T-frame. (a) initial configuration; (b) load versus displacement for elastic and elasto-plastic cases; and (c),(d),(e) deflected shapes at load levels indicated in (b).

5.4 Concluding remarks

Maximum-entropy basis functions were successfully employed in a meshfree co-rotational formulation for two-dimensional continua. A variationally consistent formulation was found not only to dramatically improve rate of convergence but essential to solving certain problems. Nodal integration and stabilization was applied within the formulation to representative problems for validation. Benchmark problems such as the cantilever beam, shallow arch, and a T-frame were considered with elastic and elasto-plastic material behavior, and the numerical results with the present co-rotational formulation were found to be in good agreement with finite element computations. Notably, the use of stabilization when performing nodal integration prevented the presence of spurious modes in the deflected shape. The numerical results evince that maximum-entropy basis functions combined with a co-rotational formulation is an effective technique for including large displacements and rotations. This work provides impetus for future research-work on the extensions to finite strains and three-dimensional computations to further the effort to improve large-scale collapse simulations.

Chapter 6

Applications and Validation Examples

In previous chapters examples are intended to demonstrate that the implementation of the meshfree analysis is functioning properly. As is shown a variety of benchmark problems are successfully solved. Specifically, linear elastic, inelastic and geometrically nonlinear problems are presented. In the current chapter it is the intent to present problems of a more practical nature specifically focused on everyday problems of structural engineering in building design. Hence, a variety of frame structures and building structure subassemblies are analyzed using the meshfree analysis program developed as part of this research. In particular, problems including cyclic loading, catenary action and stiffness softening are all demonstrated and compared to either experimental or FE results. Each of these conditions are potentially necessary for an accurate collapse analysis. It is intended by these examples to demonstrate that the analysis program created as part of this research successfully accomplishes the goals set out initially to advance collapse technology.

6.1 Portal frames under monotonic and cyclic loading conditions

A portal frame under different loading conditions is analyzed using the meshfree analysis program and results are compared to experimental results for the frame as given by Toma and Chen [88]. This frame is reported in the literature specifically to allow researchers to compare second-order analyses to the experimental results. Loading results are given for both monotonic and cyclic loading conditions. In each of these loading conditions the frames are tested with and without constant axial forces applied at the top of the columns. Hence, the cases without the constant axial forces are analyzed by a first-order meshfree analysis (max-ent shape functions) with inelasticity included. For the cases with the constant axial forces a second-order (co-rotational) meshfree analysis (max-ent shape functions) with inelasticity is used to capture the stiffness softening behavior. In all cases displacement control is used to obtain the results. The test configuration for the frames is shown in Figure 6.1. The reported material and wide flange beam section properties for the experimental frames are given in Tables 6.1 and 6.3 respectively. The actual properties used in each of the computer simulations are reported in Tables 6.2 and 6.4. In all cases the modulus of elasticity is 29000 ksi and Poisson's ratio is 0.3. Exponential hardening as proposed by Voce [93] is used for all four frames (see Appendix E for the expression for exponential hardening). In each case the frames are discretized by 946 nodes.

6.1.1 Monotonic loading

Results for monotonic loading with no axial loads are shown in Figure 6.2a and 6.2b. It is evident that the experimental results are less stiff and have an earlier yield load than is predicted by the computer simulation. Toma and Chen explain that the frame specimen

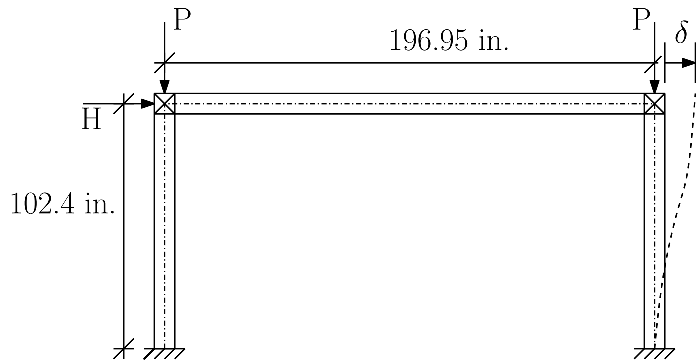


Figure 6.1: Test set up and loading arrangement for frame by Toma and Chen [88].

	Column		Beam	
Frame Loading	σ_y (ksi)	σ_u (ksi)	σ_y (ksi)	σ_u (ksi)
Monotonic ($P = 0$)	38	62.4	38	59.6
Monotonic ($P = 153$ kips)	39.2	62.6	40.6	61.3
Cyclic ($P = 0$)	37.8	60.35	40.3	58.9
Cyclic ($P = 153$ kips)	38.1	60.8	36.1	55.7

Table 6.1: Reported material properties for experimental frame by Toma and Chen [88].

	Columns and Beams		
Frame Loading	σ_y (ksi)	σ_u (ksi)	Exponential Hardening Variable δ
Monotonic ($P = 0$)	38.0	61.0	16.3
Monotonic ($P = 153$ kips)	38.0	59.6	16.3
Cyclic ($P = 0$)	38.0	59.6	16.3
Cyclic ($P = 153$ kips)	37.0	59.6	16.3

Table 6.2: Material properties used in computer simulation.

	Columns			Beams		
Frame Loading	d (in.)	I_{xx} (in.^{4})	t_w (in.)	d (in.)	I_{xx} (in.^{4})	t_w (in.)
Monotonic ($P = 0$)	6.89	65.8	0.295	9.84	97.3	0.236
Monotonic ($P = 153$ kips)	6.89	68.2	0.295	9.84	97.3	0.236
Cyclic ($P = 0$)	6.89	69.2	0.295	9.84	98.0	0.236
Cyclic ($P = 153$ kips)	6.89	68.2	0.295	9.84	97.8	0.236

Table 6.3: Reported beam and column section properties for experimental frame by Toma and Chen [88].

	Columns			Beams		
Frame Loading	d (in.)	I_{xx} (in.^{4})	t_w (in.)	d (in.)	I_{xx} (in.^{4})	t_w (in.)
Monotonic ($P = 0$)	6.89	65.8	0.295	9.84	97.3	0.236
Monotonic ($P = 153$ kips) Cyclic ($P = 0$) Cyclic ($P = 153$ kips)	6.89	68.2	0.295	9.84	97.3	0.236

Table 6.4: Beam and column section properties used in computer simulation.

imperfections caused additional bending moments which explains the early yielding and reduced initial stiffness. No information regarding the magnitude of the imperfections is given and hence the computer simulation is for straight members. The authors also indicate that base fixity in the actual frames was not perfect. No effort is made to compensate for the frame imperfections in the computer results. Also, due to the frame imperfections it is deemed unnecessarily precise to model the slight differences in material properties between beams and columns. Hence, the material properties as indicated in Table 6.2 are employed. However, in general, the computer simulation follows the overall behavior of the experimental results. Plastic hinges form at the base of the columns and at beam column joints as shown in the deflected shape of the frame, where the deflections shown are for the case of loading at the end of the monotonic load simulation.

For the monotonic loading with axial loads of 153 kips per column a post peak stiffness softening occurs as lateral load is increased. This is evident in Figure 6.2c. Such behavior is only captured by including second order effects in the computer simulation (i.e. by co-rotational formulation) and by using a displacement control scheme in the incremental iterative analysis. Again, for this case the experimental frame is less stiff, has imperfect base fixity and has a lower yield value, than the computer simulation, due to imperfections in the actual experimental frame. However, the basic behavior of the frame and the stiffness softening is evident in the computer results. Plastic hinges result at locations in the frame

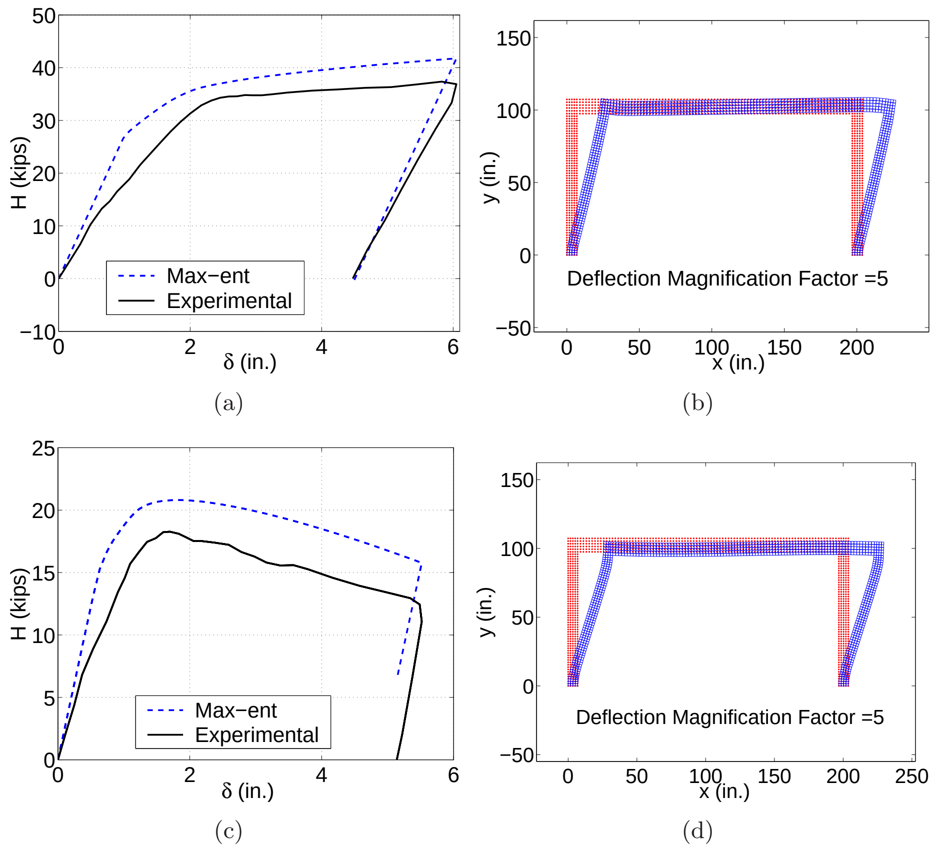


Figure 6.2: Monotonic loading of portal frame: (a) load versus displacement response ($P = 0$); (b) deflected shape at end of loading ($P = 0$); (c) load versus displacement ($P = 153$ kips); (d) deflected shape at end of loading ($P = 153$ kips).

similar to the case for monotonic loading with no axial loads. The deflected shape is shown in Figure 6.2d for the final load condition when the simulation is terminated.

6.1.2 Cyclic loading

Cyclic loading behavior for frames is of interest because it is similar to loading which takes place during seismic events. The resulting load displacement plot for cyclic loading results in a hysteresis loop consisting of alternating elasto-plastic loading and elastic unloading portions of the curve. To correctly capture unloading it is essential to have an implicit nonlinear analysis procedure at both the global and constitutive levels. As mentioned previously this consists of implicit Newton-Raphson iterations at the global level and an implicit Newton-Raphson iteration procedure with radial return to achieve $J2$ plastic consistency at the constitutive level. The following examples demonstrate that the computer simulation correctly reproduces the general loading unloading behavior.

Cyclic loading is applied with no constant axial loads. The resulting hysteresis loop for both the experimental and simulated results are shown in Figure 6.3a. Unloading along the elastic curve is demonstrated at the end of every elasto-plastic loading cycle. Isotropic exponential hardening is used to obtain the results for the computer simulation. Frame imperfections, imperfect base fixity and the resulting decreased initial yield account for most of the differences between the experimental and simulated results. However, the computer simulation is in good agreement with the overall behavior of the experimental frame test.

For the case of cyclic loading with constant axial loads of 153 kips per column the results are shown in Figure 6.3b. The most notable difference is the post peak softening that becomes evident with load cycles of sufficient lateral displacement. This behavior is captured by the computer simulation which includes second order effects by use of the co-rotational formulation. The primary cause of the stiffness softening is the external column axial forces which are applied at the beginning of the simulation and held constant throughout the cyclic loading. Again, frame imperfections as in previous cases cause most of the deviation between the experimental and simulated results.

6.1.3 Summary of results

For the monotonic and cyclic loaded frames of this section it is found that two of the most important factors are yield stress and frame panel zone modeling. When comparing to experimental results an accurate value of the true yield stress is essential. However, as

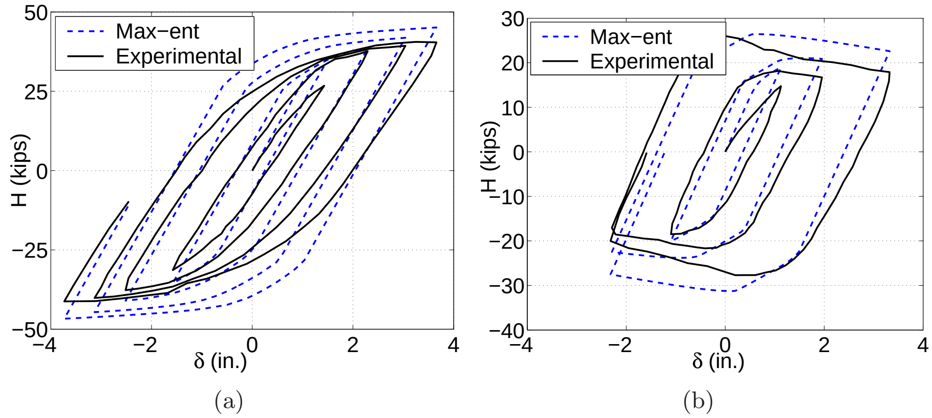


Figure 6.3: Cyclic loading of portal frame: (a) load versus displacement response ($P = 0$); (b) load versus displacement ($P = 153$ kips).

observed above this importance is adversely affected when specimen imperfections strongly influence the initial yield values. The panel zones are potentially very influential in the load displacement response of frames. Obviously the rotation behavior at the joints has a system wide effect and can dramatically influence first yield and even second order effects. In the frames tested above the experimental frames included X-shaped stiffeners in the frame panel zone. Hence in the computer simulation the effect of the stiffeners is included by using increased thickness along the diagonal nodes of the panel zone. Such modeling tends to force yielding and distortion away from the panel zone and into the beam and column sections just outside the panel joint.

6.2 Development of catenary action in building frame subassembly

For a variety of reasons a building column may cease to function properly. This could be due to blast damage, overloading of the column, fire damage or damage and instability due to earthquake loadings. Loss of column function likely results in a roof or upper floor(s) collapse. However, it is possible due to the column removal that loads transfer to adjacent columns and structural elements. Under these conditions instead of collapse the floor or roof system beams sag into the shape of a catenary and, if the structural system is adequate, the collapse does not progress further. This is of course preferable and therefore the formation of catenary behavior during partial collapse is of interest. It is useful to know what forces result when a column is removed and catenary behavior tries to develop. If such forces are determined ahead of time by a computer analysis it is then possible to design the structural system adequately and possibly prevent progressive collapse. In this section such a scenario is considered and several subassemblies of a building structure are modeled and the computer analysis attempts to capture catenary behavior.

Khandelwal and El-Tawil [46] present computer simulations for beam-column assemblies extracted from an eight story special moment resisting perimeter frame system (see Figure 6.4a). In particular, simulations are run for assemblies from the first, fifth and seventh stories. The simulations are intended to model the behavior of a typical subassembly when a column is removed from the building frame system as shown in Figure 6.4b. The computer simulations by Khandelwal and El-Tawil are quite advanced. Dynamic effects are accounted for by constructing the model within LS-DYNA and using the explicit dynamic solver. The Gurson material model is used for the finite elements near the beam-column joints to account for fracture and three dimensional shell type elements are used to account for flange and lateral torsional buckling effects. For regions away from the joints the simpler $J2$ plasticity model is used. The left half of a subassembly is illustrated in Figure 6.4c. The column lengths at the left are from the beam centerline to the mid-height column inflection points. At the right the middle column is cut off at the top and bottom of the beam flange as shown.

Meshfree co-rotational computer simulations for the beam-column subassembly are pre-

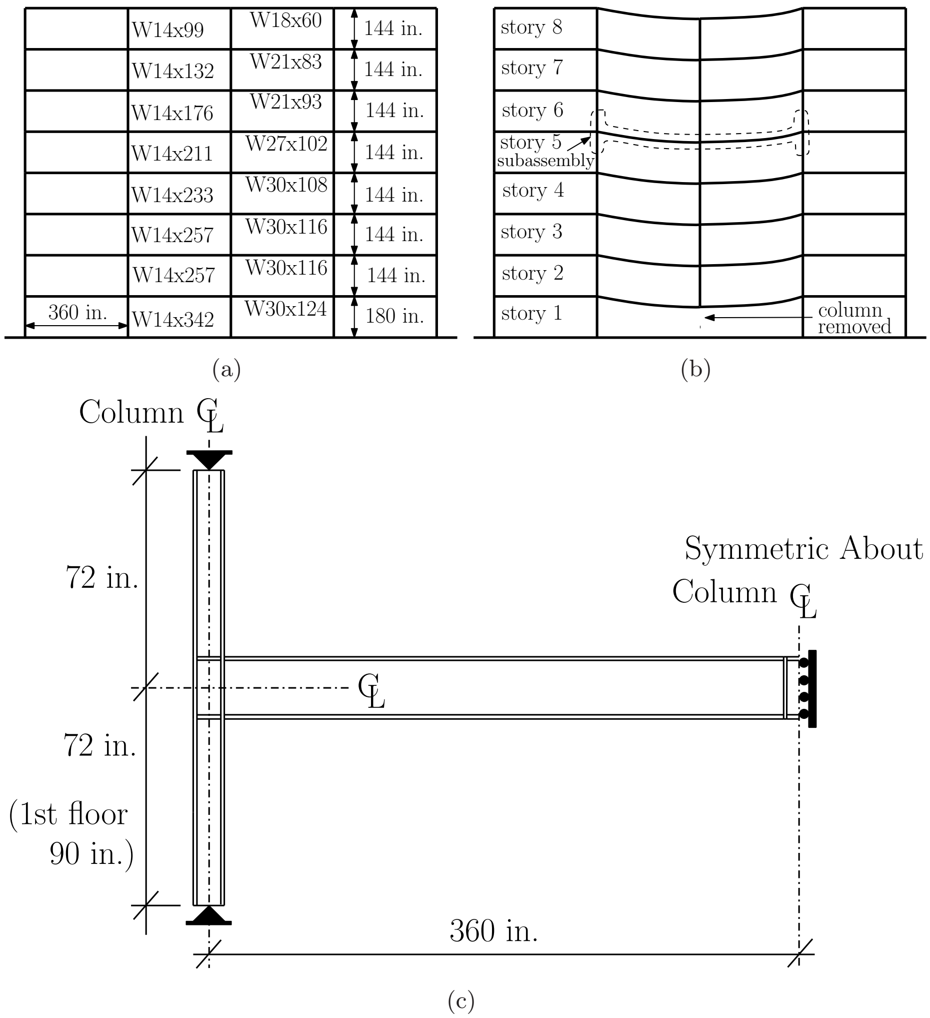


Figure 6.4: Eight story special moment resisting frame system: (a) interior beam and column sizes and typical frame dimensions; (b) resulting catenary action when bottom column removed; (c) left half of a typical beam-column subassembly.

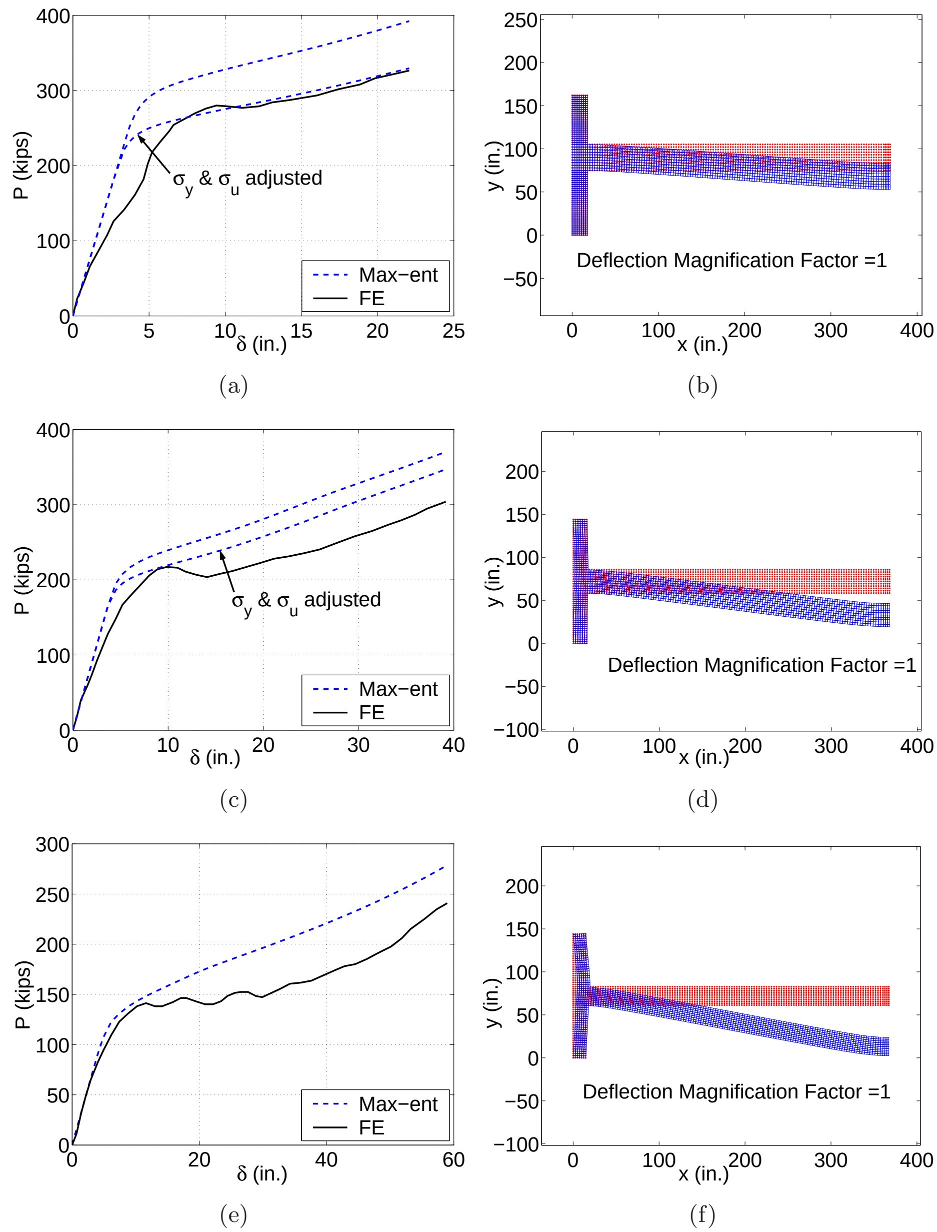


Figure 6.5: Building structure beam-column subassemblies, formation of catenary behavior: (a)(c)(e) load versus displacement response, stories 1,5,7 respectively; (b)(d)(f) deflected shape at end of loading, stories 1,5,7 respectively.

sented to compare with the results by Khandelwal and El-Tawil. Due to the symmetry of a total beam-column assembly only half of the beam-column assembly is modeled as shown in Figure 6.4c. The top and bottom of the columns at the left are pin supported and the column stub at the right is fully supported along the column web centerline with rollers at all of the meshfree nodes. Khandelwal and El-Tawil ran simulations for girders with reduced beam sections and non-reduced beam sections. The results obtained using the meshfree co-rotational simulation are compared to the results with non-reduced beam sections at stories 1, 5 and 7. Load displacement results are obtained by vertical displacement control at the middle column centerline of the beam-column subassembly. The load displacement results and final deflected shapes are shown in Figure 6.5.

The meshfree simulations do not consider dynamic effects, fracture, flange buckling or lateral torsional buckling effects. By only considering the non-reduced beam sections the lateral torsional buckling effects are avoided. However, the remaining effects cause a noticeable deviation between the meshfree and finite element results. Although this is the case, the general behavior of the meshfree simulations is similar to the more advanced finite element simulations. In fact, tension stiffening is most evident at story 7 where the load displacement curve starts to curve upwards near the end of the simulation indicating the onset of catenary behavior. To account for the dynamic and buckling effects included in the advanced FE computer simulation the beam yield and ultimate stresses are adjusted for the first (17% reduction) and fifth (9% reduction) story cases in the meshfree simulation. The effect of making these reductions for the beam is shown in the load displacement curves of Figures 6.5a and 6.5c.

For the first story the half beam-column assembly is discretized with 1822 nodes and a finer discretization of 1967 nodes (more nodes across column depth). The difference in discretization causes only minor changes in the load versus displacement plot. For the

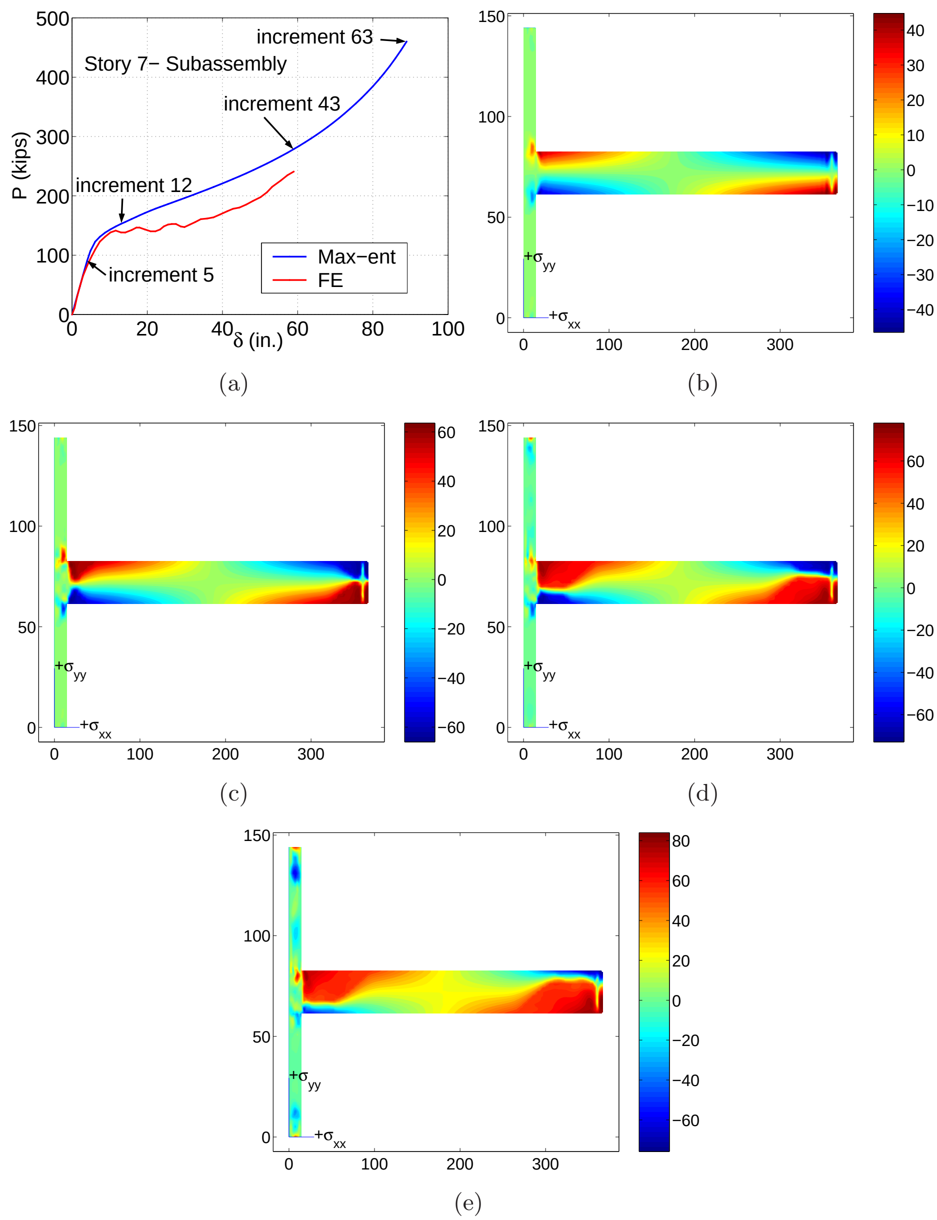


Figure 6.6: Story 7 development of catenary behavior in meshfree simulation: (a) load displacement plot with additional displacement increments demonstrating the development of tension stiffening; (b)(c)(d)(e) σ_{xx} (ksi) at end of displacement increments 5, 12, 43 and 63 respectively.

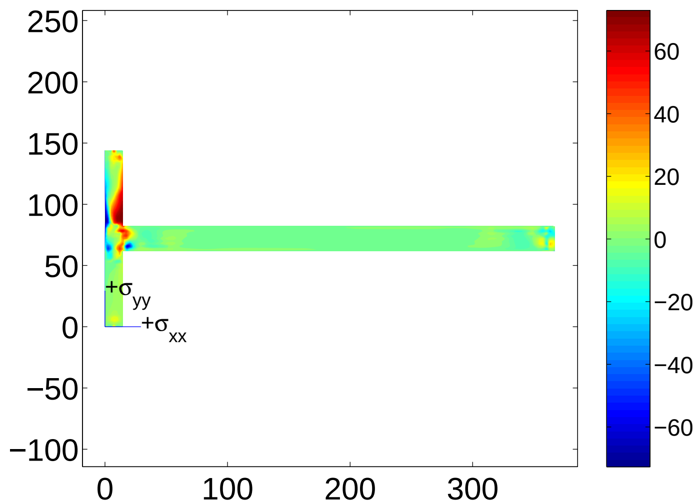
Story	Beam	Column Below	Column Above
1	W30x124	W14x342	W14x257
5	W27x102	W14x211	W14x176
7	W21x83	W14x132	W14x99

Table 6.5: Story subassembly beam and column list.

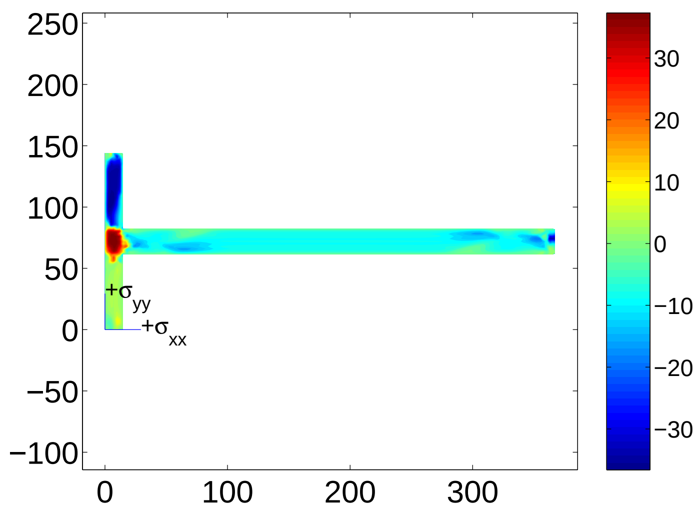
fifth and seventh story results a discretization of 1766 nodes is used. Flanges for beams and columns are modeled using the procedure similar to that described in Appendix D. Stiffeners across column webs are modeled to provide the same area as the adjacent beam flanges. Table 6.5 contains a summary of the AISC [70] beams and columns modeled at stories 1, 5 and 7.

For the seventh story the meshfree simulation is repeated with more displacement increments included at the end of the previous simulation. This is done to demonstrate the ability of the meshfree analysis program to definitively capture catenary behavior. A new load displacement plot for this simulation is given in Figure 6.6a, where tension stiffening is prominently illustrated at the end of the curve. In the plot, displacement increments 5, 12, 43 and 63 are identified. At the end of these increments the current state of stress is recorded to allow observation of the evolution of the σ_{xx} stresses in the beam. The stress plots for these cases are given in Figures 6.6bcde. By following the evolution of stresses in the beam it is apparent that linear elastic stresses develop, next plastic hinges form at each end of the beam, next tension stiffening starts to shift the neutral axis at the beam ends toward the compression flange, and finally the tension stiffening shifts the neutral axis very close to the compression flange. This behavior is symptomatic of sagging elements developing the tension of catenary behavior.

For completeness, at the end of displacement increment 43 the remaining stresses σ_{yy} and σ_{xy} are given in Figure 6.7. The σ_{yy} stresses of Figure 6.7a show the development of



(a)



(b)

Figure 6.7: Story 7 remaining stresses at end of increment 43: (a) σ_{yy} (ksi); and (b) σ_{xy} (ksi).

a plastic hinge at the base of the column just above the beam. The development of this plastic hinge is consistent with the findings of Khandelwal and El-Tawil.

6.2.1 Summary of results

The meshfree co-rotational formulation definitely captures tension stiffening. The stress plots also illustrate the shifting of the neutral axis in the plastic hinge regions, which indicates development of the tensile forces of catenary behavior. Although the finite element results are more advanced the general trends in the load displacement plots are captured by the meshfree method.

6.3 'El Zanaty' portal frame with stiffness softening

El Zanaty et al. [33] first analyzed the frame shown in Figure 6.8 as a test case for second order analysis simulations. As a result this frame is now a common benchmark problem. As indicated in the paper by Toma et al. [89], White [96] analyzed the El Zanaty frame using a second order inelastic analysis (plastic-zone theory) and the stress-strain response shown in Figure 6.9. Furthermore, White included initial residual stresses in the frame analysis by subdividing the cross-section of the beam (essentially using fiber elements). Each beam and column of the frame was modeled by 12 bar type finite elements. The frame was loaded by constant axial forces of $P = 0.6P_y$ per column. Then a monotonically increasing lateral load H was applied at the centerline of the frame beam. The results of the finite element analysis are summarized by Toma et al. [89]. These results are compared to a similar analysis using the meshfree co-rotational formulation.

For the meshfree co-rotational analysis the domain of the frame is discretized with 2853 nodes. The meshfree analysis uses plane stress with $J2$ plasticity. The stress-strain response

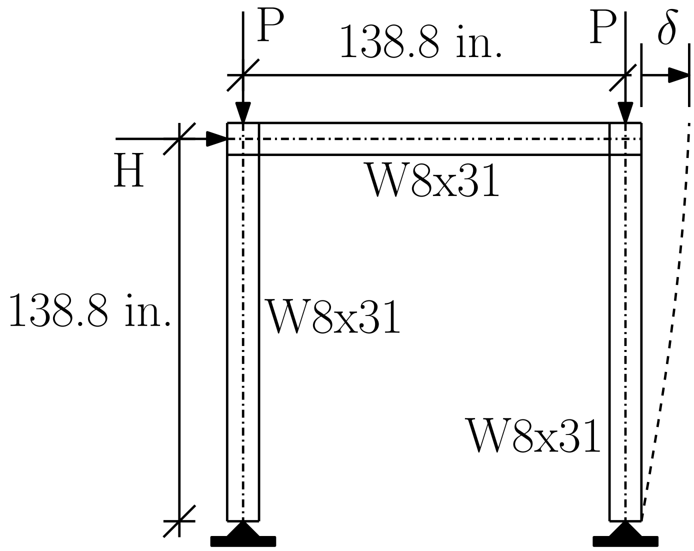


Figure 6.8: El Zanaty portal frame test set up.

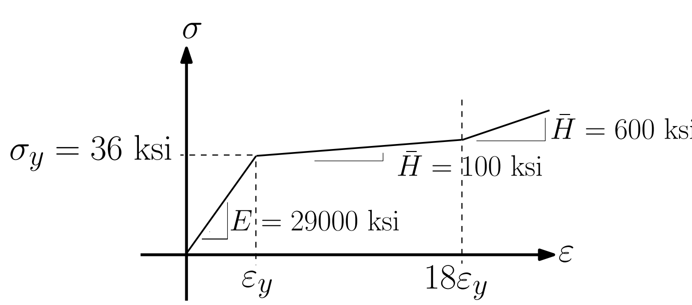


Figure 6.9: El Zanaty frame assumed stress-strain curve.

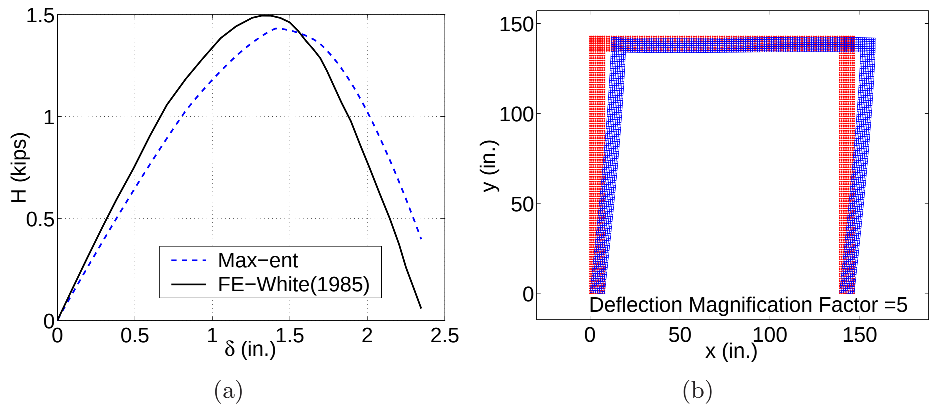


Figure 6.10: El Zanaty portal frame: (a) load versus displacement response; (b) final deflected shape.

used for hardening is the same as given in Figure 6.9. The meshfree analysis however does not include consideration of initial residual stresses that exist in the beams and columns due to fabrication. The load displacement results are shown in Figure 6.10a and the final deflected shape is shown in Figure 6.10b.

The meshfree co-rotational load versus displacement results are in general agreement with the results obtained by White. The analysis definitely captures the stiffness softening behavior due to the large axial forces relative to the yield capacity of the columns. It is likely that the deviation is due to using a continuum rather than 1D beam elements, excluding the effect of residual forces and the inevitable differences in behavior of the beam column joints by using 1D elements versus a continuum.

Chapter 7

Conclusions

The feasibility of using meshfree methods for simulations involving large scale structural systems subject to large displacements and large rotations is evident from the results provided. This work also provides additional research directions in structural collapse simulation, particularly in the area of meshfree co-rotational formulations. In the following a review of the work presented along with discussions of findings and suggestions for future research is given.

7.1 Use of meshfree basis functions

Meshfree methods are given as a possible alternative to current collapse simulation techniques. The methodology is successfully employed with meshfree MLS and maximum-entropy basis functions for problems of plane stress. However, although MLS basis functions are successfully implemented it is found that the bookkeeping required in the computer implementation is cumbersome due to difficulties of enforcing essential boundary conditions, which, in this work, is accomplished by the method of continuous blending. It was initially thought that this blending method using meshfree with finite elements would lend some computational advantages by allowing the use of meshfree only where necessary and finite elements elsewhere due to their better computational efficiency. However, with the introduction of maximum-entropy basis functions two benefits are realized. First, the cumbersome method of continuous blending is no longer necessary for enforcement of essential boundary conditions since maximum-entropy basis functions have a weak Kronecker-delta property on their boundary. Second, because of this the maximum-entropy basis functions in theory are amenable for combination with finite elements. For this reason maximum-entropy basis functions are shown to be very effective, easy to implement and fairly robust for constructing a meshfree formulation.

7.2 *I*-shaped beam frame structures

The meshfree method using MLS shape functions for plane stress is applied to problems of frame like structures. These problems only consider small strains, displacements and rotations while using the nonlinear material model of $J2$ plasticity. Several representative problems are successfully solved. In particular, the problems are for structures composed of beams with I-shaped cross-sections. This necessitates the specification of larger thickness in flange regions of the plane stress domain. A successful methodology for doing this is demonstrated by the beam and frame problems shown. It is also worth mentioning that the formulation is achieved by the use of nodal integration with stabilization. These results paved the way for the following step.

7.3 Meshfree co-rotational formulation

A meshfree method within a co-rotational formulation is formulated for the case of plane stress with allowance for the material model of $J2$ plasticity. Inclusion of the co-rotational formulation allows the additional benefits of large displacements and rotations, however, keeping the restriction of small strains. The co-rotational formulation successfully solves a set of large displacement and rotation type benchmark problems including snap through and inelasticity. By using a variationally consistent formulation it is demonstrated that an optimum rate of convergence is maintained within the context of nodal integration and stabilization. This extension of the work is successfully accomplished using the more easily implemented maximum-entropy basis functions. This implementation is truly independent of a mesh, since no blending with finite elements is necessary as is the case with MLS shape functions, there is no dependence on a mesh data structure and integration is accomplished without Gaussian quadrature over a background mesh. This implementation allows the solution of problems more directly related to collapse.

7.4 Applications and validation

The developed methodology is applied to model representative real life structural behavior for purposes of further validation. First, the meshfree co-rotational formulation with $J2$ plasticity material behavior is applied to steel wide flange portal frames. The frames are laterally loaded monotonically and cyclically. For each case of loading the frames are modeled with and without constant axial forces applied to the columns. The results are compared to experimental results available in the literature. The numerical simulations indicate that the co-rotational formulation successfully captures stiffness softening for the cases that include the application of constant axial forces. Second, the formulation is applied to beam-column subassemblies extracted from the frame system of a multi-story structure. These subassemblies are tested to simulate the removal of a supporting frame column. As a result the beam column assembly sags due to the applied loads and develops tensile forces associated with catenary behavior. The model results are compared to high fidelity finite element simulations provided in the literature. It is evident from the numerical results that the meshfree co-rotational formulation correctly captures the development of catenary behavior and tension stiffening. Last, the so-called ‘El Zanaty’ portal frame, a common benchmark used in the evaluation of second order analysis simulations, is analyzed by the current research formulation and compared to published results. Although the published results are created using 1D beam elements the numerical results are in good agreement, and once again the method’s ability to capture stiffness softening caused by including large constant axial forces during the frame’s lateral analysis is demonstrated.

7.5 Observations regarding 1D beam elements versus plane stress continuum

A variety of observations were made during the process of modeling I-shaped beam elements in a plane stress continuum and subsequently comparing the results to 1D beam elements. There are two approaches to modeling frame joints with 1D beam elements. One can use rigid joints for the panel zone or somehow try to model the joint behavior with some form of nonlinear material behavior. Regardless of the approach taken it is difficult to achieve the same results with the 1D beam elements and the frame modeled as a plane stress continuum. This should not pose a problem, because in actuality one should not expect the two modeling techniques to match because they are not of the same order of complexity. Furthermore, the biggest single difference in the models is the modeling of the joints. Results can vary widely for the plane stress results depending on how the panel zone of a beam-column joint is modeled. For example, thicknesses specified in the panel zone area or across column webs to simulate stiffeners can make a significant difference in behavior for both elastic and inelastic panel zones. This should not be a surprise since frame deflections are significantly affected by joint rotations. It is for these reasons that great care is required when modeling frames for comparison to other computer simulations or experimental results.

7.6 Future research directions

The current research establishes the feasibility of the method. It now remains to extend the capabilities of the present research. Specifically, the meshfree co-rotational formulation is extendable to three dimensions, interfacing with finite elements, finite strains, dynamics, including impact, arc length control, the case of reinforced concrete and fracture. Furthermore, as indicated in the literature review an advance version of this formulation may also one day be used as a calibration tool for simpler macromodels. Comments on several of these possible research directions are given next.

Interfacing with (co-rotational) finite elements on the meshfree boundary poses no obstacles and is in fact easier with max-ent basis functions than it is with MLS basis functions and the method of continuous blending. It is possible, for example, to directly connect quadrilateral continuum elements to the meshfree boundary and include their contribution to the global stiffness matrix in the standard way. Referring to Figure 7.1, it is worthwhile to research the benefit of using meshfree domains only where necessary and using 1D beam (or continuum) elements elsewhere. This of course is possible within a co-rotational frame work. In the example of the figure a portal frame is considered. The meshfree domain has nodes with translation degrees of freedom. The question that arises is how to connect a standard beam element to the meshfree domain. A possible technique is to attach an 'interface' beam as shown in the figure. This interface beam is actually a series of small beam

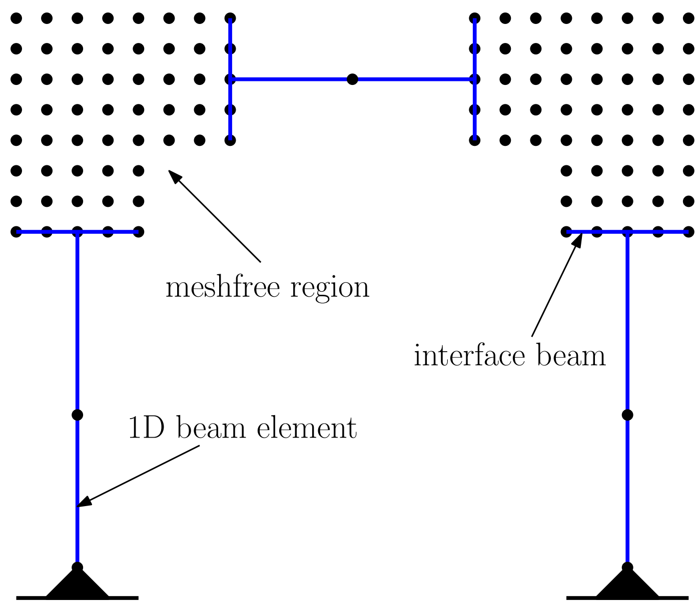


Figure 7.1: Possible scheme for interfacing a mesfree continuum to 1D beam elements.

elements connecting each node across the beam section. Each node of the interface beam has translational degrees of freedom which correspond to the meshfree nodes' translational degrees of freedom. However, the interface beam rotation degrees of freedom are free. This interface beam is constructed with very large flexural stiffness but at the same time has an axial stiffness such that it does not inappropriately restrain the meshfree cross-section. The 1D beam element and the interface beam are then connected by both translation and rotation degrees of freedom. Thus the 1D beam element axial, shear and moment are all transferred to the continuum region through the interface beam. It seems as though this scheme is easily implementable and it is just an issue of assembling the local stiffness matrix degrees of freedom at the correct locations in the global stiffness matrix.

It is uncertain exactly how, for example, 1D beam elements are connected to nodes on the interior of the meshfree domain. Perhaps some form of penalty method or method of Lagrange multipliers is a possible direction of research to accomplish this task. In fact, one example, of connecting (quadrilateral) finite elements used as rebar in a meshfree concrete continuum using a Lagrange multiplier method, is given by Rabczuk and Belytschko [73]. The case of a concrete continuum with co-rotational beam elements acting as rebar is also an interesting case to consider. It is foreseeable that an advanced research code can be constructed so that the meshfree concrete continuum could fracture and separate exposing rebar modeled with 1D co-rotational beam elements with the capability of capturing buckling. Before such a case is considered perhaps a simpler intermediate case as explained next is possible.

The research completed in this dissertation is now possible to extend to the case of multiple material regions. In fact, for several of the examples in the validation chapter yield and ultimate stress values differed between the beams and columns. It is a fairly simple matter to allow for different materials such as concrete and steel. It is interesting to consider the case of reinforced concrete for example. This is possible to do using the already existing $J2$ plasticity model for a line of meshfree nodes in the continuum with correctly specified thickness to mimic the steel rebar. For nodes designated as concrete a simple first approach is to have a separate concrete material model based on a modified $J2$ plasticity. The modified $J2$ plasticity model is constructed as follows. The principle back stress variables $\tilde{\boldsymbol{\beta}}$ are initially set so that the center of stress space is shifted toward the compression stress region. This is equivalent to giving a one-time initial Bauschinger effect to the yield surface. By proper adjustment of the back stress variable an approximate perfectly plastic concrete yield surface could be set to have an appropriate compressive strength and a low tensile capacity. In Figure 7.2, for the case of plane stress, an example concrete yield surface based on this idea is shown in principle space.

In future work it is worthwhile to implement the various research directions within an object oriented programming framework. A major aspect of implementation is the issue of bookkeeping, which is a daunting task. Object oriented programming is uniquely con-

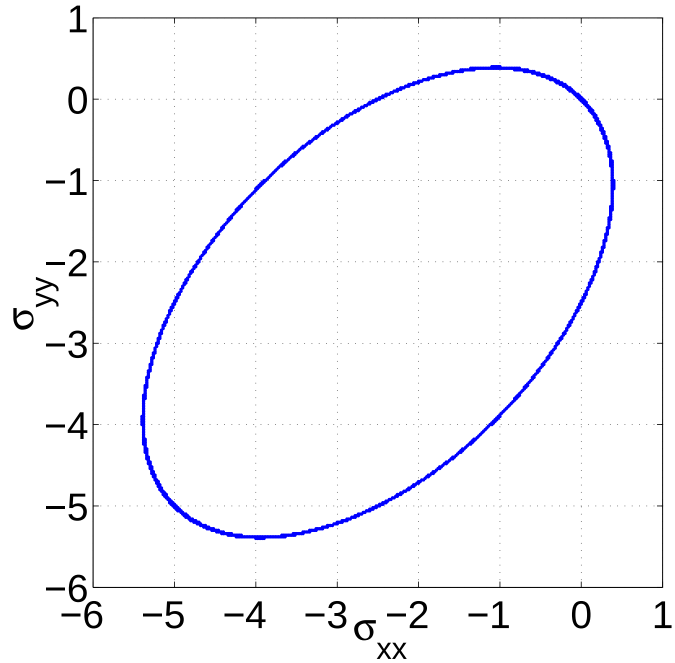


Figure 7.2: Scheme for approximate concrete yield surface in principle stress space (ksi), by using modified $J2$ plasticity model.

structed to assist with these types of problems and also allows for easier future modifications and additions to the research code. The only limitation to this is that researchers need to invest a certain amount of time becoming well-versed in the methods of object oriented programming.

7.7 Summary of most noteworthy results

In conclusion, some of the most noteworthy results and observations are given. First, at the present time it appears that maximum-entropy basis functions allow for the easiest meshfree implementation once robust basis function generation routines are constructed. Unless better and more computationally efficient basis functions are discovered, these are the basis functions recommended. Second, mesh distortion limitations are removed by nodally integrating the weak form. Third, all forms of nodal integration currently available require some form of stabilization to avoid hourglass modes, shear locking, or spurious low energy modes in an eigen analysis. The stabilization recently given in the literature by Puso et al. [72] and implemented as part of this dissertation has proven to be very effective. Fourth, a meshfree co-rotational formulation using nodal integration and stabilization is implemented and applied to a variety of benchmark problems. Within a co-rotational formulation the inclusion of inelastic material behavior has proven to be straightforward. Furthermore, it is demonstrated that a variationally consistent co-rotational formulation is essential to obtaining an optimum rate of convergence during Newton-Raphson iterations for global equilibrium. Fifth, the implemented method successfully solves problems of practical importance as shown in the chapter of applied validation problems. Specifically, the behavior of snap through, stiffness softening and tension stiffening is captured within a co-rotational framework. Last, the feasibility and success of the meshfree co-rotational formulation provides many possible research directions for the advancement of collapse simulation.

References

[1] B. N. Alemdar. *Distributed plasticity analysis of steel building structural systems*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2001.

[2] B. N. Alemdar and D. W. White. Displacement, flexibility, and mixed beamcolumn finite element formulations for distributed plasticity analysis. *Journal of Structural Engineering, ASCE*, 131(12):1811–1819, 2005.

[3] J. H. Argyris. An excursion into large rotations. *Computer Methods in Applied Mechanics and Engineering*, 32:85–155, 1982.

[4] J. H. Argyris, H. Balmer, J. S. Doltsinis, P.C. Dunne, M. Haase, M. Kleiber, G. A. Malejannakis, H. P. Mlejenek, M. Müller, and D. W. Scharpf. Finite element method–the natural approach. *Computer Methods in Applied Mechanics and Engineering*, 17/18:1–106, 1979.

[5] M. Arroyo and M. Ortiz. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, 65(13):2167–2202, 2006.

[6] S. N. Atluri, J.Y. Cho, and H.G. Kim. Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, 2:334–347, 1999.

[7] S. N. Atluri and S. Shen. *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Encino, CA, 1st edition, 2002.

[8] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[9] J. F. Baker and J. W. Roderick. Tests on full-scale portal frames. *Proceedings of the Institution of Civil Engineers*, 1:71–94, 1952. Part 1.

[10] Y. Bao, S. K. Kunnath, S. El-Tawil, and H. S. Lew. Macromodel-based simulation of progressive collapse: RC frame structures. *Journal of Structural Engineering, ASCE*, 134(7):1079–1091, 2008.

[11] K. J. Bathe and S. Bolourchi. Large displacement analysis of three-dimensional beam structures. *International Journal for Numerical Methods in Engineering*, 14:961–986, 1979.

[12] J. Battini and C. Pacoste. Plastic instability of beam structures using co-rotational elements. *Computer Methods in Applied Mechanics and Engineering*, 191:5811–5831, 2002.

[13] L. S. Beedle and R. Christopher. Tests of steel moment connections. *AISC Engineering Journal*, pages 116–125, 1964.

[14] T. Belytschko and L. P. Binderman. Assumed strain stabilization of the eight node hexahedral element. *Computer Methods in Applied Mechanics and Engineering*, 105:225–260, 1993.

[15] T. Belytschko and B. J. Hsieh. Non-linear transient finite element analysis with convected coordinates. *International Journal for Numerical Methods in Engineering*, 7:255–271, 1973.

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

[17] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley and Sons, Chichester, England, 1st edition, 2000.

[18] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[19] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, Cambridge, UK, 2nd edition, 2008.

[20] J. S. Chen, C. T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[21] J. S. Chen, S. Yoon, and C. T. Wu. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53:2587–2615, 2002.

[22] N. H. Christ, R. Friedberg, and T. D. Lee. Weights of links and plaquettes in a random lattice. *Nuclear Physics B*, 210(3):337–346, 1982.

[23] M. J. Clarke and G. J. Hancock. A study of incremental-iterative strategies for non-linear analyses. *International Journal for Numerical Methods in Engineering*, 29:1365–1391, 1990.

[24] A. Conci. Stiffness matrix for nonlinear analysis of thin-walled frames. *Journal of Engineering Mechanics, ASCE*, 118(9):1859–1875, 1992.

[25] M. A. Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Computer Methods in Applied Mechanics and Engineering*, 181:131–150, 1990.

[26] M. A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures – Vol 1 Essentials*. John Wiley & Sons Ltd., Chichester, England, 1991.

[27] M. A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures – Vol 2 Advanced Topics*. John Wiley & Sons Ltd., Chichester, England, 1997.

[28] M. A. Crisfield and G. F. Moita. A co-rotational formulation for 2-d continua including incompatible modes. *International Journal for Numerical Methods in Engineering*, 39:2619–2633, 1996.

[29] M. A. Crisfield and G. F. Moita. A finite element formulation for 3-d continua using the co-rotational technique. *International Journal for Numerical Methods in Engineering*, 39:3775–3792, 1996.

[30] M. A. Crisfield and J. Willis. Solution strategies and softening materials. *Computer Methods in Applied Mechanics and Engineering*, 66(3):267–289, 1988.

[31] L. A. Crivelli and C. A. Felippa. A three-dimensional non-linear timoshenko beam based on the core-congruential formulation. *International Journal for Numerical Methods in Engineering*, 36:3647–3673, 1993.

[32] B. M. Donning and W. K. Liu. Meshless methods for shear deformable beams and plates. *Comput. Methods Appl. Mech. Engrg.*, 152(1–2):47–71, 1997.

[33] M. H. El-Zanaty, D. W. Murray, and R. Bjorhovde. Inelastic behavior of multistory steel frames. Structural Engineering Report 83, University of Alberta, Canada, 1980.

[34] M. Ettouney, R. Smilowitz, M. Tang, and A. Hapij. Global system considerations for progressive collapse with extensions to other natural and man-made hazards. *Journal of Performance of Constructed Facilities*, 20(4):403–417, 2006.

[35] S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Comput. Methods Appl. Mech. Engrg.*, 193:1257–1275, 2004.

[36] T. P. Fries and H. G. Matthies. Classification and overview of meshfree methods. Technical Report Informatikbericht-Nr. 2003-03, Institute of Scientific Computing, Technical University Braunschweig, Braunschweig, Germany, 2004.

[37] M. R. Gosz. *Finite Element Method – Applications in Solids, Structures, and Heat Transfer*. CRC Press, Boca Raton, Florida, 2006.

[38] J. O. Hallquist. *LS-DYNA Theory Manual*. Livermore Software Technology Corporation, Livermore, California, 2006.

[39] E. Hansen, H. Levine, D. Lawyer, and D. Tennant. Computational failure analysis of reinforced concrete structures subjected to blast loading. In *17th Analysis and Computation Specialty Conference*, pages 1–13, St. Louis, Missouri, 2006. ASCE.

[40] A. Huerta and S. Fernández-Méndez. A comparison of two formulations to blend finite elements and mesh-free methods. *Comput. Methods Appl. Mech. Engrg.*, 193:1257–1275, 2004.

[41] D. Isobe and M. Tsuda. Seismic collapse analysis of reinforced concrete framed structures using the finite element method. *Earthquake Engineering and Structural Dynamics*, 32(13):2027–2046, 2003.

[42] G. Jelenić and M. Saje. A kinematically exact space finite strain beam model - finite element formulation by generalized virtual work principle. *Computer Methods in Applied Mechanics and Engineering*, 120:131–161, 1995.

[43] P. H. Jetteur and S. Cescotto. A mixed finite element for the analysis of large ineleastic strains. *International Journal for Numerical Methods in Engineering*, 31:229–239, 1991.

[44] B. G. Johnston, C. H. Yang, and L. S. Beedle. Progress report no. 8, an evaluation of plastic analysis as applied to structural design. *The Welding Journal*, 32:224s–239s, 1953. Research Supplement.

[45] G. Kaewkulchai and E. B. Williamson. Modeling the impact of failed members for progressive collapse analysis of frame structures. *Journal of Performance of Constructed Facilities*, 20(4):375–383, 2006.

[46] K. Khandelwal and S. El-Tawil. Collapse behavior of steel special moment resisting frame connections. *Journal of Structural Engineering, ASCE*, 133(5):646–655, 2007.

[47] K. Khandelwal, S. El-Tawil, S. K. Kunnath, and H. S. Lew. Macromodel-based simulation of progressive collapse: Steel frame structures. *Journal of Structural Engineering, ASCE*, 134(7):1070–1078, 2008.

[48] P. Khosravi, R. Ganesan, and R. Sedaghati. Corotational non-linear analysis of thin plates and shells using a new shell element. *International Journal for Numerical Methods in Engineering*, 69:859–885, 2007.

[49] M. Kojić and K. J. Bathe. *Inelastic analysis of solids and structures*. Springer-Verlag, New York, NY, 2005.

[50] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[51] N. S. Lee and K. J. Bathe. Error indicators and adaptive remeshing in large deformation finite element analysis. *Finite Elements in Analysis and Design*, 16(2):99–139, 1994.

[52] G. Li and J. Li. *Advanced analysis and design of steel structures*. Wiley, Chichester, England, 1st edition, 2007.

[53] S. Li and W. K. Liu. Meshfree and particle methods and their applications. *Appl Mech Rev*, 55(1):1–34, 2002.

[54] Z. X. Li. A co-rotational formulation for 3d beam element using vectorial rotational variables. *Computational Mechanics*, 39:309–322, 2007.

[55] Z. X. Li. A mixed co-rotational formulation of 2d beam elements using vectorial rotational variables. *Communications in Numerical Methods in Engineering*, 23:45–69, 2007.

[56] R. J. Y. Liew, D. W. White, and W. F. Chen. Second-order refined plastic-hinge analysis for frame design. part i. *Journal of Structural Engineering, ASCE*, 119(11):3196–3216, 1993.

[57] R. J. Y. Liew, D. W. White, and W. F. Chen. Second-order refined plastic-hinge analysis for frame design. part ii. *Journal of Structural Engineering, ASCE*, 119(11):3217–3237, 1993.

[58] W. K. Liu, Y. Guo, S. Tang, and T. Belytschko. A multiple-quadrature eight-node hexahedral finite element for large deformation elastoplastic analysis. *Computer Methods in Applied Mechanics and Engineering*, 154:69–132, 1998.

[59] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.

[60] B. M. Luccioni, R.D. Ambrosini, and R.F. Danesi. Analysis of building collapse under blast loads. *Engineering Structures*, 26:6371, 2004.

[61] S. Marjanishvili and E. Agnew. Comparison of various procedures for progressive collapse analysis. *Journal of Performance of Constructed Facilities*, 20(4):365374, 2006.

[62] A. Masud, C.L. Tham, and W.K. Liu. A stabilized 3-d co-rotational formulation for geometrically nonlinear analysis of multi-layered composite shells. *Computational Mechanics*, 26:1–12, 2000.

[63] K. Mattiasson. *On the co-rotational finite element formulation for large deformation problems*. PhD thesis, Chalmers University of Technology, Goteborg, Sweden, 1983.

[64] K. Mattiasson, A. Bengtsson, and A. Samuelsson. *On the accuracy and efficiency of numerical algorithms for geometrically nonlinear structural analysis, in: P.G. Bergan, K.J. Bathe and W. Wunderlich, eds., Finite Element Methods for Nonlinear Problems*. Springer, 1986.

[65] S. Mazzoni, F. McKenna, M. H. Scott, and G. L. Fenves. *OpenSees Command Language Manual*. The Regents of the University of California, http://opensees.berkeley.edu/index.php, 2007.

[66] W. McGuire, R. H. Gallagher, and R. D. Ziemian. *Matrix structural analysis*. Wiley, Chichester, England, 2nd edition, 2000.

[67] O. A. Mohamed. Progressive collapse of structures: Annotated bibliography and comparison of codes and standards. *Journal of Performance of Constructed Facilities*, 20(4):418–425, 2006.

[68] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse appoximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[69] A. Neuenhofer and F. C. Filippou. Evaluation of nonlinear frame finite-element models. *Journal of Structural Engineering, ASCE*, 123(7):958–966, 1997.

[70] American Institute of Steel Construction. *Manual of steel construction - Load and resistance factor design.* AISC, Inc., Chicago, 3rd edition, 2001.

[71] A. Phillips. *Introduction to Plasticity.* The Ronald Press Co., New York, 1956.

[72] M. A. Puso, J. S. Chen, E. Zywicz, and W. Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, 74:416–446, 2008.

[73] T. Rabczuk and T. Belytschko. Application of particle methods to static fracture of reinforced concrete structures. *International Journal of Fracture*, 137:19–49, 2006.

[74] M. Schulz and F. C. Fillipou. Non-linear spatial timoshenko beam element with curvature interpolation. *International Journal for Numerical Methods in Engineering*, 50:761–785, 2001.

[75] M. H. Scott and G. L. Fenves. Plastic hinge integration methods for force–based beam-column elements. *Journal of Structural Engineering, ASCE*, 132(2):244–252, 2005.

[76] R. Sibson. A vector identity for the Dirichlet tesselation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151–155, 1980.

[77] J. C. Simo. A finite strain beam formulation. the three-dimensional dynamic problem. part i. *Computer Methods in Applied Mechanics and Engineering*, 49:55–70, 1985.

[78] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity.* Springer-Verlag, New York, 1998.

[79] J. C. Simo, J. Oliver, and F. Armero. An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids. *Computational Mechanics*, 12(5):267–327, 1993.

[80] J. C. Simo and R. L. Taylor. Return mapping algorithm for plane stress elastoplasticity. *International Journal for Numerical Methods in Engineering*, 22:649–670, 1986.

[81] J. C. Simo and L. Vu-Quoc. A three-dimensional finite-strain rod model. part ii: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 58:79–116, 1986.

[82] M. Sivaselvan and A. M. Reinhorn. Collapse analysis: Large inelastic deformations analysis of planar frames. *Journal of Structural Engineering, ASCE*, 128(12):1575–1583, 2002.

[83] Y. Suetake. Element-free method based on Lagrange polynomial. *Journal of Engineering Mechanics*, 128(2):231–239, 2002.

[84] N. Sukumar. Construction of polygonal interpolants: A maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61(12):2159–2181, 2004.

[85] N. Sukumar and R. W. Wright. Overview and construction of meshfree basis functions: From moving least squares to entropy approximants. *International Journal for Numerical Methods in Engineering*, 70(2):181–205, 2007.

[86] F. F. Taucer, E. Spacone, and G. L. Fenves. A fiber beam–column element for seismic response analysis of reinforced concrete structures. *UCB/EERC-91/17, Earthquake Engineering Research Center, University of California, Berkeley*, 1991.

[87] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill, New York, 2nd edition, 1951.

[88] S. Toma and W. F. Chen. Calibration frames for second-order inelastic analysis in Japan. *Journal of Constructional Steel Research*, 28(1):51–77, 1994.

[89] S. Toma, W. F. Chen, and D. W. White. A selection of calibration frames in North America for second-order inelastic analysis. *Engineering Structures*, 17(2):104–112, 1995.

[90] M. A. M. Torkamani and M. Sonmez. Inelastic large deflection modeling of beam-columns. *Journal of Structural Engineering, ASCE*, 127(8):876–887, 2001.

[91] Y. Urthaler and J. N. Reddy. A co-rotational finite element formulation for the analysis of planar beams. *Communications in Numerical Methods in Engineering*, 21:553–570, 2005.

[92] R. Villaverde. Methods to assess the seismic collapse capacity of building structures: State of the art. *Journal of Structural Engineering, ASCE*, 133(1):57–66, 2007.

[93] E. Voce. A practical strain hardening function. *Metallurgia*, 51(307):219–226, 1955.

[94] R. Weitzmann. Limit state design of hybrid structures with meshless methods using mathematical optimization. Bauhaus-University Weimar, Institute of Structural Engineering, 2004. `http://e-pub.uni-weimar.de/volltexte/2004/7/pdf/M_113.pdf`.

[95] G. Wempner. Finite elements, finite rotations and small strains of flexible shells. *Int. J. Solids Struct.*, 5:117–153, 1969.

[96] D. W. White. Material and geometric nonlinear analysis for local planar behavior in steel frames. Master's thesis, Cornell University, Ithaca, NY, 1985.

[97] T. X. Yu and L. C. Zhang. *Plastic Bending - Theory and Applications*. World Scientific Publishing, Singapore, 1996.

[98] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Solid and Structural Mechanics*. Butterworth-Heinemann, Oxford, 6th edition, 2005.

Appendix A

Moving Least Squares Terminology and Alternative Derivation

A.1 Some relevant terminology.

Some background terminology is helpful to understand the following sections. The terminology here is consistent with terminology often used in the literature.

m – The order of the polynomial base.

Polynomial base – A column vector, $\mathbf{p}$, of a *complete* polynomial of order m. For example, for a 1D linear (order m=1) polynomial base $\mathbf{p}^T = [1\ x]$, for a 1D quadratic (order m=2) polynomial base $\mathbf{p}^T = [1\ x\ x^2]$, and similarly for higher order polynomials. In each case these examples give a complete polynomial base since no terms of the polynomial, from order 0 to m, are missing. For a 2D polynomial (order m=1) linear base $\mathbf{p}^T = [1\ x\ y]$, for quadratic (order m=2) $\mathbf{p}^T = [1\ x\ y\ x^2\ xy\ y^2]$.

k – The number of terms in the polynomial base. When programming the moving least squares shape functions it is useful to know how many terms are in the mth order polynomial base vector. The following table gives the relationship between m and k for 1, 2 or 3 dimensions.

Dimensions	k
1D	$m+1$
2D	$(m+1)(m+2)/2$
3D	$(m+1)(m+2)(m+3)/6$

Table A.1: Relationship between m and k based on the number of dimensions.

ϕ_a – A shape (or basis) function associated with a particular point a in the domain of the problem.

$\boldsymbol{x}$ – A point at which a shape function is evaluated. It is just a number in 1D, and a vector in 2D or 3D.

n – The number of points that have nonzero weight function values at evaluation point $\boldsymbol{x}$. These points are a subset of the points used to discretize the entire domain. The points used to discretize the entire domain are the alternative to covering the domain with finite elements.

$\boldsymbol{x}_a$ – A particular point a has its own vector of coordinates $\boldsymbol{x}_a$ each of which are associated with shape function ϕ_a. The collection of points $\boldsymbol{x}_a$, there are n of them, are used to construct the MLS shape functions.

Kronecker-delta property – If the shape functions have the Kronecker-delta property then $\phi_a(x_b) = \delta_{ab}$. In FEM the shape functions have the Kronecker-delta property. However, MLS shape functions generally do not have the Kronecker-delta property. This is the reason that imposition of essential boundary conditions has been difficult in meshfree methods.

Compact Support – In FEM the shape functions are said to have compact support. For instance they have the Kronecker-delta property. So their *support* is limited to a small region around a particular node. MLS shape functions can have compact support also. Generally, even though the Kronecker-delta property is absent for MLS shape functions, the *support* of each MLS shape function is kept compact. This is done by the use of a support radius, ρ_a, for each node a, and the weight function.

Support Radius - Each shape function has a support or zone of influence. Usually this is a circular region in 2D or a sphere in 3D centered on the point associated with the shape function. The radius of this zone of influence is called the support radius, ρ. The radius is usually taken to be the distance to the 2nd nearest neighboring point in 1D and the 3rd or 4th nearest neighboring point in 2D, times a user input parameter, α.

Zone of influence – The domain within which a given shape function is nonzero.

Weight Function – Generally, weight functions are chosen to be the same for all nodes. However, this is not required. A variety of weight functions are available. Each weight function a is constructed so that it equals 1 at the point a and is zero at a distance equal to the support radius away from the point. The 4th order quartic spline weight function is effective and easy to use. With $q = \|\boldsymbol{x} - \boldsymbol{x}_a\|/\rho_a$ the 4th order quartic spline is defined as follows:

$$w(q) \in C^2 = \begin{cases} 1 - 6q^2 + 8q^3 - 3q^4 & q \leq 1 \\ 0 & q > 1 \end{cases} \tag{A.1}$$

Bubnov-Galerkin Method– The common method of using the same shape functions for both the trial functions and the test functions in the weak form of a BVP.

Element Free Galerkin – The meshfree method which makes use of MLS shape functions and solves the system of governing pde's by a Galerkin method (usually the Bubnov-Galerkin Method).

Weighted Least Squares – Each squared term in a traditional least squares summation is multiplied by its own weight function. Hence the influence of each squared term is affected or limited by its associated weight. For instance, a weighted least squares functional (which is explained in more detail later in this appendix) differs from a traditional least squares functional only by the addition of the w_a term in the summation as follows:

$$J(\mathbf{g}) = \frac{1}{2}\sum_{a=1}^{n} w_a \left\{\mathbf{p}^T(\boldsymbol{x}_a)\mathbf{g} - u_a\right\}^2 \tag{A.2}$$

Moving Least Squares (MLS) – For a particular fixed point $\boldsymbol{x}$ each w_a is calculated for the weighted least squares functional. The functional J is then minimized with respect to each g_i (i ranges from 1 to k) and solved for the g_i values (contained in $\mathbf{g}$). These g_i values are then dependent on the fixed point used to evaluate the w_a's. Later, it is shown how this minimization procedure is used to determine the value of the MLS shape functions at a particular fixed point $\boldsymbol{x}$. Hence every time there is a move to a new point the minimization procedure is repeated. This is why this is a moving least squares procedure, because it is dependent on the current evaluation point $\boldsymbol{x}$.

Moment Matrix – In the literature this is often denoted as the A matrix. This matrix is defined during the derivation of the MLS shape functions given below.

Interpolation property – In FEM the approximation is of the form $u^h = \Sigma\phi_a(\boldsymbol{x}_b)u_a$, so that when b is equal to a, $u^h = u_b$ due to the Kronecker-delta property. This is the interpolating property of FEM shape functions. However, MLS shape functions generally do not have the Kronecker-delta property and hence are not interpolating. Hence more than one shape function may be nonzero at each node. Another way of saying this is that the values u_a are not nodal values like they are in FEM, hence they are sometimes called nodal coefficients instead.

Partition of unity – If the MLS shape functions are constructed correctly then the shape functions evaluated at any given point $\boldsymbol{x}$ sum to 1. This is a good way to numerically check if the shape functions are constructed correctly. If shape functions are constructed correctly this criteria is satisfied to machine precision. The partition of unity property is stated mathematically as follows:

$$\sum_{a=1}^{n} \phi_a(\boldsymbol{x}) = 1 \tag{A.3}$$

Partition of nullity – If the derivative of the MLS shape functions are constructed correctly then the shape function derivatives evaluated at any given point $\boldsymbol{x}$ sum to 0. This is a good way to numerically check if the derivatives are constructed correctly. If the derivatives are constructed correctly this criteria is satisfied very close to machine precision:

$$\sum_{a=1}^{n} \nabla\phi_a(\boldsymbol{x}) = \mathbf{0} \tag{A.4}$$

Consistency – The highest polynomial order which is represented exactly. For example, MLS shape functions constructed with a linear polynomial base have 1st order consistency. That is, if a differential equation has a linear solution then the MLS shape functions can represent the solution exactly. Another example, consider an mth order polynomial base $\mathbf{p}(\boldsymbol{x})$ and n points at which MLS shape functions are constructed. So to each point $\boldsymbol{x}_a$ (a vector in 2 and 3 dimensions) there is a corresponding shape function ϕ_a. Then, with mth order consistency, the following is true:

$$\sum_{i=a}^{n} \phi_a(\boldsymbol{x})\mathbf{p}(\boldsymbol{x}_a) = \mathbf{p}(\boldsymbol{x}), \quad \forall\boldsymbol{x} \in \Omega \tag{A.5}$$

A.2 Moving least squares (MLS) shape functions derivation

The derivation given is for 1D problems (hence the x variable is non bold in the following, the derivation is easily extended to 2 or 3 dimensions merely by making x bold as necessary to reflect its vector character).

Consider the task of finding an approximate solution $u^h(x)$, while knowing the true solution, u_a at selected points x_a. Then in a least squares sense minimization of the expression $[u^h(x_a) - u_a]^2$ for each a is the objective. Suppose a polynomial approximation is chosen so that

$$u^h(x) = g_1 + g_2 x + g_3 x^2 + \ldots g_{m+1} x^m. \tag{A.6}$$

The approximation is then written in matrix form

$$u^h(x) = \mathbf{p}^T(x)\mathbf{g} = \{1 \; x \; x^2 \; \ldots x^m\} \left\{ \begin{array}{c} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_{m+1} \end{array} \right\}. \tag{A.7}$$

Using the above the least squares functional is written with the approximation substituted in for $u^h(x)$ (the 1/2 in front is added for mathematical convenience):

$$J = \frac{1}{2} \sum_{a=1}^{n} \left\{ u^h(x_a) - u_a \right\}^2 = \frac{1}{2} \sum_{a=1}^{n} \left\{ \mathbf{p}^T(x_a)\mathbf{g} - u_a \right\}^2. \tag{A.8}$$

Now, recall that compact support for each node a is intended. Therefore, the local solution is influenced by the local nodes. Whereas, nodes far away have no influence. Hence, each summation term, indexed by a, in the least squares functional is weighted by a weight function w_a, which limits the term's influence to point a and usually several surrounding nodes. Based on this intuition the functional J is modified and becomes a weighted least squares functional as follows:

$$J = \frac{1}{2}\sum_{a=1}^{n} w_a \left\{\mathbf{p}^T(x_a)\mathbf{g} - u_a\right\}^2. \tag{A.9}$$

Next, it is necessary to minimize J with respect to each g_i. However, before this operation, it is helpful to first write the functional in matrix form. To this end, the functional J is written as follows:

$$J = \frac{1}{2}(\mathbf{Pg} - \mathbf{u})^T\mathbf{W}(\mathbf{Pg} - \mathbf{u}), \tag{A.10}$$

where,

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \dots & p_k(x_1) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \dots & p_k(x_n) \end{bmatrix}. \tag{A.11}$$

Notice that each row of the $\mathbf{P}$ matrix is just $\mathbf{p}^T(x_a)$ for each row a. And, this matrix is n by k in size.

For $\mathbf{W}$, an n by n matrix results (see Eq. (A.1) for the definition of diagonal terms w_a)

$$\mathbf{W} = \begin{bmatrix} w_1(x - x_1) & & & 0 \\ & w_2(x - x_2) & & \\ & & \ddots & \\ 0 & & & w_n(x - x_n) \end{bmatrix}. \tag{A.12}$$

The $\mathbf{g}$ vector is k by 1 and the $\mathbf{u}$ vector is n by 1.

Now, set $\frac{\partial J}{\partial \mathbf{g}} = \mathbf{0}$. This yields the following:

$$(\mathbf{Pg} - \mathbf{u})^T \mathbf{WP} = \mathbf{0}. \tag{A.13}$$

Transposing the whole equation yields

$$(\mathbf{WP})^T (\mathbf{Pg} - \mathbf{u}) = \mathbf{0}. \tag{A.14}$$

Multiplying through gives

$$\mathbf{P}^T \mathbf{WPg} - \mathbf{P}^T \mathbf{Wu} = \mathbf{0} \tag{A.15}$$

and finally

$$\mathbf{P}^T \mathbf{WPg} = \mathbf{P}^T \mathbf{Wu}. \tag{A.16}$$

Now define the moment matrix $\mathbf{A} = \mathbf{P}^T \mathbf{WP}$ and $\mathbf{B} = \mathbf{P}^T \mathbf{W}$. Note that $\mathbf{A}$ is k by k and $\mathbf{B}$ is k by n. Using these definitions Eq. (A.16) becomes

$$\mathbf{Ag} = \mathbf{Bu}. \tag{A.17}$$

Solve now for the unknown coefficients $\mathbf{g}$

$$\mathbf{g} = \mathbf{A}^{-1} \mathbf{Bu}. \tag{A.18}$$

Substitute this into the first part of Eq. (A.7)

$$u^h = \mathbf{p}^T(x)\mathbf{g} = \mathbf{p}^T(x)\mathbf{A}^{-1}\mathbf{Bu}. \tag{A.19}$$

The approximations u^h are usually written as

$$u^h = \boldsymbol{\phi}^T \mathbf{u} = \sum_{a=1}^{n} \phi_a u_a. \tag{A.20}$$

Comparison of Eq. (A.20) with Eq. (A.19) reveals that the vector of MLS shape functions is

$$\boldsymbol{\phi}^T = \mathbf{p}^T(x)\mathbf{A}^{-1}\mathbf{B}. \tag{A.21}$$

Notice that the $\mathbf{A}$ and $\mathbf{B}$ matrices depend on $\mathbf{W}$. The $\mathbf{W}$ matrix in turn is a function of the x_a and the evaluation point x. Hence every time a new evaluation point x is chosen the matrices $\mathbf{A}^{-1}$ and $\mathbf{B}$ are recomputed to calculate the MLS shape functions based on the equation $\boldsymbol{\phi}^T = \mathbf{p}^T(x)\mathbf{A}^{-1}\mathbf{B}$.

A.3 MLS shape function characteristics

The properties of MLS shape functions are different then FEM shape functions in the following ways:

- MLS shape functions do not have the Kronecker-delta property
- MLS shape functions are not known in closed form
- MLS shape functions are non-interpolating
- The size of support can be controlled by the radius of support parameter, ρ. It is common to set ρ equal to the distance to the 3rd or 4th nearest neighboring point (in 2D for example) and adjust this as necessary by a parameter called α. This parameter is a user specified input when constructing MLS shape functions. It is sometimes necessary to adjust the radius of support by use of α to eliminate a singular moment

matrix **A** when insufficient nodes are included within the support radius.

- More computational effort is required to calculate the MLS shape functions.
- Because the MLS shape functions do not have the Kronecker-delta property special attention must be paid to enforcing essential boundary conditions during the solution of a boundary value problem.

A.4 MLS shape function derivatives

Shape function derivatives are calculated by application of the product rule on Eq. (A.21). For example, if the derivative of $\boldsymbol{\phi}$ is taken with respect to the kth dimension (could be x, y or z)

$$\boldsymbol{\phi}^T_{,k}(\boldsymbol{x}) = \mathbf{p}^T_{,k}\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^T\mathbf{A}^{-1}_{,k}\mathbf{B} + \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B}_{,k} \tag{A.22}$$

with $\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}$. This last expression is found as follows. Take the derivative of Eq. (A.17) to get

$$\mathbf{A}_{,k}\mathbf{g} + \mathbf{A}\mathbf{g}_{,k} = \mathbf{B}_{,k}\mathbf{u}. \tag{A.23}$$

Solving Eq. (A.23) for $\mathbf{g}_{,k}$ and substituting in Eq. (A.18) for **a** yields

$$\mathbf{g}_{,k} = \mathbf{A^{-1}}\mathbf{B}_{,k}\mathbf{u} - \mathbf{A^{-1}}\mathbf{A}_{,k}\mathbf{A^{-1}}\mathbf{Bu}. \tag{A.24}$$

Now take the derivative of Eq. (A.18), which is

$$\mathbf{g}_{,k} = \mathbf{A^{-1}}\mathbf{B}_{,k}\mathbf{u} + \mathbf{A}^{\mathbf{-1}}_{,\mathbf{k}}\mathbf{Bu}. \tag{A.25}$$

Finally compare Eq. (A.24) and Eq. (A.25) and observe that they are the same except for the coefficient of the **Bu** term. Hence the coefficients of this term must be equivalent.

Therefore $\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}$, which completes the derivation. Second derivatives are found similarly, see Fries and Matthies [36, page 21].

Appendix B

Derivation of Analytical Solution for Elasto-Plastic Rectangular and I-shaped Beams

B.1 Introduction

Analytic solutions for linear structural mechanics problems are plentiful. However, analytical nonlinear solutions of structural mechanics problems are rare if not impossible to find. If such solutions are found they are helpful for validation of nonlinear numerical solutions provided by finite difference, finite element or meshfree methods for example. In the following, an analytical solution of an elementary Euler-Bernoulli cantilever beam with a bilinear material model is presented. The presentation illustrates the solution for a rectangular cross-section and then extends the results to the case of an I-beam type cross-section. First, a review of the elementary equations is given.

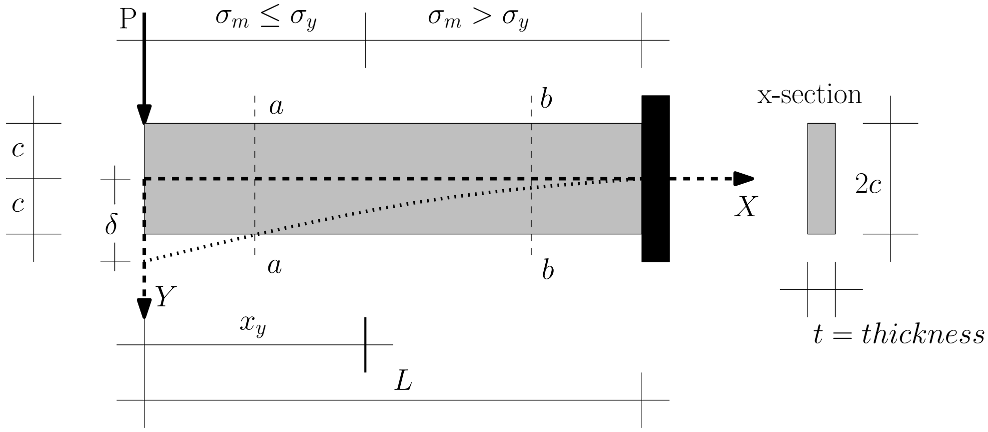


Figure B.1: Cantilever beam.

B.2 Rectangular cantilever beam – elasto-plastic analytical equations

In the development of the elementary equations use is made of the geometry, loading and coordinate system, for a standard cantilever beam, as given in Figure B.1. Referring to this figure, when load P is large enough to cause yielding, there are two distinct regions along the length of the cantilever. The cross-sections not yet yielded are on the left ($x \leq x_y$) and cross-sections which have yielded are on the right ($x > x_y$). The x-distance beyond which yield is taking place is denoted by x_y.

First, define the internal moment as caused by the external load P as a function of x. This equation is valid for all cross-sections of the beam.

$$M_e(x) = Px \tag{B.1}$$

Next, consider the first region ($x \leq x_y$). For all sections $a - a$, as cut in Figure B.1, the maximum stress is

$$\sigma_m(x) = \frac{M_e c}{I} = \frac{3Px}{2tc^2} \tag{B.2}$$

and maximum strain is

$$\varepsilon_m(x) = \frac{\sigma_m}{E} = \frac{3Px}{2Etc^2}. \tag{B.3}$$

Proceed now to the second region ($x > x_y$). Find the expression for x_y by solving Eq. (B.2) for x and setting $\sigma_m = \sigma_y$.

$$x_y = \frac{2\sigma_y tc^2}{3P} \tag{B.4}$$

Now, considering the second region, a bilinear material model is chosen as shown in Figure B.2. For the elastic range a modulus of elasticity E and for the plastic range a hardening modulus $\bar{H}$ is chosen.

Now consider a typical section $b - b$ as cut in Figure B.1 and refer to the stress and strain profiles for section $b - b$ as given in Figure B.3. Note that in Figure B.3 an as yet unknown parameter α is defined. In terms of the parameter α, by similar triangles, an expression for the maximum strain is written as follows:

$$\varepsilon_m = \frac{c}{\alpha c}\varepsilon_y = \frac{\varepsilon_y}{\alpha}. \tag{B.5}$$

Using Eq. (B.5) and referring to the stress response of Figure B.2 the expression for σ_m is

$$\sigma_m(\alpha) = \sigma_y + (\varepsilon_m - \varepsilon_y)\bar{H} = \sigma_y + \left(\frac{\varepsilon_y}{\alpha} - \varepsilon_y\right)\bar{H} = \sigma_y + \left(\frac{1-\alpha}{\alpha}\right)\varepsilon_y\bar{H}. \tag{B.6}$$

Using Figure B.3 write an expression for the internal cross-section moment due to the stress profile. The expression is found by summing moments about the neutral axis of the cross-section.

$$M_i(\alpha) = 2t\left[\frac{1}{2}\sigma_y\alpha c\left(\frac{2}{3}\alpha c\right) + \frac{\sigma_m(\alpha)+\sigma_y}{2}(1-\alpha)c\left(\alpha c + \frac{[2\sigma_m(\alpha)+\sigma_y]b}{3[\sigma_m(\alpha)+\sigma_y]}\right)\right]. \tag{B.7}$$

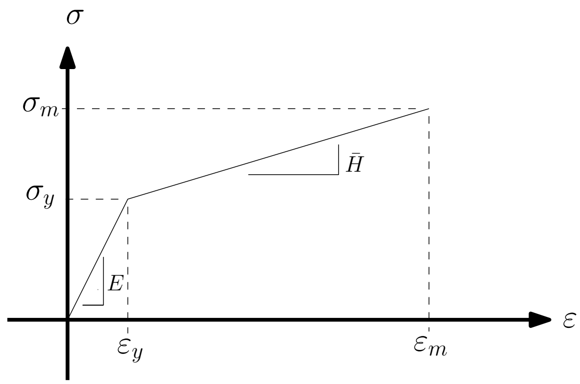


Figure B.2: Bilinear material model.

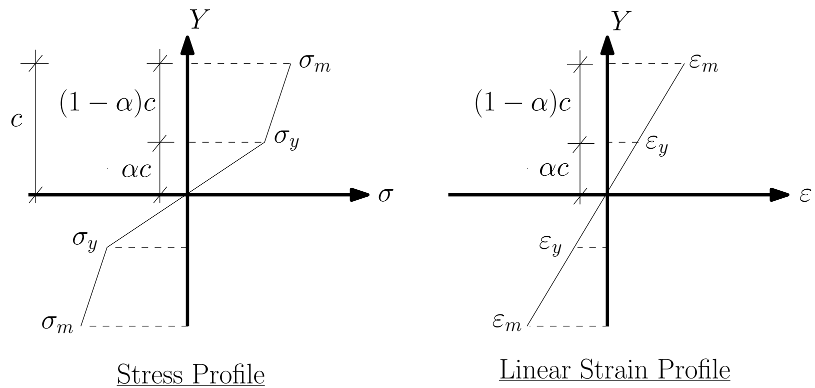


Figure B.3: Section $b-b$.

Equation (B.7) is simplified to give

$$M_i(\alpha) = \frac{c^2 t}{3}[\sigma_m(\alpha)(2-\alpha-\alpha^2) + \sigma_y(1+\alpha)]. \tag{B.8}$$

B.3 Solving the rectangular elasto-plastic cantilever beam problem

Given the equations above the stresses, strains, and deflections consistent with the bilinear material chosen are found. In the elastic region of the cantilever the solution is trivial. In the yielded regions of the beam recognize that the moment M_e, a function of x and P, must be in equilibrium with the moment M_i. This then gives an equation from which the unknown parameter α is solved.

$$M_e(x, P) - M_i(\alpha) = 0 \tag{B.9}$$

Equation (B.9) does have one real positive root but the expression is quite complicated. Instead the solution for this one real positive root is expressed as a function of x and P as follows:

$$\alpha(x, P) = \mathrm{root}(M_e - M_i, \alpha \in [0, 1]). \tag{B.10}$$

For specified x and P the root of the above expression is found numerically. With the solution for α in hand a set of equations is obtained in both regions of the cantilever beam. In the yielded portion of the beam the equations are of course nonlinear. Over the length of the beam the expression for the maximum strain as a function of x and P is given as

$$\varepsilon_m(x, P) = \begin{cases} \dfrac{3Px}{2tc^2E} & x \le x_y \\ \dfrac{\varepsilon_y}{\alpha(x, P)} & x > x_y \end{cases} \tag{B.11}$$

and the expression for $\sigma_m(x,P)$ is similarly expressed as

$$\sigma_m(x,P) = \begin{cases} \dfrac{3Px}{2tc^2} & x \leq x_y \\ \sigma_y + \left(\dfrac{1-\alpha(x,P)}{\alpha(x,P)}\right)\varepsilon_y \bar{H} & x > x_y. \end{cases} \tag{B.12}$$

In order to calculate the nonlinear displacement versus load for the cantilever beam recall from basic beam theory the 2nd Moment Area Theorem and that for small displacements the curvature of the elastic curve is given by ε_m/c. The deflection for the tip of the cantilever as a function of P is then calculated as follows:

$$\delta(P) = \int_0^L x \frac{\varepsilon_m(x,P)}{c}\, dx \tag{B.13}$$

B.4 *I*-beam cantilever – elasto-plastic analytical equations

For the case of an I-beam type cross-section a procedure similar to the rectangular cross-section is followed. When possible, use is made of previously found results. For the I-beam cross-section the notation of Figure B.4 is used. First, for all cross-sections of the beam the internal moment as caused by the external load P (see Figure B.1) is

$$M_e(x) = Px. \tag{B.14}$$

Consider the first the region where $x \leq x_y$ (i.e. no yielding). The maximum stress is

$$\sigma_m(x) = \frac{M_e c}{I_{xx}} = \frac{Pcx}{I_{xx}}, \tag{B.15}$$

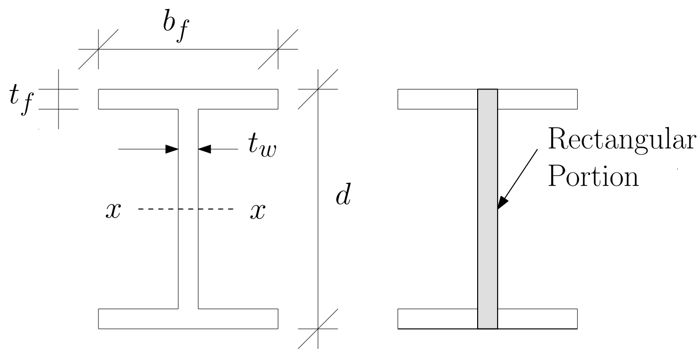


Figure B.4: I-beam cross-section.

where I_{xx} is the moment of inertia about the $x - x$ axis as shown in Figure B.4 and the maximum strain is

$$\varepsilon(x) = \frac{\sigma_m}{E} = \frac{Pcx}{EI_{xx}}. \tag{B.16}$$

Next consider the regions for which $x > x_y$. First, x_y is found by solving Eq. (B.15) for x and setting $\sigma_m = \sigma_y$ which gives

$$x_y = \frac{\sigma_y I_{xx}}{Pc}. \tag{B.17}$$

For the region $x > x_y$ a bilinear material model as shown in Figure B.2 is chosen similar to the case of a rectangular beam. Now consider a typical section $b - b$ as cut in Figure B.1 and refer to the stress and strain profiles of Figure B.3. In terms of α the maximum strain is

$$\varepsilon_m = \frac{c}{\alpha c}\varepsilon_y = \frac{\varepsilon}{\alpha}. \tag{B.18}$$

Upon using Eq. (B.18) and referring to the stress response of Figure B.2 the maximum stress is

$$\sigma_m(\alpha) = \sigma_y + (\varepsilon_m - \varepsilon_y)\bar{H} = \sigma_y + \left(\frac{1-\alpha}{\alpha}\right)\varepsilon_y \bar{H}. \tag{B.19}$$

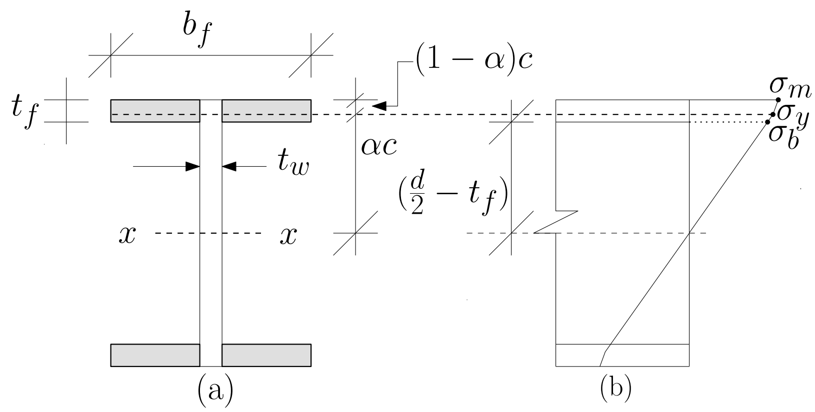


Figure B.5: Case 1: (a) flange area contribution, (b) yield in flanges stress profile.

Referring to Figure B.3 the goal is to write an expression for the internal moment due to the stress profile acting on the I-beam section of Figure B.4. For the rectangular portion of the I-beam cross-section the internal moment found previously (see Eq. (B.8)) based on α is

$$M_i(\alpha) = \frac{c^2 t}{3}[\sigma_m(\alpha)(2 - \alpha - \alpha^2) + \sigma_y(1 + \alpha)]. \tag{B.20}$$

The additional internal moment due to the flanges must be added to the internal moment due to the rectangular portion of the I-beam. In so doing, consideration is made for two cases. For case 1, the unknown parameter, α, is such that $\alpha c > (d/2 - t_f)$ (i.e., the flanges have not yielded through their entire thickness). For case 2, α is of a value such that $\alpha c \leq (d/2 - t_f)$ (i.e., the flanges have yielded through their entire thickness).

Consider first case 1 and refer to Figure B.5. The contributing flange width is

$$b = b_f - t_w. \tag{B.21}$$

The stress at the bottom of the flange is

$$\sigma_b = \frac{(\frac{d}{2} - t_f)\sigma_y}{\alpha c} = \left(\frac{d - 2t_f}{2\alpha c}\right)\sigma_y. \tag{B.22}$$

For the unyielded portion of the flange the average stress is

$$\sigma_{up} = \frac{\sigma_y + \sigma_b}{2} = \frac{2\alpha c\sigma_y + (d - 2t_f)\sigma_y}{4\alpha c} = \left(\frac{2\alpha c + d - 2t_f}{4\alpha c}\right)\sigma_y. \tag{B.23}$$

For the yielded portion of the flange the average stress is

$$\sigma_{yp} = \frac{\sigma_y + \sigma_m}{2}. \tag{B.24}$$

The force of the unyielded portion of the flange is

$$P_{up} = \sigma_{up} b \left(\alpha c - \left(\frac{d}{2} - t_f\right)\right). \tag{B.25}$$

The moment arm of the unyielded portion of the flange is given by

$$\bar{Y}_{up} = \alpha c - \frac{1}{3}\left(\frac{2\sigma_b + \sigma_y}{\sigma_b + \sigma_y}\right)\left(\alpha c - \left(\frac{d}{2} - t_f\right)\right). \tag{B.26}$$

The force of the yielded portion is

$$P_{yp} = \sigma_{yp} b \left(\frac{d}{2} - \alpha c\right). \tag{B.27}$$

The moment arm of the yielded portion of the flange is given by

$$\bar{Y}_{yp} = \alpha c + \frac{1}{3}\left(\frac{2\sigma_m + \sigma_y}{\sigma_m + \sigma_y}\right)\left(\frac{d}{2} - \alpha c\right). \tag{B.28}$$

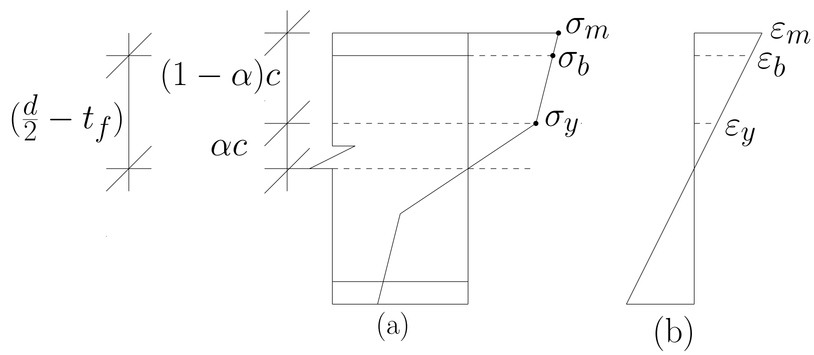


Figure B.6: Case 2: (a) yield below flange stress profile, (b) linear strain profile.

Finally, the expression for the additional moment caused by the flanges (top and bottom flanges) is written as

$$M_i(\alpha)_{\text{flanges case 1}} = 2(P_{up}\bar{Y}_{up} + P_{yp}\bar{Y}_{yp}), \tag{B.29}$$

so that the total internal moment for case 1 is

$$M_i(\alpha) = M_i(\alpha)_{\text{rectangular}} + M_i(\alpha)_{\text{flanges case 1}}. \tag{B.30}$$

Next consider case 2 for which the flanges yield through their entire thickness. Referring to Figure B.6, the stress at the bottom of the flange is written as

$$\sigma_b = \sigma_y + (\varepsilon_b - \varepsilon_y)\bar{H}. \tag{B.31}$$

By use of similar triangles the expression for the strain at the bottom of the flange is

$$\varepsilon_b = \left(\frac{d - 2t_f}{2\alpha c} \right) \varepsilon_y. \tag{B.32}$$

Upon substituting Eq. (B.32) into Eq. (B.31) the stress at the bottom of the flange is

$$\sigma_b = \sigma_y + \left(\left(\frac{d - 2t_f}{2\alpha c} \right) - 1 \right) \varepsilon_y \bar{H}. \tag{B.33}$$

The force on the flange (not including the flange width) is

$$P_{yp} = \sigma_{yp} b t_f, \tag{B.34}$$

where in this case $\sigma_{yp} = (\sigma_m + \sigma_b)/2$. The moment arm of this force is given by

$$\bar{Y}_{yp} = \frac{d}{2} - \frac{1}{3} \left(\frac{2\sigma_b + \sigma_m}{\sigma_b + \sigma_m} \right) t_f. \tag{B.35}$$

Hence, for case 2, the additional moment due to the contribution of the flanges is

$$M_i(\alpha)_{\text{flanges case 2}} = 2(P_{yp} \bar{Y}_{yp}), \tag{B.36}$$

so that the total internal moment for case 2 is

$$M_i(\alpha) = M_i(\alpha)_{\text{rectangular}} + M_i(\alpha)_{\text{flanges case 2}}. \tag{B.37}$$

Then, as was done previously for the rectangular cross-section, the external moment minus the internal moment is set equal to zero, that is

$$M_e(\alpha) - M_i(\alpha) = 0, \tag{B.38}$$

for $x > x_y$. If $\alpha c < (d/2 - t_f)$ then $M_i(\alpha)$ is as given in Eq. (B.37), otherwise Eq. (B.30) is to be used. Equation (B.38) is solved numerically for α. The solution is easily obtained using Mathcad or some other scientific calculation software. The solved for value of α is a function of P and x and is denoted here as $\alpha(x, P)$ similar to Eq. (B.10). Finally, similar to Eqs. (B.11), (B.12) and (B.13) expressions for maximum strain, maximum stress and cantilever tip displacement are written for the case of an I-beam cross-section. That is

$$\varepsilon_m(x, P) = \begin{cases} \dfrac{Pcx}{EI} & x \leq x_y \\ \dfrac{\varepsilon_y}{\alpha(x, P)} & x > x_y \end{cases}, \tag{B.39}$$

$$\sigma_m(x, P) = \begin{cases} \dfrac{Pxc}{I} & x \leq x_y \\ \sigma_y + \left(\dfrac{1 - \alpha(x, P)}{\alpha(x, P)}\right) \varepsilon_y \bar{H} & x > x_y \end{cases} \tag{B.40}$$

and

$$\delta(P) = \int_0^L x \frac{\varepsilon_m(x, P)}{c}\, dx. \tag{B.41}$$

To this last expression, for cantilever tip displacement, a shear term is added. This addition provides more accurate results particularly in the linear range. The added shear term is

$$\delta(P)_{\text{shear}} = \frac{PL}{Gt_w d}. \tag{B.42}$$

Appendix C

Derivation of Analytical Solution for Large Displacement Cantilever

C.1 Introduction

An analytical solution for the large displacements of an elastic cantilever, with end load, is derived below. The solution closely follows that given by Khosravi et al. [48]. This solution is based on the use of a linear elastic material and is consistent with Euler-Bernoulli beam theory. Axial deformations are neglected in the solution by Khosravi, however, it is a simple matter to include axial deformations as indicated in the section after the main derivation. Shear deformations can be added similarly.

C.2 Derivation

Consider the arrangement shown in Figure C.1. A cantilever is shown in its deflected configuration with a point load, P, acting upwards at the right end. The origin of the coordinate system is chosen to coincide with the deflected end of the cantilever with positive

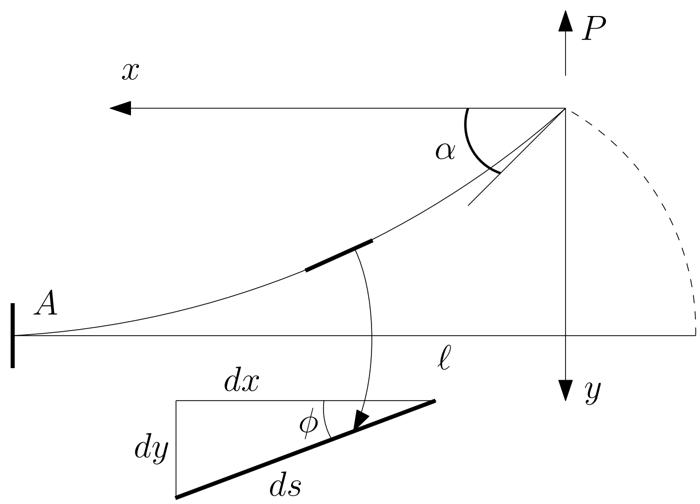


Figure C.1: Differential element of length ds.

x axis to the left and positive y axis downwards as shown. The angle α is measured from the positive x axis to the tangent line drawn through the end of the elastic curve. The cantilever is fixed at support A. The cantilever has length ℓ as shown. The cantilever material has modulus of elasticity E and the cantilever has moment of inertia I.

Consider now the differential element ds taken from the elastic curve. This differential element is at an angle of ϕ from the the positive x axis as shown. Using the standard moment curvature relations

$$EI\frac{d\phi}{ds} = M = -Px. \tag{C.1}$$

Differentiating both sides with respect to s

$$EI\frac{d^2\phi}{ds^2} = -P\frac{dx}{ds} = -P\cos\phi \tag{C.2}$$

$$\frac{d^2\phi}{ds^2} = -k^2\cos\phi \tag{C.3}$$

where $k^2 = P/EI$. Next integrate both sides of Eq. (C.3) with respect to ϕ to get

$$\int \frac{d^2\phi}{ds^2}\, d\phi = -k^2 \int \cos\phi \, d\phi. \tag{C.4}$$

By the chain rule modify the first integral so that

$$\int \frac{d^2\phi}{ds^2}\frac{d\phi}{ds}\, ds = -k^2 \int \cos\phi \, d\phi. \tag{C.5}$$

Next recognize that Eq. (C.5) is equivalent to

$$\frac{1}{2}\int \frac{d^2\phi}{ds^2}\frac{d\phi}{ds} + \frac{d^2\phi}{ds^2}\frac{d\phi}{ds}\, ds = -k^2 \int \cos\phi \, d\phi, \tag{C.6}$$

or,

$$\frac{1}{2}\int \frac{d}{ds}\left(\frac{d\phi}{ds}\right)^2 ds = -k^2 \int \cos\phi \, d\phi. \tag{C.7}$$

Upon carrying out the integration of Eq. (C.7)

$$\frac{1}{2}\left(\frac{d\phi}{ds}\right)^2 = -k^2 \sin\phi + c. \tag{C.8}$$

The boundary conditions at $x = 0$ are $d\phi/ds = 0$ (i.e. $M = 0$) and $\phi = \alpha$, which implies from Eq. (C.8) that

$$c = k^2 \sin\alpha. \tag{C.9}$$

This last result substituted into Eq. (C.8) yields

$$\left(\frac{d\phi}{ds}\right)^2 = 2k^2(\sin\alpha - \sin\phi). \tag{C.10}$$

From Eq. (C.1) it follows that $d\phi/ds < 0$, consequently solving Eq. (C.10) for ds yields

$$ds = \frac{-d\phi}{k\sqrt{2}\sqrt{\sin\alpha - \sin\phi}}. \tag{C.11}$$

Making use of the expression for ds it follows that

$$\ell = \int_{\alpha}^{0} ds = \int_{0}^{\alpha} -ds = \frac{1}{k\sqrt{2}} \int_{0}^{\alpha} \frac{d\phi}{\sqrt{\sin\alpha - \sin\phi}}, \tag{C.12}$$

$$X_A = \int_{\alpha}^{0} dx = \int_{\alpha}^{0} \cos\phi \; ds = \int_{0}^{\alpha} \cos\phi \; (-ds) = \frac{1}{k\sqrt{2}} \int_{0}^{\alpha} \frac{\cos\phi \; d\phi}{\sqrt{\sin\alpha - \sin\phi}}, \tag{C.13}$$

$$Y_A = \int_{\alpha}^{0} dy = \int_{\alpha}^{0} \sin\phi \; ds = \int_{0}^{\alpha} \sin\phi \; (-ds) = \frac{1}{k\sqrt{2}} \int_{0}^{\alpha} \frac{\sin\phi \; d\phi}{\sqrt{\sin\alpha - \sin\phi}}, \tag{C.14}$$

where ℓ is the length of the cantilever and (X_A, Y_A) are the coordinates of the fixed end from the origin located at the deflected end of the cantilever.

Equations (C.12), (C.13) and (C.14) are solved in the following steps.

1. For a given load P, properties E, I and length ℓ, equation Eq. (C.12) is used to numerically solve for α.

2. Having α from step 1 the values for X_A and Y_A are numerically solved for given P,E and I.

3. Finally, the deflected values of the tip are expressed as

$$\delta_x = \ell - X_A, \tag{C.15}$$

$$\delta_y = Y_Z. \tag{C.16}$$

4. The deflected values for any load P are obtained by repeating steps 1 to 3 as necessary.

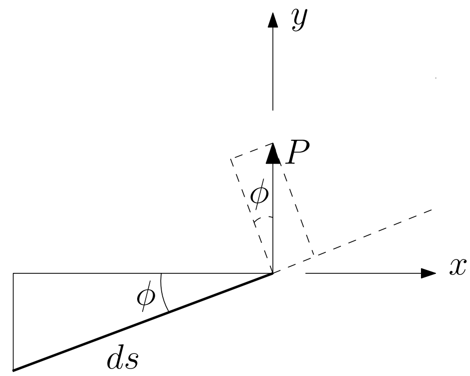


Figure C.2: Elastic curve segment ds.

C.3 Adding the effect of axial deformations

In this section the additional x and y displacements at the tip of the cantilever, due to axial deformations, are derived. Use is made of several of the preceding equations. Consider the differential line segment ds shown in Figure C.2. For any segment ds the force acting along the segment is $P \sin\phi$. The axial deformation for the segment is

$$\delta_s = \frac{P \sin\phi \; ds}{AE}, \tag{C.17}$$

where A is the cross-sectional area of the beam. The components of deflection in the x and y direction for the differential segment ds are

$$\delta_{sx} = \delta_s \cos\phi = \frac{P \cos\phi \sin\phi \; ds}{AE} \tag{C.18}$$

$$\delta_{sy} = \delta_s \sin\phi = \frac{P \sin^2\phi \; ds}{AE}. \tag{C.19}$$

The expressions for $\cos\phi \; ds$ and $\sin\phi \; ds$ are found in Eqs. (C.13) and (C.14) respectively. Substituting these results in Eqs. (C.18) and (C.19) and integrating over the range of angle

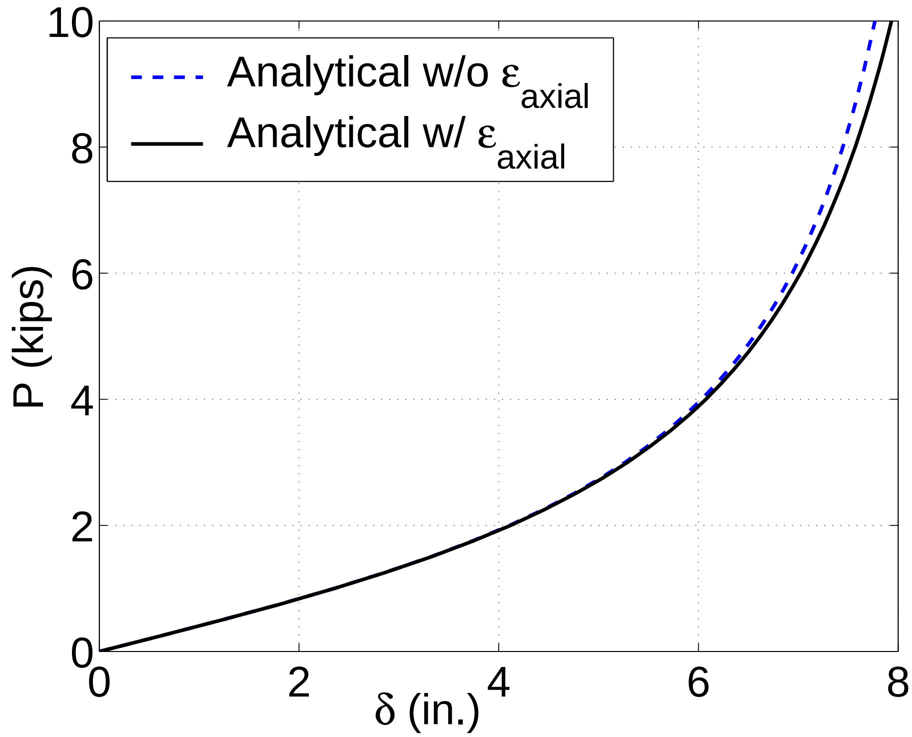


Figure C.3: Cantilever tip deflection δ_y.

α the final results for the axial deformations are

$$\delta_{x(\text{axial})} = \frac{1}{k\sqrt{2}} \int_0^{\alpha} \frac{P \sin\phi}{AE} \frac{\cos\phi \; d\phi}{\sqrt{\sin\alpha - \sin\phi}} \tag{C.20}$$

$$\delta_{y(\text{axial})} = \frac{1}{k\sqrt{2}} \int_0^{\alpha} \frac{P \sin\phi}{AE} \frac{\sin\phi \; d\phi}{\sqrt{\sin\alpha - \sin\phi}}. \tag{C.21}$$

C.4 Numerical example

Using Mathcad a load deflection plot is obtained for a linear elastic cantilever beam loaded at its free end. For this example $E = 100.0$ ksi, $I = 1.333$ in^4 and $\ell = 10.0$ in. Results are given in Figure C.3 for cases with and without including axial strain.

Appendix D

Meshfree Modeling of Wide Flange Beam Cross-Sections

D.1 Introduction

In the present work, beams are modeled as a 2D continuum of non-constant thickness in the direction perpendicular to the 2D plane of the domain. For example, in Figure D.1, the Voronoi diagram of a nodal set used to discretize a plane stress cantilever beam domain is shown. In the formulation a thickness, b_f, is specified for the flanges (top and bottom nodes) and a thickness, t_w, for the intermediate nodes of the web (see also Figure 4.4c). The specified thickness covers the Voronoi cell area corresponding to each node. As a result the grid spacing defines a grid specific t_f value and one can adjust b_f so the required I_{xx} results. The procedure to accomplish this for an I-section follows.

Given an I-beam as shown in Figure 4.4c, the moment of inertia is written as the sum

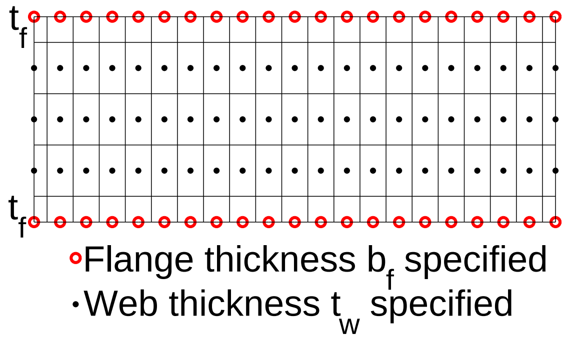


Figure D.1: Voronoi diagram for cantilever I-beam profile.

of contributions from the web and the flanges. This gives

$$I_w = \frac{t_w(d - 2t_f)^3}{12}, \tag{D.1}$$

$$I_f = \left[2t_f \left(\frac{d - t_f}{2}\right)^2 + \frac{t_f^3}{6}\right] b_f = B_r b_f, \tag{D.2}$$

where B_r is the coefficient that multiplies b_f in Eq. (D.2), so that

$$I_{xx} = I_w + I_f. \tag{D.3}$$

Now, if the grid spacing is arbitrary then t_f is set by the chosen grid. Hence, given I_{xx}, and specified values of t_w and d, the required thickness b_f is obtained as

$$b_f = \frac{I_{xx} - I_w}{B_r}. \tag{D.4}$$

Equation (D.4) gives the necessary domain thickness perpendicular to the 2D domain at the flanges. The specific procedure described above can be applied to any cross-section composed of rectangular sections, and, the concept can be extended to fairly arbitrary cross-sections.

D.2 Analytical solution for comparison to numerical results for elastic *I*-beam

Using the above procedure for a cantilever I-beam the results of Table 4.1 are obtained. The numerical analysis and comparison to analytical results proceeds as follows. First, consider the exact analytical elasticity solution for a cantilever beam (see Figure 4.4), with transverse shear load at its free end, as discussed in Timoshenko and Goodier [87], Belytschko *et al.* [16]

$$u_x = \frac{-Py}{6EI}\left[(6L-3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right] \tag{D.5a}$$

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

$$\sigma_{xx} = \frac{-P(L-x)y}{I} \tag{D.5c}$$

$$\sigma_{yy} = 0 \tag{D.5d}$$

$$\sigma_{xy} = \frac{P}{2I}\left(\frac{D^2}{4} - y^2\right) \tag{D.5e}$$

where I, which is the moment of inertia for a rectangular cross-section of unit thickness, is given by

$$I = \frac{D^3}{12}. \tag{D.5f}$$

Even though the solution above is for a *rectangular* beam, it can be applied to I-beams by simply replacing the moment of inertia term, I, in Eqs. (D.5a-c) by I_{xx}, the moment of inertia for the I-section. Similarly, the moment of inertia of the I-beam web, I_w is used in place of I in Eq. (D.5e) to determine the shear traction at the right end of the cantilever. The tip displacement solution of the I-beam cantilever is computed from the formula provided by Euler-Bernoulli beam theory with a shear term added, i.e.,

$$\delta_{\text{theoretical}} = \frac{PL^3}{3EI_{xx}} + K\frac{PL}{GA_w}, \tag{D.6}$$

where $K \approx 1.0$ is typical for I-beams, A_w is the I-beam web area and G is the shear modulus.

Appendix E

Algorithm for Plane Stress $J2$ Plasticity with Radial Return

The inelastic material model implemented in this dissertation is for the case of $J2$ plasticity under the conditions of plane stress. Several different hardening models are possible in the material formulation. For example, the implemented model includes the cases of linear and/or exponential isotropic hardening [93], as well as linear kinematic hardening. An implicit algorithm with radial return is employed in the computer implementation. The material model is summarized below and is based on the works by Simo and Taylor [80] and Simo and Hughes [78].

E.1 Preliminary definitions

In plane stress $J2$ flow plasticity the yield function is expressed in a convenient form by use of a $\mathbf{P}$ matrix defined as follows

$$\mathbf{P} = \frac{1}{3}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix} \tag{E.1}$$

To map the trial stress back to the yield surface use is made of a transformation matrix, $\mathbf{\Gamma}$, which is a function of the consistency parameter, $\Delta\gamma$, as follows:

$$\mathbf{\Gamma} = \begin{bmatrix} \alpha_1 & \alpha_2 & 0 \\ \alpha_2 & \alpha_1 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix} \tag{E.2}$$

where,

$\alpha_1 = \frac{1}{2+2a\Delta\gamma} + \frac{1}{2+2b\Delta\gamma}$

$\alpha_2 = \frac{1}{2+2a\Delta\gamma} - \frac{1}{2+2b\Delta\gamma}$

$\alpha_3 = \frac{1}{1+b\Delta\gamma}$

$a = \frac{E}{3(1-\nu)} + \frac{2}{3}H'$

$b = 2\mu + \frac{2}{3}H'$.

The squared form of the yield function is used in the return mapping algorithm to solve for the consistency parameter by Newton-Raphson iteration. The squared form of the yield function is

$$f^2(\Delta\gamma) = \frac{1}{2}\bar{f}^2(\Delta\gamma) - R^2(\Delta\gamma). \tag{E.3}$$

Functions $\bar{f}^2(\Delta\gamma)$ and $R^2(\Delta\gamma)$ are given as follows

$$\bar{f}^2(\Delta\gamma) = [\boldsymbol{\Gamma\xi}_{n+1}^{trial}]^T \mathbf{P}[\boldsymbol{\Gamma\xi}_{n+1}^{trial}] \tag{E.4}$$

$$R^2(\Delta\gamma) = \frac{1}{3}K^2(\alpha) \tag{E.5}$$

with K^2 evaluated at $\alpha = \alpha_n + \sqrt{\frac{2}{3}}\Delta\gamma\bar{f}(\Delta\gamma)$. Hence $f^2(\Delta\gamma)$ is evaluated appropriately as part of an implicit backward Euler integration scheme.

As part of the Newton-Raphson iterations the derivative of $f^2(\Delta\gamma)$ with respect to $\Delta\gamma$ is required. Denoting the derivative of $\boldsymbol{\Gamma}$ with respect to $\Delta\gamma$ as $d\boldsymbol{\Gamma}$, by the product rule

$$\frac{d(\bar{f}^2)}{d\Delta\gamma} = [d\boldsymbol{\Gamma\xi}_{n+1}^{trial}]^T \mathbf{P}[\boldsymbol{\Gamma\xi}_{n+1}^{trial}] + [\boldsymbol{\Gamma\xi}_{n+1}^{trial}]^T \mathbf{P}[d\boldsymbol{\Gamma\xi}_{n+1}^{trial}]. \tag{E.6}$$

The derivative of R^2 with respect to $\Delta\gamma$ is also easily found so that the derivative of the squared yield function is

$$\frac{d(f^2)}{d\Delta\gamma} = \frac{1}{2}\frac{d(\bar{f}^2)}{d\Delta\gamma} - \frac{d(R^2)}{d\Delta\gamma}. \tag{E.7}$$

The following additional definitions are necessary to understand the expressions given above and the algorithm of Section E.2.

$\boldsymbol{\xi}_{n+1}^{trial}$ = the trial relative stress = $\boldsymbol{\sigma}_{n+1}^{trial} - \tilde{\boldsymbol{\beta}}_n$.

$\tilde{\boldsymbol{\beta}}_n$ = a variable in plane stress subspace analogous to deviatoric back stress, this variable becomes nonzero in the presence of kinematic hardening (see the algorithm in Section E.2 for further details).

$K(\alpha) = \sigma_y + \theta\bar{H}\alpha + (\bar{K}_\infty - \bar{K}_o)[1 - e^{-\delta\alpha}]$ = the linear and exponential isotropic hardening laws combined.

α = the internal variable often called the equivalent plastic strain or plastic flow parameter.

$\bar{H}$ = the hardening modulus.

$\bar{K}_\infty$ = the saturation stress or ultimate stress σ_u.

$\bar{K}_o$ = the initial or yield stress σ_y.

δ = a material constant which dictates how much strain hardening is required to reach the ultimate stress after first yield.

K' = the derivative of K, the isotropic hardening law, with respect to α.

$H(\alpha) = (1-\theta)\bar{H}\alpha$ = the kinematic hardening law.

H' = the derivative of $H(\alpha)$, the kinematic hardening law, with respect to α.

$\mathbf{C} = \mathbf{C}_{elast}$ = the elastic modulus matrix.

Finally, θ is selected so that $\theta \in [0,1]$. By adjusting θ and the variables $\bar{H}, \bar{K}_\infty$, $\bar{K}_o$ and δ it is possible to obtain various types of hardening behavior. For example, using $\theta = 1$ and setting $\delta = 0$, one obtains linear isotropic hardening.

E.2 Algorithm for plane stress *J*2 flow plasticity with return mapping at the constitutive level

1. Update total strain and calculate trial elastic stresses.

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \Delta\boldsymbol{\varepsilon}_{n+1}$$

$$\boldsymbol{\sigma}^{trial} = \mathbf{C}[\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^p]$$

$$\boldsymbol{\xi}^{trial} = \boldsymbol{\sigma}^{trial} - \tilde{\boldsymbol{\beta}}_n$$

2. **IF** $f_{n+1}^{trial} \leq 0$ **THEN**

$$\boldsymbol{\varepsilon}_{n+1}^e = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^p$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}^{trial}$$

ELSE solve $f^2(\Delta\gamma) = 0$ for $\Delta\gamma$ by Newton-Raphson iteration.

3. Using $\Delta\gamma$ compute the modified (*algorithmic*) elastic tangent moduli

$$\boldsymbol{\Xi} = [\mathbf{C}^{-1} + \frac{\Delta\gamma}{1+\frac{2}{3}\Delta\gamma H'}\mathbf{P}]^{-1}$$

4. Using the solved for value of $\Delta\gamma$, update relative stress, backstress, stress, plastic strain, and plastic flow parameter.

$$\boldsymbol{\xi}_{n+1} = \frac{1}{1+\frac{2}{3}\Delta\gamma H'}\boldsymbol{\Xi}\mathbf{C}^{-1}\boldsymbol{\xi}^{trial} = \boldsymbol{\Gamma}\boldsymbol{\xi}^{trial}$$

$$\tilde{\boldsymbol{\beta}}_{n+1} = \tilde{\boldsymbol{\beta}}_n + \Delta\gamma\frac{2}{3}H'\boldsymbol{\xi}_{n+1}$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\xi}_{n+1} + \tilde{\boldsymbol{\beta}}_{n+1}$$

$$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n + \Delta\gamma\mathbf{P}\boldsymbol{\xi}_{n+1}$$

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + \Delta\gamma\sqrt{\frac{2}{3}}\bar{f}_{n+1}$$

5. Compute the *consistent* elasto-plastic tangent moduli

$$\mathbf{C}^{ep}_{consistent} = \boldsymbol{\Xi} - \frac{[\boldsymbol{\Xi}\mathbf{P}\boldsymbol{\xi}_{n+1}][\boldsymbol{\Xi}\mathbf{P}\boldsymbol{\xi}_{n+1}]^T}{\boldsymbol{\xi}^T_{n+1}\mathbf{P}\boldsymbol{\Xi}\mathbf{P}\boldsymbol{\xi}_{n+1} + \bar{\beta}_{n+1}}$$

$$\theta_1 = 1 + \frac{2}{3}H'\Delta\gamma, \quad \theta_2 = 1 - \frac{2}{3}K'_{n+1}\Delta\gamma, \quad \bar{\beta}_{n+1} = \frac{2\theta_1}{3\theta_2}(K'_{n+1}\theta_1 + H'\theta_2)\boldsymbol{\xi}^T_{n+1}\mathbf{P}\boldsymbol{\xi}_{n+1}$$

6. Update ε_{33}

$$\varepsilon_{33_{n+1}} = \frac{-\nu}{E}(\sigma_{11_{n+1}} + \sigma_{22_{n+1}}) - (\varepsilon^p_{11_{n+1}} + \varepsilon^p_{22_{n+1}})$$

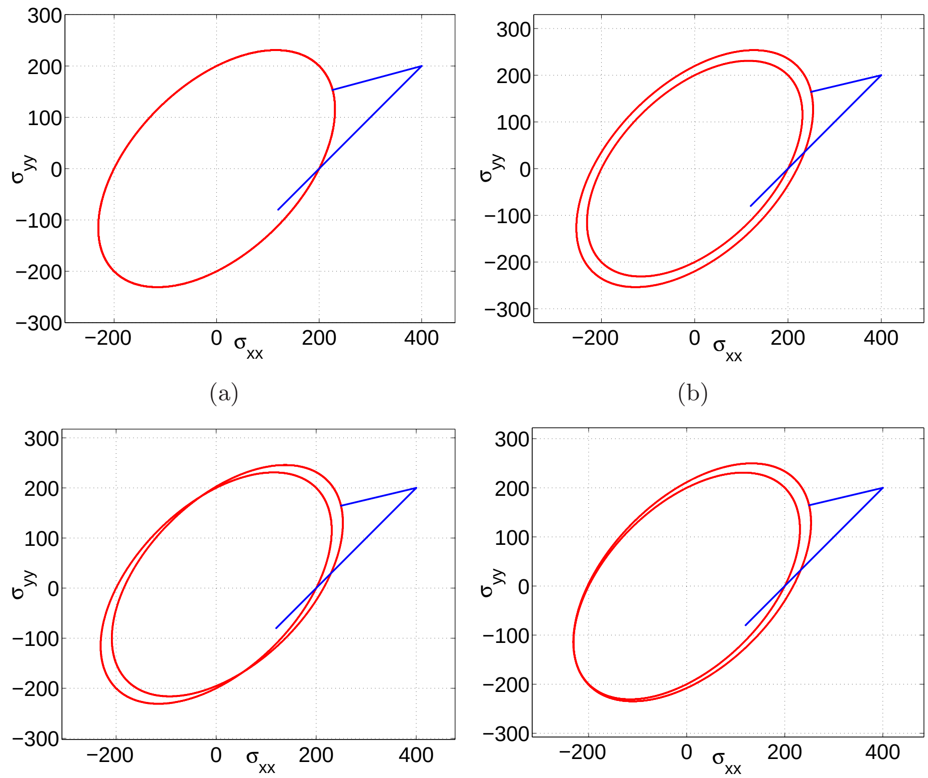
ENDIF

E.3 Examples

Several examples are presented to demonstrate the evolution of the yield surface according to $J2$ plasticity with radial return. To facilitate visualization of the various cases the examples are given in principal stress space. For all cases the initial stress state starts inside the yield surface at ($\sigma_{xx} = 120$ N/mm^2, $\sigma_{yy} = -80$ N/mm^2). A single strain increment ($\varepsilon_{xx} = 0.0014$, $\varepsilon_{yy} = 0.0014$) is given such that a trial stress state of ($\sigma_{xx} = 400$ N/mm^2, $\sigma_{yy} = 200$ N/mm^2) occurs outside the initial yield surface. Finally, the radial return mapping takes the stresses back in a direction normal to the final yield surface. The material properties used are yield stress $\sigma_y = 200$ N/mm^2, modulus of elasticity $E = 200000$ N/mm^2, Poisson's ratio $\nu = 0.0$ and hardening modulus $\bar{H} = 20000$ N/mm^2. Examples are given in Figures E.1a–d for the cases of no hardening (perfect plasticity), linear isotropic hardening, linear kinematic hardening and a case of 50 percent isotropic hardening combined with 50 percent kinematic hardening. The final stress state for each of these cases is given in Table E.1 (the results match those given by Crisfield [27, pages 181,182]). Since the hardening for isotropic, kinematic and combined are all in the same direction from the initial stress state, the final stress states, after the radial return, are identical.

Hardening Type	σ_{xx} (N/mm^2)	σ_{yy} (N/mm^2)
None	226.229	153.306
Linear Isotropic	249.585	164.404
Linear Kinematic	249.585	164.404
1/2 Isotropic + 1/2 Kinematic	249.585	164.404

Table E.1: Final stresses on yield surface.



(a) (b) (c) (d)

Figure E.1: Principal stress space (N/mm^2) yield surface evolution with radial return: (a) no hardening; (b) linear isotropic hardening; (c) linear kinematic hardening; and (d) 50 percent isotropic hardening combined with 50 percent kinematic hardening.

Increments	σ_{xx} (N/mm^2)	σ_{yy} (N/mm^2)
1	226.229	153.306
10	218.0830	174.8430

Table E.2: Final stresses on yield surface for different number of strain increments.

E.4 Increment size and sub-incrementation

It is worth noting that the plasticity algorithm given above with radial return is not error free. Even though the consistency parameter, necessary to achieve a yield function equal to zero, is solved for within tolerance by Newton-Raphson iterations the algorithm is not exact if the strain increments are too large. To demonstrate this the example of Figure E.1a is repeated. However, this time the one increment of strain is broken up into 10 equal increments. This is usually termed sub-incrementation. As seen in Figure E.2 the final stress state using sub-incrementation is not equivalent to the stress state obtained by a single strain increment (see Table E.2). To overcome this error it is necessary to use sufficiently small load or displacement steps at the global level of the nonlinear analysis to ensure small strain increments at the constitutive level. Alternatively, some form of sub-incrementation is necessary at the constitutive level in the plasticity algorithm. In this dissertation, sub-incrementation is not used and care is taken to ensure enough increments are used at the global level to ensure a converged solution.

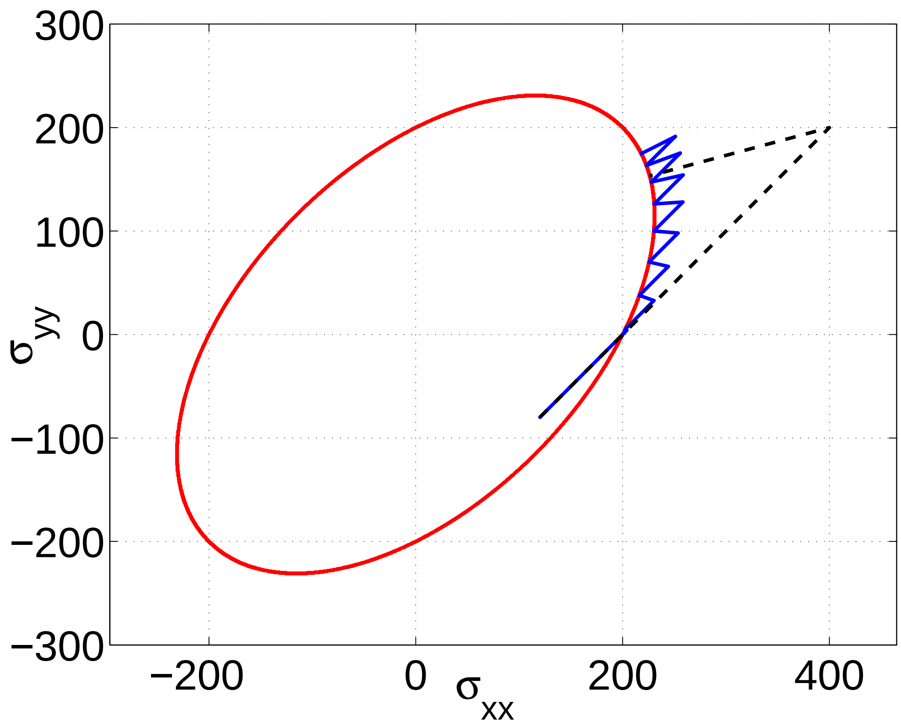


Figure E.2: Final stress state (N/mm^2) using single strain increment versus 10 sub-increments.

Appendix F

OpenSees Scripts

Several problems presented in this dissertation are analyzed using OpenSees [65] for comparison with the meshfree solutions. The scripts for the OpenSees analyses are included in this appendix.

F.1 Elasto-Plastic *I*-beam cantilever 1D fiber-section beam elements

```
# OpenSees Elasto-Plastic-Cantilever
# Using 1D Nonlinear Beam-Column elements and fiber-sections.
# by Louie L. Yaw - 3-21-08
#
# Units: kips, in, sec
# -------------------------------
# Start of model generation
# -------------------------------
wipe
# Create ModelBuilder (with two-dimensions and 3 DOF /node for
# beam column elements)
model BasicBuilder -ndm 2 -ndf 3

# Create nodes & add to Domain - command: node nodeId xCrd yCrd
set endnode 2
node  1  0.0 0.0
```

```
node  3  1.0 0.0
node  4  2.0 0.0
node  5  3.0 0.0
node  6  4.0 0.0
node  7  5.0 0.0
node  8  6.0 0.0
node  9  7.0 0.0
node 10  8.0 0.0
node 11  9.0 0.0
node  $endnode 10.0 0.0

# Set the boundary conditions - command: fix nodeID xResrnt?
# yRestrnt? zRestrnt
fix 1 1 1 1
puts "Nodes and Supports Defined"

# Define materials for beam/column elements
# -----------------------------------
uniaxialMaterial Hardening 1 29000.0 36.0 500.0 0.0

#Transformation tag definition
geomTransf Linear 1

#Define the wide flange section
set bf 3.0
set tw 1.0
set tf 0.2
set d 2.0

# START FIBER DEFINITION
set nfdw 16; # number of fibers along web depth
set nftw 1;  #number of fibers along web thickness, more if
             #bidirectional response wanted
set nfbf 1;  # use 16 when bidirectional response being modeled
 #number of fibers along flange width (you want this
             #many in a bi-directional loading)
set nftf 4;  # number of fibers along flange thickness

  set dw [expr $d - 2 * $tf]
  set y1 [expr -$d/2]
  set y2 [expr -$dw/2]
  set y3 [expr  $dw/2]
  set y4 [expr  $d/2]
```

```
  set z1 [expr -$bf/2]
  set z2 [expr -$tw/2]
  set z3 [expr  $tw/2]
  set z4 [expr  $bf/2]

  #
  section fiberSec  1 {
     #              nfIJ  nfJK    yI  zI    yJ  zJ    yK  zK    yL  zL
     patch quadr  1 $nfbf $nftf   $y1 $z4   $y1 $z1   $y2 $z1   $y2 $z4
     patch quadr  1 $nftw $nfdw   $y2 $z3   $y2 $z2   $y3 $z2   $y3 $z3
     patch quadr  1 $nfbf $nftf   $y3 $z4   $y3 $z1   $y4 $z1   $y4 $z4
  }

# Define elements
# ---------------
# Create nonlinear beamcolumn-command: element $type $elemtag
# $node1 $node2 $#intgrpts $sectag $transftag
element nonlinearBeamColumn 1 1 3 5 1 1
element nonlinearBeamColumn 2 3 4 5 1 1
element nonlinearBeamColumn 3 4 5 5 1 1
element nonlinearBeamColumn 4 5 6 5 1 1
element nonlinearBeamColumn 5 6 7 5 1 1
element nonlinearBeamColumn 6 7 8 5 1 1
element nonlinearBeamColumn 7 8 9 5 1 1
element nonlinearBeamColumn 8 9 10 5 1 1
element nonlinearBeamColumn 9 10 11 5 1 1
element nonlinearBeamColumn 10 11 $endnode 5 1 1

puts "Elements Defined"

# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 60
set Pmax -8.0
set uniload 1.0
set loadinc [expr $Pmax/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce zMoment
load $endnode 0.0 [expr $uniload] 0.0
}
puts "Loads Defined"
```

```
# ------------------------------
# End of model generation
# ------------------------------

# ------------------------------
# Start of analysis generation
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is used as
# homogeneous constraints
constraints Plain
# Create the integration scheme, the LoadControl scheme using
# steps of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-1 30
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------

# ------------------------------
# Start of recorder generation
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file plasticbeamcol.out -load -node $endnode -dof 2 disp
#recorder Element -file examplef.out -ele 1 force
# --------------------------------
# End of recorder generation
# ---------------------------------

# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
```

```
print node $endnode
print ele
```

F.2 Elasto-Plastic *I*-beam cantilever 2D FE enhanced-strain quadrilateral elements

```
# OpenSees Plastic-Cantilever
# Using 2D Quad 4 continuum plane stress elements.
# by Louie L. Yaw - 12-3-07
#
# Units: kips, in, sec
# ------------------------------
# Start of model generation
# ------------------------------
wipe
# Create ModelBuilder (with two-dimensions and 2 DOF /node)
model BasicBuilder -ndm 2 -ndf 2

# Define materials for truss elements
# -----------------------------------
# Create Elastoplast mat. prototype-command:
#nDMaterial J2Plasticity matID k G fy fu delta H eta
#nDMaterial ElasticIsotropic 2 29000 0.3
nDMaterial J2Plasticity 2 24166.7 11153.8 36.0 55.0 0.0 500.0 0.0
nDMaterial PlaneStress 1 2
#This next one is for use only with bbarquad making it
#do plane stress
nDMaterial J2Plasticity 3 16996.3 11153.8 36.0 55.0 0.0 500.0 0.0

#Define cantilever with a single 2D block command
set nx 50; #number of elements in x direction
set ny 10; #number of elements in y direction
set e1 1; #starting element number for generation
set n1 1; #starting node number for generation

#For regular quadrilateral element
#set eleArgs "1.0 PlaneStress 1"
#set Quad "quad"

#For Enhanced strain quadrilateral element
set eleArgs "PlaneStress 1"
set Quad "enhancedQuad"
```

```
#For bbar quadrilateral element
#set eleArgs "3"
#set Quad "bbarQuad"

block2D $nx $ny $n1 $e1 $Quad $eleArgs {
1  0.0 0.0
2 10.0 0.0
3 10.0 2.0
4  0.0 2.0
}
#the bottom node of a row is (row-1)*(nx+1)+nx+1

#Add another layer of same elements over the top, stiffness
#should double
# i variable is for rows
# j variable is for cols
# enum keeps track of the next element number if more
# would be added
set startrow 1
set endrow $ny
set enum [expr $nx*$ny+1]
for {set i $startrow} {$i <= $endrow} {incr i} {
for {set j 1} {$j <= $nx} {incr j} {
set nodei [expr ($i-1)*($nx+1)+$j]
set nodej [expr ($i-1)*($nx+1)+$j+1]
set nodek [expr $i*($nx+1)+$j+1]
set nodel [expr $i*($nx+1)+$j]
#Thicken the model at flanges
if {$i == $startrow || $i == $endrow} {
for {set k 1} {$k <= 2} {incr k} {
element $Quad $enum $nodei $nodej $nodek $nodel PlaneStress 1
incr enum
}
}
#Thicken the web at the supports
if {$i != $startrow && $i != $endrow && $j == 1} {
for {set k 1} {$k <= 2} {incr k} {
element $Quad $enum $nodei $nodej $nodek $nodel PlaneStress 1
incr enum
}
}
} ;#end loop over columns
} ;#end loop over rows
```

```
# Set the boundary conditions - all nodes at x=0.0
fixX 0.0 1 0
fix 256 1 1

puts "Nodes and Supports Defined"

puts "Elements Defined"
# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 100
set Pmax -8.0
set uniload 1.0
set loadinc [expr $Pmax/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce
load 51  0 [expr $uniload/11.0]
load 102 0 [expr $uniload/11.0]
load 153 0 [expr $uniload/11.0]
load 204 0 [expr $uniload/11.0]
load 255 0 [expr $uniload/11.0]
load 306 0 [expr $uniload/11.0]
load 357 0 [expr $uniload/11.0]
load 408 0 [expr $uniload/11.0]
load 459 0 [expr $uniload/11.0]
load 510 0 [expr $uniload/11.0]
load 561 0 [expr $uniload/11.0]

#load 8 0 [expr $uniload/4.0]
#load 12 0 [expr $uniload/4.0]
#load 16 0 [expr $uniload/4.0]
}
puts "Loads Defined"
# ------------------------------
# End of model generation
# ------------------------------
# ------------------------------
# Start of analysis generation
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
```

```
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is
# used as homogeneous constraints
constraints Transformation
# Create the integration scheme, the LoadControl scheme using steps
# of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-1 30 1
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------
# ------------------------------
# Start of recorder generation
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file plasticCantpvsu.out -load -node 306 -dof 2 disp
#recorder Element -file examplef.out -ele 1 force
# --------------------------------
# End of recorder generation
# ---------------------------------
# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
print node 306
#print ele
```

F.3 Frame corner connection with elasto-plastic panel zone

```
# OpenSees Elasto-Plastic-Beedle Frame Corner Connection -
# elasto-plastic panel zone
# Using 1D Nonlinear Beam-Column elements and sections.
# by Louie L. Yaw - 3-25-08
#
# Units: kips, in, sec
```

```
# ------------------------------
# Start of model generation
# ------------------------------
wipe
# Create ModelBuilder (with two-dimensions and 3 DOF /node for
# beam column elements)
model BasicBuilder -ndm 2 -ndf 3

# Create nodes & add to Domain - command: node nodeId xCrd yCrd
set endnode 23
node  1        0.0       0.0
node  2     8.5277    8.5277
node  3    17.0554   17.0554
node  4    25.5831   25.5831
node  5    34.1108   34.1108
node  6    42.6385   42.6385
node  7    51.1662   51.1662
node  8    59.6940   59.6940
node  9    68.2217   68.2217
node 10    76.7494   76.7494
node 11    85.2771   85.2771
node 12    95.8413   95.8413
node 13    85.2771  106.4054
node 14    76.7494  114.9331
node 15    68.2217  123.4608
node 16    59.6940  131.9886
node 17    51.1662  140.5163
node 18    42.6385  149.0440
node 19    34.1108  157.5717
node 20    25.5831  166.0994
node 21    17.0554  174.6271
node 22     8.5277  183.1548
node 23        0.0  191.6825

# Set the boundary conditions - command: fix nodeID xResrnt?
# yRestrnt? zRestrnt
fix 1 1 1 0
fix 23 1 0 0
puts "Nodes and Supports Defined"

# Define materials for beam/column elements
# -----------------------------------
#Steel Material
uniaxialMaterial Hardening 1 29000.0 36.0 150.0 0.0
```

```
#Transformation tag definition
geomTransf Linear 1

#Define the wide flange section
set bf 10.5
set tw 0.545
set tf 0.76
set d 29.875

# START FIBER DEFINITION
set nfdw 16; # number of fibers along web depth
set nftw 1;  #number of fibers along web thickness, more if
             # bidirectional response wanted
set nfbf 1;  # use 16 when bidirectional response being modeled
 #number of fibers along flange width (you want this many in
 # a bi-directional loading)
set nftf 4;  # number of fibers along flange thickness

  set dw [expr $d - 2 * $tf]
  set y1 [expr -$d/2]
  set y2 [expr -$dw/2]
  set y3 [expr  $dw/2]
  set y4 [expr  $d/2]

  set z1 [expr -$bf/2]
  set z2 [expr -$tw/2]
  set z3 [expr  $tw/2]
  set z4 [expr  $bf/2]

  #
  section fiberSec  1 {
     #              nfIJ  nfJK    yI  zI    yJ  zJ    yK  zK    yL  zL
     patch quadr  1 $nfbf $nftf   $y1 $z4   $y1 $z1   $y2 $z1   $y2 $z4
     patch quadr  1 $nftw $nfdw   $y2 $z3   $y2 $z2   $y3 $z2   $y3 $z3
     patch quadr  1 $nfbf $nftf   $y3 $z4   $y3 $z1   $y4 $z1   $y4 $z4
  }

# Define elements
# ---------------
# Create nonlinear beamcolumn-command: element $type $elemtag $node1 $node2
# $#intgrpts $sectag $transftag
element dispBeamColumn 1  1 2 5 1 1
element dispBeamColumn 2  2 3 5 1 1
```

```
element dispBeamColumn 3  3 4 5 1 1
element dispBeamColumn 4  4 5 5 1 1
element dispBeamColumn 5  5 6 5 1 1
element dispBeamColumn 6  6 7 5 1 1
element dispBeamColumn 7  7 8 5 1 1
element dispBeamColumn 8  8 9 5 1 1
element dispBeamColumn 9  9 10 5 1 1
element dispBeamColumn 10 10 11 5 1 1
element dispBeamColumn 11 11 12 5 1 1
element dispBeamColumn 12 12 13 5 1 1
element dispBeamColumn 13 13 14 5 1 1
element dispBeamColumn 14 14 15 5 1 1
element dispBeamColumn 15 15 16 5 1 1
element dispBeamColumn 16 16 17 5 1 1
element dispBeamColumn 17 17 18 5 1 1
element dispBeamColumn 18 18 19 5 1 1
element dispBeamColumn 19 19 20 5 1 1
element dispBeamColumn 20 20 21 5 1 1
element dispBeamColumn 21 21 22 5 1 1
element dispBeamColumn 22 22 23 5 1 1

puts "Elements Defined"

# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 100
set Pmax -150.0
set uniload 1.0
set loadinc [expr $Pmax/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce zMoment
load $endnode 0.0 [expr $uniload] 0.0
}
puts "Loads Defined"

# ------------------------------
# End of model generation
# ------------------------------

# ------------------------------
# Start of analysis generation
```

```
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is used as
# homogeneous constraints
constraints Plain
# Create the integration scheme, the LoadControl scheme using
# steps of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-1 30
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------

# ------------------------------
# Start of recorder generation
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file beedleeppanel.out -load -node $endnode -dof 2 disp
#recorder Element -file examplef.out -ele 1 force
# --------------------------------
# End of recorder generation
# ---------------------------------

# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
print node $endnode
print ele
```

F.4 Frame corner connection with elastic panel zone

```
# OpenSees Elasto-Plastic-Beedle Frame Corner Connection -
```

```
# elastic panel zone
# Using 1D Nonlinear Beam-Column elements and sections.
# by Louie L. Yaw - 3-25-08
#
# Units: kips, in, sec
# ------------------------------
# Start of model generation
# ------------------------------
wipe
# Create ModelBuilder (with two-dimensions and 3 DOF /node
# for beam column elements)
model BasicBuilder -ndm 2 -ndf 3

# Create nodes & add to Domain - command: node nodeId xCrd yCrd
set endnode 23
node  1         0.0       0.0
node  2      8.5277    8.5277
node  3     17.0554   17.0554
node  4     25.5831   25.5831
node  5     34.1108   34.1108
node  6     42.6385   42.6385
node  7     51.1662   51.1662
node  8     59.6940   59.6940
node  9     68.2217   68.2217
node 10     76.7494   76.7494
node 11     85.2771   85.2771
node 12     95.8413   95.8413
node 13     85.2771  106.4054
node 14     76.7494  114.9331
node 15     68.2217  123.4608
node 16     59.6940  131.9886
node 17     51.1662  140.5163
node 18     42.6385  149.0440
node 19     34.1108  157.5717
node 20     25.5831  166.0994
node 21     17.0554  174.6271
node 22      8.5277  183.1548
node 23         0.0  191.6825

# Set the boundary conditions - command: fix nodeID xResrnt?
# yRestrnt? zRestrnt
fix 1 1 1 0
fix 23 1 0 0
puts "Nodes and Supports Defined"
```

```
# Define materials for beam/column elements
# -----------------------------------
#Steel Material
uniaxialMaterial Hardening 1 29000.0 36.0 150.0 0.0

#Transformation tag definition
geomTransf Linear 1

#Define the wide flange section
set bf 10.5
set tw 0.545
set tf 0.76
set d 29.875

# START FIBER DEFINITION
set nfdw 16; # number of fibers along web depth
set nftw 1;  #number of fibers along web thickness, more if
             #bidirectional response wanted
set nfbf 1;  # use 16 when bidirectional response being modeled
 #number of fibers along flange width (you want this many
 #in a bi-directional loading)
set nftf 4;  # number of fibers along flange thickness

  set dw [expr $d - 2 * $tf]
  set y1 [expr -$d/2]
  set y2 [expr -$dw/2]
  set y3 [expr  $dw/2]
  set y4 [expr  $d/2]

  set z1 [expr -$bf/2]
  set z2 [expr -$tw/2]
  set z3 [expr  $tw/2]
  set z4 [expr  $bf/2]

  #
  section fiberSec  1 {
     #              nfIJ  nfJK    yI  zI    yJ  zJ    yK  zK    yL  zL
     patch quadr  1 $nfbf $nftf   $y1 $z4   $y1 $z1   $y2 $z1   $y2 $z4
     patch quadr  1 $nftw $nfdw   $y2 $z3   $y2 $z2   $y3 $z2   $y3 $z3
     patch quadr  1 $nfbf $nftf   $y3 $z4   $y3 $z1   $y4 $z1   $y4 $z4
  }

# Define elements
```

```
# ---------------
# Create nonlinear beamcolumn-command: element $type $elemtag $node1 $node2
# $#intgrpts $sectag $transftag
element dispBeamColumn 1  1 2 5 1 1
element dispBeamColumn 2  2 3 5 1 1
element dispBeamColumn 3  3 4 5 1 1
element dispBeamColumn 4  4 5 5 1 1
element dispBeamColumn 5  5 6 5 1 1
element dispBeamColumn 6  6 7 5 1 1
element dispBeamColumn 7  7 8 5 1 1
element dispBeamColumn 8  8 9 5 1 1
element dispBeamColumn 9  9 10 5 1 1
element dispBeamColumn 10 10 11 5 1 1
element elasticBeamColumn 11 11 12 31.7 29000 4470 1
element elasticBeamColumn 12 12 13 31.7 29000 4470 1
#rigidLink beam 11 12
#rigidLink beam 12 13
element dispBeamColumn 13 13 14 5 1 1
element dispBeamColumn 14 14 15 5 1 1
element dispBeamColumn 15 15 16 5 1 1
element dispBeamColumn 16 16 17 5 1 1
element dispBeamColumn 17 17 18 5 1 1
element dispBeamColumn 18 18 19 5 1 1
element dispBeamColumn 19 19 20 5 1 1
element dispBeamColumn 20 20 21 5 1 1
element dispBeamColumn 21 21 22 5 1 1
element dispBeamColumn 22 22 23 5 1 1

puts "Elements Defined"

# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 100
set Pmax -160.0
set uniload 1.0
set loadinc [expr $Pmax/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce zMoment
load $endnode 0.0 [expr $uniload] 0.0
}
puts "Loads Defined"
```

```
# ------------------------------
# End of model generation
# ------------------------------

# ------------------------------
# Start of analysis generation
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is used as
# homogeneous constraints
constraints Plain
# Create the integration scheme, the LoadControl scheme using
# steps of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-1 30
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------

# ------------------------------
# Start of recorder generation
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file beedleepanel.out -load -node $endnode -dof 2 disp
#recorder Element -file examplef.out -ele 1 force
# --------------------------------
# End of recorder generation
# ---------------------------------

# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
```

```
print node $endnode
print ele
```

F.5 Elasto-plastic portal frame with elastic panel zone (Baker and Roderick [9])

```
# OpenSees Elasto-Plastic-Baker Frame
# Using 1D Nonlinear Beam-Column elements and sections.
# by Louie L. Yaw - 4-3-08
#
# Units: kips, in, sec
# ------------------------------
# Start of model generation
# ------------------------------
wipe
# Create ModelBuilder (with two-dimensions and 3 DOF /node
# for beam column elements)
model BasicBuilder -ndm 2 -ndf 3

# Create nodes & add to Domain - command: node nodeId xCrd yCrd
set dispnode 16
set vertnode 11
node  1         0.0       0.0
node  2         0.0      23.0
node  3         0.0      46.0
node  4         0.0      69.0
node  5         0.0      92.0
node  6         0.0      96.0
node  7         4.0      96.0
node  8        27.0      96.0
node  9        50.0      96.0
node 10        73.0      96.0
node 11        96.0      96.0
node 12       119.0      96.0
node 13       142.0      96.0
node 14       165.0      96.0
node 15       188.0      96.0
node 16       192.0      96.0
node 17       192.0      92.0
node 18       192.0      69.0
node 19       192.0      46.0
node 20       192.0      23.0
```

```
node 21       192.0       0.0

# Set the boundary conditions - command: fix nodeID xResrnt?
# yRestrnt? zRestrnt
fix 1 1 1 0
fix 21 1 1 0
puts "Nodes and Supports Defined"

# Define materials for beam/column elements
# -------------------------------------
#Steel Material
uniaxialMaterial Hardening 1 29000.0 33.0 150.0 0.0

#Transformation tag definition
geomTransf Linear 1

#SECTION 1 DEFINED HERE
#Define the wide flange section
set bf 4.0
set tw 0.25
set tf 0.422
set d 8.0

# START FIBER DEFINITION
set nfdw 16; # number of fibers along web depth
set nftw 1;  #number of fibers along web thickness, more if
             #bidirectional response wanted
set nfbf 1;  # use 16 when bidirectional response being modeled
 #number of fibers along flange width (you want this many in
 #a bi-directional loading)
set nftf 4;  # number of fibers along flange thickness

  set dw [expr $d - 2 * $tf]
  set y1 [expr -$d/2]
  set y2 [expr -$dw/2]
  set y3 [expr  $dw/2]
  set y4 [expr  $d/2]

  set z1 [expr -$bf/2]
  set z2 [expr -$tw/2]
  set z3 [expr  $tw/2]
  set z4 [expr  $bf/2]

  #
```

```
  section fiberSec  1 {
     #              nfIJ  nfJK  yI  zI    yJ  zJ    yK  zK   yL  zL
     patch quadr  1 $nfbf $nftf $y1 $z4   $y1 $z1   $y2 $z1  $y2 $z4
     patch quadr  1 $nftw $nfdw $y2 $z3   $y2 $z2   $y3 $z2  $y3 $z3
     patch quadr  1 $nfbf $nftf $y3 $z4   $y3 $z1   $y4 $z1  $y4 $z4
  }
#END SECTION 1 DEFINITION

# Define elements
# ---------------
# Create nonlinear beamcolumn-command: element $type $elemtag $node1
# $node2 $#intgrpts $sectag $transftag
element dispBeamColumn 1  1 2 5 1 1
element dispBeamColumn 2  2 3 5 1 1
element dispBeamColumn 3  3 4 5 1 1
element dispBeamColumn 4  4 5 5 1 1
#element dispBeamColumn 5  5 6 5 1 1
#element dispBeamColumn 6  6 7 5 1 1
element elasticBeamColumn 5 5 6 5.165 29000 56.18 1
element elasticBeamColumn 6 6 7 5.165 29000 56.18 1
element dispBeamColumn 7  7 8 5 1 1
element dispBeamColumn 8  8 9 5 1 1
element dispBeamColumn 9  9 10 5 1 1
element dispBeamColumn 10 10 11 5 1 1
element dispBeamColumn 11 11 12 5 1 1
element dispBeamColumn 12 12 13 5 1 1
element dispBeamColumn 13 13 14 5 1 1
element dispBeamColumn 14 14 15 5 1 1
#element dispBeamColumn 15 15 16 5 1 1
#element dispBeamColumn 16 16 17 5 1 1
element elasticBeamColumn 15 15 16 5.165 29000 56.18 1
element elasticBeamColumn 16 16 17 5.165 29000 56.18 1
element dispBeamColumn 17 17 18 5 1 1
element dispBeamColumn 18 18 19 5 1 1
element dispBeamColumn 19 19 20 5 1 1
element dispBeamColumn 20 20 21 5 1 1

#element dispBeamColumn 11 11 12 5.165 29000 56.18 1
#element dispBeamColumn 12 12 13 5.165 29000 56.18 1
puts "Elements Defined"

# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 100
set Pmax 14.0
```

```
set uniload 1.0
set loadinc [expr $Pmax/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce zMoment
#the lateral point load
load $dispnode [expr $uniload] 0.0 0.0
#the vertical point load
load $vertnode 0.0 [expr -$uniload] 0.0
}
puts "Loads Defined"

# ------------------------------
# End of model generation
# ------------------------------

# ------------------------------
# Start of analysis generation
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is used as
# homogeneous constraints
constraints Plain
# Create the integration scheme, the LoadControl scheme using
# steps of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-1 30
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------

# ------------------------------
# Start of recorder generation
```

```
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file bakerpvsu1D.out -load -node $dispnode -dof 1 disp
#recorder Element -file examplef.out -ele 1 force
# --------------------------------
# End of recorder generation
# ---------------------------------

# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
print node $dispnode
print ele
```

F.6 Elastic cantilever beam using 1D co-rotational beam element

```
# OpenSees co-rotational beam element
# by Louie L. Yaw - 1-24-08
#
# Units: kips, in, sec
# ------------------------------
# Start of model generation
# ------------------------------
# Create ModelBuilder (with two-dimensions and 3 DOF /node)
model BasicBuilder -ndm 2 -ndf 3
# Create nodes & add to Domain - command: node nodeId xCrd yCrd
node 1 0.0 0.0
node 2 1.0 0.0
node 3 2.0 0.0
node 4 3.0 0.0
node 5 4.0 0.0
node 6 5.0 0.0
node 7 6.0 0.0
node 8 7.0 0.0
node 9 8.0 0.0
node 10 9.0 0.0
node 11 10.0 0.0
```

```
# Set the boundary conditions - command: fix nodeID
# xResrnt? yRestrnt?
fix 1 1 1 1
puts "Nodes and Supports Defined"
# Define materials for truss elements
# ----------------------------------
# Create Elastic material prototype - command:
# uniaxialMaterial Elastic matID E
uniaxialMaterial Elastic 1 1e2
# Create the type of geomtransformation - command:
# geomTransf transftype transtag
geomTransf Co-rotational 1
# Define elements
# ---------------
# Create beam elements - command: element elasticbeamcolumn
# eleID node1 node2 A E I transfTag
element elasticBeamColumn 1 1 2 4 100.0 1.33333 1
element elasticBeamColumn 2 2 3 4 100.0 1.33333 1
element elasticBeamColumn 3 3 4 4 100.0 1.33333 1
element elasticBeamColumn 4 4 5 4 100.0 1.33333 1
element elasticBeamColumn 5 5 6 4 100.0 1.33333 1
element elasticBeamColumn 6 6 7 4 100.0 1.33333 1
element elasticBeamColumn 7 7 8 4 100.0 1.33333 1
element elasticBeamColumn 8 8 9 4 100.0 1.33333 1
element elasticBeamColumn 9 9 10 4 100.0 1.33333 1
element elasticBeamColumn 10 10 11 4 100.0 1.33333 1

puts "Elements Defined"
# My Own Definitions for numLoad Steps, Pmax, loadinc per step
set nsteps 80
set Pload -10.0
set uniload 1.0
set loadinc [expr $Pload/$nsteps]

# Define loads
# ------------
# Create a Plain load pattern with a linear TimeSeries
pattern Plain 1 "Linear" {
# Create the nodal load -command: load nodeID xForce yForce
load 11 0 $uniload 0
}
puts "Loads Defined"
# ------------------------------
# End of model generation
```

```
# ------------------------------
# ------------------------------
# Start of analysis generation
# ------------------------------
# Create the system of equation, a SPD using a band storage scheme
system BandSPD
# Create the DOF numberer, the reverse Cuthill-McKee algorithm
numberer RCM
# Create the constraint handler, a Plain handler is used as
# homogeneous constraints
constraints Transformation
# Create the integration scheme, the LoadControl scheme using
# steps of loadpercent of total
integrator LoadControl $loadinc
# Specify Test For Convergence Criteria
test NormUnbalance 1.0e-4 10
# Create the solution algorithm, a Linear algorithm is created
algorithm Newton
# create the analysis object
analysis Static
# ------------------------------
# End of analysis generation
# ------------------------------
# ------------------------------
# Start of recorder generation
# ------------------------------
# create a Recorder object for the nodal displacements at node 4
recorder Node -file example.out -load -node 11 -dof 2 disp
recorder Element -file examplef.out -ele 2 force
# --------------------------------
# End of recorder generation
# ---------------------------------
# ------------------------------
# Finally perform the analysis
# ------------------------------
# Perform the analysis
analyze $nsteps
# Print the current state at node 2 and at all elements
print node
print ele
```

Appendix G

Analysis Program and User Manual

In this appendix, a summary of the meshfree research code used to generate the results in many of the examples of this dissertation is presented. It is the intent of the summary to describe the major aspects of the code that were implemented as part of this research. Also a user manual is given in the section on input.

G.1 Meshfree co-rotational analysis program implementation

The meshfree co-rotational analysis program is divided into the four basic parts of input, preprocessing, analysis, and post processing. Each of these are described along with relevant implementation details in the following subsections. The description provided below is based on a load control scheme, however, certain details required for displacement control are mentioned as well.

G.1.1 Input - user manual

Introduction

This section describes the input file required for execution of the Element Free Galerkin Analysis Program (EFGAP). EFGAP is a small strain plane stress J2 elasto-plasticity “meshfree” analysis program. It is meshfree in the sense that the interpolation functions are maximum-entropy basis functions and hence are not dependent on the initial mesh used to generate the layout of nodes. It is meshfree also in the sense that integration of the weak form of the elasto-statics problem is performed by using nodal integration over Voronoi diagram cell boundaries enclosing each node in the domain. EFGAP is also implemented using a co-rotational formulation so that even though strains are small, large displacements and rotations are allowed. EFGAP has many built in options that are specified by the user in the main input file. This main input file is explained first.

Main Input File

The executable file for EFGAP is run in a windows console. The user is immediately prompted for the input file. The input file can have a name of up to 80 characters in length. The input file itself starts with a descriptive title followed by a variety of sections each of which contain numeric data. Each of the sections must have a descriptive title of up to 90 characters in length. Descriptive titles are required in the input file to help locate numeric data and to make editing more manageable. On lines following the descriptive title numeric data must be provided with one or more spaces or a return between each numeric value. Numeric data must never be on a descriptive title line or it is assumed to be part of the descriptive title. A description of a typical input file is given below. Note that the data descriptions under each section are given in the required order that they must be provided.

Also, be sure to enter integers as integers (i.e., without decimals as shown in the examples below).

1. Model Title

   Title – A model title is required on the first line of the input file. A descriptive title of up to 90 characters in length is allowed. An example is shown in the following box.

   Example

   ```
   EFGAP input file - Model Cantilever Beam (January 1, 2007)
   ```

2. MLS Variables

   Title – A title for this section and all subsequent sections is required. Titles must be on one line and cannot exceed 90 characters in length.

   α – MLS variable α adjusts the support radius of the MLS shape functions. The value should be based on the quadrilateral in the mesh (if a mesh is used for laying out the grid of nodes) that gives the biggest value when the longest side length is divided by the shortest side length of the quadrilateral, this then defines the *worst* aspect ratio. If the aspect ratio increases the value of α will likely need to increase or the program may fail to execute. The user should always try to use the smallest value of α possible. Values as low as 0.9 have been successful for aspect ratios of 1.0. If a warning arises during execution of EFGAP, that the **A** matrix is singular, then it is likely that the value of α should be increased. It is recommended that the aspect ratio be kept close to 1.0 as much as possible for better accuracy and reduced analysis time. Values greater than 2.0 should be avoided. In the example shown below the value of α is 1.0.

   Example

```
MLS variables
1.0
```

3. Analysis variables

Title – The title must be on one line and cannot exceed 90 characters in length.

Number of load increments – the number of load increments in which the load is to be applied. The first load increment will be the user estimated yield load, P_y. After the first load increment is applied the remaining uniform load increments add up to $P_{max} - P_y$. This scheme is devised to help minimize the amount of time wasted calculating elastic load steps.

P_{max} – The maximum load to be applied, unless collapse occurs before this load is reached.

P_y - The estimated yield load. Actually this should be something less than the yield load. This helps minimize the analysis time wasted on the elastic loading phase. Although this may not be known at first one may estimate the value or run the analysis with a small number of load increments first and get an estimate of the yield load.

Collapse Residual – The residual at which collapse is considered to have occurred. If a residual during Newton-Raphson iterations exceeds this value the program writes to file the data from the last stable load increment and the program terminates. Basically this is a collapse criteria. A good value is 1000.0. If load steps are sufficiently small the residuals stay small and this criteria is not violated.

Maximum Number of Iterations – The maximum number of Newton-Raphson iterations that are to be used to try and achieve equilibrium. If equilibrium is not reached the program stops iterations and goes on to the next load increment. Such a case likely means that final results are questionable and probably are not valid. In such a case the program will report that the equilibrium tolerance has not been met within the maximum number of iterations allowed. A value of 200 works fine and is really much bigger than necessary. If many iterations are required a convergence problem is likely the result and step size needs to be reduced or something is probably incorrect in the input file.

Equilibrium Tolerance – The residual below which Newton-Raphson iterations are considered to have achieved equilibrium for the current load increment. This seems to give acceptable results with a value as large as 0.1. The smaller the residual of course the more iterations and hence time required. Values smaller than 0.1 do not appear to give significantly different results.

Displacement Node – One of the output files is a load displacement file. The user must provide the node number at which to track displacement so that the load displacement file is created.

Displacement Node DOF – the degree of freedom of displacement to track at the Displacement Node. A value of 0 is for the x degree of freedom and a value of 1 indicates the y degree of freedom.

Example

```
Analysis Variables
100 12.0 9.0 10.0 200 0.1 98 1
```

4. Material Properties

Title – The title must be on one line and cannot exceed 90 characters in length.

$\bar{H}$ – The linear hardening modulus for isotropic or kinematic hardening depending on the value of θ specified (see Appendix E).

$\bar{K}_\infty$ – The saturation stress (equivalent to ultimate stress σ_u) for the case of isotropic exponential hardening. In other words the maximum stress that can be achieved if *only* exponential hardening is used.

$\bar{K}_o$ – The yield stress (commonly denoted as σ_y).

θ – A parameter in the range $[0, 1]$ that allows specification of different types of hardening behavior (see Appendix E).

δ – A parameter that dictates how rapidly the saturation stress is reached after first yield. In other words, the bigger the value the smaller the strain required to reach the saturation stress after first yield.

ν – Possion's ratio.

E – Young's modulus of elasticity.

σ_y – The yield stress for the material model. This value and Ko should be identical in value.

α_s – The stabilization factor, a value of 1.0 is usually used.

μ_s – The μ value for the modulus matrix $\mathbf{C}_s$ used to provide stabilization for the nodally integrated stiffness matrices. Per the recommendations of Puso *et al.* [72] this value should be set to the value $\bar{H}/2$ in the case of linear hardening. In the case of exponential hardening this value should be set to $0.5\delta(K_\infty - Ko)$, which is one half the slope value of the exponential hardening curve at the yield strain, ε_y. Using a μ_s value less than zero sets the stabilization modulus matrix, $\mathbf{C}_s$, equal to the elastic modulus matrix, $\mathbf{C}_{elast}$.

Example

```
Material Properties
0.0 58.0 36.0 1.0 16.3 0.3 29000.0 36.0
1.0 179.0
```

5. Mesh Variables

Title – The title must be on one line and cannot exceed 90 characters in length.

Number of Nodes – The number of nodes in the mesh.

Number of Elements – The number of quadrilateral elements in the mesh.

Mesh Input File Name – The file name of the generated mesh to be used in the model. Note carefully that the input file name must be on a separate line all by itself as shown in the example below. The required generated mesh file format is described later. Note that this mesh is used mostly to have node coordinates provided in a uniform pattern over the domain and for plotting purposes.

Example

```
Mesh Variables
357 300
beam357.dat
```

6. Displacement Boundary Conditions

Title – The title must be on one line and cannot exceed 90 characters in length.

Number of Active Nodes – The number of individual nodes at which the user wants to specify displacement boundary conditions. If no active nodes are to be specified use a value of zero.

Active Nodes – Active node numbers must be given. The quantity of active node numbers must match the value of *Number of Active Nodes*. If zero active nodes are specified, proceed to the next item of input.

Number of Active Lines – The number of lines at which all nodes along each line are to be marked as active and hence can have displacement boundary conditions prescribed. This allows a whole edge of a model to be selected by a line along the edge. This is helpful when a whole edge of nodes is to be pinned or to have rollers. At least one active line must be specified even if the line crosses no nodes.

Active Lines – Active lines must be given as two pairs of coordinates. In other words, a start coordinate for the line and an end coordinate for the line. The quantity of active lines provided in this manner must match the value of *Number of Active Lines*.

Active x displacement – *All* active nodes (individual or along active lines) will be marked as zero displacement nodes or as free nodes in the x direction. A value of 0 indicates a zero displacement node and a value of 1 indicates a free node.

Active y displacement – *All* active nodes (individual or along active lines) will be marked as zero displacement nodes or as free nodes in the y direction. A value of 0 indicates a zero displacement node and a value of 1 indicates a free node.

Number of Displacement Nodes – The number of individual nodes at which the user wishes to specify displacement boundary conditions. These nodes *must* be in the list of nodes indicated previously as either an active node or on an active line. The displacement boundary conditions specified on these nodes supersede any displacement boundary conditions previously made on these individual nodes (such as in the active x or active y displacement specification). Note that the following three variables must be given for each displacement node (the first triple must be given, then the next, and so on, see the example).

Node Number – The node number of a node at which displacement boundary conditions are to be given.

X displacement – The value of the specified displacement in the x direction. A value of 9999.0 specifies that the node be free in the x direction.

Y displacement – The value of the specified displacement in the y direction. A value of 9999.0 specifies that the node be free in the y direction. (Note: for the co-rotational displacement control version of EFGAP the x and y displacement specification must be either 0.0 for fixed or 9999.0 for free, i.e. nonzero displacements are not allowed.)

Example

```
Displacement Boundary Conditions
2
10 98
1
10.56 -1.0 10.56 250.0
1 1
2
10 0.0 0.0
98 0.0 9999.0
```

7. Traction Boundary Conditions

   Title – The title must be on one line and cannot exceed 90 characters in length.

   Number of Traction Paths – The user must specify the number of paths along which tractions are specified.

   Number of Traction Nodes Along Path 1 – The number of nodes along traction path number 1.

   Direction of Traction at Path 1 – Traction points along the x axis for a value of 0 and along the y axis for a specified value of 1.

   Traction Type at Path 1 – A value of 1 is a uniform traction, a value of 2 gives a parabolic traction.

   Traction Load Factor at Path 1 - A load factor with a plus or minus sign to orient it in the positive or negative sense. The total force of the traction will be P_{max} if the load factor is 1.0 (as long as collapse does not prevent P_{max} being reached).

   .

   .

   .

   Number of Traction Nodes Along Path n – The number of nodes along traction path number n.

Direction of Traction at Path n – Traction points along the x axis for a value of 0 and along the y axis for a specified value of 1.

Traction Type at Path n – A value of 1 is a uniform traction, a value of 2 gives a parabolic traction.

Traction Load Factor at Path n - A load factor with a plus or minus sign to orient it in the positive or negative sense. The total force of the traction will be P if the load factor is 1.0 (as long as collapse does not prevent P being reached).

Node 1 path 1 to Node n path 1 – The sequence of nodes in path 1.

.

.

.

Node 1 path n to Node n path n – The sequence of nodes in path n.

Example

```
Traction Boundary Conditions
2
3 1 1 -1.0
3 0 1 1.0
104 105 106
7 129 5
```

8. Voronoi Diagram Generation Variables

Title – The title must be on one line and cannot exceed 90 characters in length.

Number of perimeter nodes – The user must provide a sequence of perimeter nodes. These nodes must be in CCW order around the model perimeter. The lines connecting these nodes, as they proceed CCW around the model, must past through all nodes on the boundary of the domain. Hence not all the nodes on the boundary must be included in this number of perimeter nodes. This is just the number of nodes sufficient such that the lines connecting these perimeter nodes pass through all boundary nodes. During preprocessing boundary nodes on these perimeter lines are automatically determined.

Perimeter node number 1 – The first node in the sequence.

.

.

.

Perimeter node number n – The nth node in the sequence.

Perimeter node number 1 – The first node in the sequence is repeated.

Example

```
Voronoi Diagram Generation Variables
9
0 1 3 6 8 9 5 4 0
```

9. Thickness and Special thickness lines

Title – The title must be on one line and cannot exceed 90 characters in length.

Overall Thickness – A constant thickness initially specified over the entire domain.

Number of thickness lines – The number of lines along which nodes are to have a thickness specified different than the overall thickness.

Node Number 1 Line 1 – The first node number of a line along which a special thickness is to be specified.

Node Number 2 Line 1 – The second node number of a line along which a special thickness is to be specified.

Special thickness Line 1 – The thickness specified for all nodes along line 1 including start and end nodes.

.

.

.

Node Number 1 Line n – The first node number of a line along which a special thickness is to be specified.

Node Number 2 Line n – The second node number of a line along which a special thickness is to be specified.

Special thickness Line n – The thickness specified for all nodes along line n including start and end nodes.

Example

```
Thickness and Special thickness lines
0.545
10
8 12 3.0
0 5 4.01
5 6 4.01
6 7 4.01
2 4 4.01
1 4 4.01
70 74 2.6238
34 38 2.6238
231 239 1.0
239 247 1.0
```

10. Read from data files directives

Title – The title must be on one line and cannot exceed 90 characters in length.

$b_{Ii}(\boldsymbol{x}_L)$ flag – 0 means do not read $b_{Ii}(\boldsymbol{x}_L)$ data file, 1 means read the $b_{Ii}(\boldsymbol{x}_L)$ data file. Always use 0 for this flag.

ϕ flag – 0 means do not read ϕ data file, 1 means read the ϕ data file. Always use 0 for this flag.

Example

```
Read from data files directives
0 0
```

11. Read displacement increments (*if displacement control version of EFGAP used). If displacement control is being used the number of load increments is not used. Instead the number of displacement increments controls the incremental iterative analysis. The incremental displacements control the displacement node in the degree of freedom direction specified. Also, P_{max} is in general not reached, but rather the maximum value of P is dependent on the final specified displacement at the displacement node. The input required is as follows.

    Title – The title must be on one line and cannot exceed 90 characters in length.

    Number of displacement increments – The number of displacement increments to be read from the displacement increment data file.

    Increment increase factor – The displacement increment data file may have for example 10 displacement increments. The increment increase factor should be 1 if no change to the increments is desired. However, if the 10 increments are to be cut in half the increment increase factor should be 2. This factor allows the user to basically divide up the displacement increment data file into finer increments, rather than having to edit the data file. The displacement increment factor should be an integer greater than or equal to 1.

    Name of displacement increment data file – A text file with one displacement increment per line in the file.

    Example

```
Read file of displacement increments
30 1
dispincs.txt
```

12. Read constant traction input (*if displacement control co-rotational version used)

Constant tractions may be specified at certain nodes. These tractions are applied at the beginning of the analysis and are held constant throughout the nonlinear analysis procedure. The input required is as follows.

Title – The title must be on one line and cannot exceed 90 characters in length.

Number of constant traction paths – The number, k, of constant traction paths to be specified.

Number of load increments – The number of load increments used to apply the traction paths. Ideally this could be performed in one step, since the total traction loads should not exceed yield. However, within a co-rotational formulation the second order effects sometimes dictate that several load increments for application of constant tractions is better.

Number of traction nodes for path 1 – The number of nodes along the path of traction 1.

Constant traction direction for path 1 – The direction of the traction for traction path 1. A value of 0 is for x direction and value of 1 is for y direction.

Constant traction type for path 1 – A value of 1 gives a uniform traction. A value of 2 gives a parabolic traction.

Load factor for path 1 – A load factor for the traction along load 1. The load factor here is not dependent on the P_{max} value previously input. The load factor here should have the correct magnitude and sign so that the load factor equals the desired total load caused by the traction.

.

.

.

Number of traction nodes for path k – The number of nodes along the path of traction k.

Constant traction direction for path k – The direction of the traction for traction path k. A value of 0 is for x direction and value of 1 is for y direction.

Constant traction type for path k – A value of 1 gives a uniform traction. A value of 2 gives a parabolic traction.

Load factor for path k – A load factor for the traction along load k. The load factor here is not dependent on the P_{max} value previously input. The load factor here should have the correct magnitude and sign so that the load factor equals the desired total load caused by the traction.

Node 1 path 1 to Node n path 1 – The list of node numbers in order for traction path 1.

.

.

.

Node 1 path k to Node n path k – The list of node numbers in order for traction path k.

Example

```
C-tractionBCs
1 1
5 0 1 -9.57
14 9 5 2 0
```

Mesh Input File

The mesh input file must be in the following format.

Mesh input file in variable form example

```
Number of Nodes      Number of Elements
Number of Dimensions(2 in this case)
Node 1 xcoordinate ycoordinate
.
.
.
Node n xcoordinate ycoordinate
Element 1 element type node1 node2 node3 node4
.
.
.
Element n element type node1 node2 node3 node4
```

Note that element nodes must be consistently in CW order or consistently in CCW order. The node numbers and element numbers in the mesh file must start at 1. EFGAP will automatically put the element nodes into CCW order regardless of the order given in the mesh input file. EFGAP also renumbers the nodes and elements starting at 0. Always use element type of 1 (although it probably doesn't matter since this value is not used but must be present in the input file). Below is a numerical example of a mesh file.

Mesh input file example (6 nodes and 2 elements)

```
6   2
2
1 0.0 0.0
2 2.0 0.0
3 4.0 0.0
4 0.0 2.0
5 2.0 2.0
6 4.0 2.0
1 1 1 2 5 4
2 1 2 3 6 5
```

Mistakes to Avoid

1. Mesh file node numbers start at 1. However, in the EFGAP main input file node numbering starts at 0. Hence node 1 in the mesh file is node 0 in the EFGAP input file.

2. Mesh file element numbers start at 1. However, in the EFGAP main input file element numbering starts at 0. Hence element 1 in the mesh file is element 0 in the EFGAP input file.

3. Setting P_y too high so that the first load step causes yield. This will often cause immediate collapse and the program terminates with no displacement calculations provided.

4. Use thplot.m to verify that model thicknesses are specified correctly

5. Be sure to input the correct boundary for the model so that the Voronoi diagram is generated correctly.

Error Messages

1. A matrix is singular. Possible Solution: The variable α should be made larger.

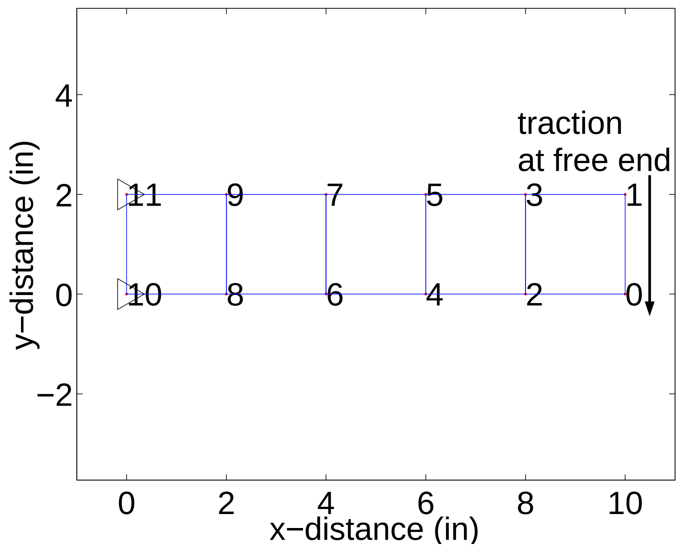


Figure G.1: Example model for example input file.

2. Edge list full. Edge list should be checked. Voronoi diagram generation has likely failed. There is likely an error in the input. Plotting the Voronoi diagram may lend some clues to the problem.

3. If no error message is provided by EFGAP, and the system crashes, it is very likely an error in the input file. The user should check the input file carefully. Plot preliminary output if available.

An Example Model and Input File

An example input file, for the case of load control, is given next for the model shown in Figure G.1. A twelve node cantilever is loaded with a downward vertical load at its free end (at right). At the left end of the cantilever nodes 10 and 11 are pinned to create fixity.

Start of Input File Example

```
Title - Test Cantilever 12 Nodes (12-5-07)
MLS Variables (alpha)
```

```
1.0
Analysis Variables(nldincs,Pload,Py,clpseresid,maxiter,eqtol,dispnd,dof)
20 20 0.5 1.0e6 10 1.0e-2 1 1
Matl Props(Hbar,Kinf,Ko,theta,delta,nu,E,sigmayield,sfactor,mus)
0.0 55.0  36.0  1.0 18.0 0.0 10000.0 36.0
1 171.0
Mesh Variables(ngnodes,nelems,mfilename)
12 5
cant12.m
Disp BC's(nanodes,anodes,nalines,alines,axdisp,aydisp,ndispnds,nnum,dx,dy)
0
1
0.0 0.0 0.0 6.0
0 0
0
Traction BC's(ntrcpaths,ntrctnodespk,tdirectpk,ttypepk,Lfact,nd1pk..ndnpk)
1
2 1 1 -1.0
0 1
Voronoi Diagram Generation(npnds,nodes in ccw order, 1st & last node same)
5
0 1 11 10 0
Thickness & Special thickness lines(thickness,nlines,node1Lk,nodenLk,thLk)
2.0
1
0 1 2.0
Read from data files directives(read bIi_xL, read nodal phi vals,0 n 1 y)
0 0
```

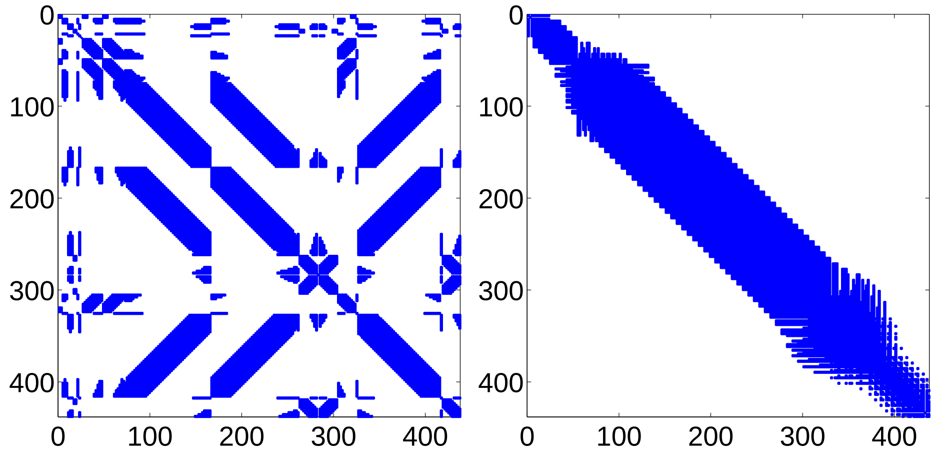
End of Input File

G.1.2 Preprocessing

Preliminary calculations that are not repeated throughout the analysis are completed first. Many of these results are stored for use during the analysis phase, although some are used immediately during preprocessing. Unless noted otherwise the preprocessing is accomplished entirely within the analysis program.

Renumbering, by the reverse Cuthill McGee (RCM) algorithm, of the generated nodes is one item of preprocessing which is done outside of the analysis program. This preprocessing is essential for two reasons. First, renumbering of the nodes by the RCM algorithm causes the stiffness matrix to have a significantly reduced bandwidth. Solution of the resulting system of algebraic equations is more computationally efficient because the solution process using a standard LU decomposition method can take advantage of the banded nature of the stiffness matrix. Figure G.2a illustrates the location of nonzero stiffness matrix (438x438 size) entries for a meshfree model with 219 nodes in which no care is taken to renumber the model nodes. Figure G.2b illustrates the same meshfree model stiffness matrix after using the RCM node renumbering scheme. The banded nature of the resulting matrix is evident. As a result of the renumbering, the solution of the algebraic equations for this example is almost 18 times faster, which is a significant computational improvement. The second advantage of the renumbering is that only the banded portion of the stiffness matrix is stored in memory. In this example, the banded matrix requires approximately one fifth the memory required for the full 438x438 matrix. These computational and memory advantages become even more pronounced as the matrix size increases. In fact, if these issues are not taken into account a nonlinear analysis soon becomes computationally prohibitive and perhaps impossible due to memory limitations. For these reason a banded storage LU decomposition solver was created as part of this dissertation. A C++ implementation of the RCM algorithm is available at `http://people.scs.fsu.edu/~burkardt/cpp_src/rcm/rcm.html`.

The only other part of preprocessing not done within the analysis program is the mesh generation. This is done with Professor Mark Rashid's simple quadrilateral mesh generator. Several comments are worth noting regarding the mesh. Originally a quadrilateral mesh was used in conjunction with the MLS basis functions and the method of continuous blending to enforce essential boundary conditions. Hence, there was a dependence on having a mesh. The max-ent basis functions allow easier imposition of essential boundary conditions and hence do not require the method of continuous blending. So, even though a mesh is still



(a) (b)

Figure G.2: Plot of nonzero entries in example 438x438 stiffness matrix: (a) results without renumbering of nodes; (b) results with renumbering of nodes by the Reverse Cuthill McGee algorithm.

a required part of the input, the program is mesh independent for the following reasons. Element connectivity and the relationship between nodes and elements for the mesh is never used inside the analysis program. Furthermore, the benefit of using a quadrilateral mesh is three-fold. First, the mesh is helpful for discretizing the domain and observing the aspect ratio of the node spacing. If, for example, one observes that in the x direction the spacing is much larger than in the y direction it can explain why the program may fail to run. This may be due to the following reason. If the aspect ratio is severe the set of nearest neighbors might all be along a vertical line and the max-ent shape function generation routines fail. Or if they do not fail the global stiffness matrix may not have sufficient interdependence along the x direction. If this happens the solution is to increase the support radius parameter α or change the mesh to have an aspect ratio closer to 1.0, which tends to be more efficient. Second, triangles are easily generated from the quadrilateral elements merely by adding diagonals. The resulting triangular data structure is then in a usable form for input to the reverse Cuthill McGee algorithm. Third, it is helpful to plot the mesh grid in the deformed configuration and verify that spurious modes are avoided. Essentially, it is a visualization tool.

Some output to file is written during preprocessing. Hence if the program crashes during preprocessing sometimes it is helpful to look at the preliminary output which lends clues to errors in the input. Specifically, "xnodes.txt" provides the nodal coordinates, "elements.txt" provides the element connectivity, and "th.txt" provides the specified thicknesses for every node in the domain. Matlab scripts are used to load these files and plot information to visually verify correct input.

The support radius is generated for every node in the domain. It is somewhat arbitrary, but it is computationally best to keep the support radius as small as possible. In this dissertation the support radius for a particular node is chosen as α times the distance to the fifth nearest neighboring node.

The elastic modulus matrix is generated for the case of plane stress. This is later used to initialize the stored modulus matrix for every node in the domain.

The generation of the Voronoi diagrams is completed by a Voronoi diagram generator provided on the internet by O'Sullivan (`http://www.skynet.ie/%7Esos/masters.htm`). The program in its given form only generates a Voronoi diagram for the nodes. The output consists of the list of Voronoi diagram lines or cell edges. Hence, to make this information useful the following items are implemented.

1. The user input boundary of the discretized domain clips any Voronoi diagram edges that cross it and the list of edges is augmented to include the boundary edges. This is necessary for every cell on the boundary.

2. For each node a list of Voronoi cell edges is generated in counter-clockwise order.

3. The length of every Voronoi cell edge is calculated and stored.

4. The coordinates of the second endpoint of every Voronoi cell edge in counter-clockwise order is determined from the list of edges for every node.

5. The vector components of the unit normal to every Voronoi cell edge is generated in counter-clockwise order.

6. The Voronoi cell area of every node in the domain is calculated. This is done as follows. Each Voronoi cell edge and the node associated with the Voronoi cell make a triangle. The Voronoi cell area is then just the sum of triangular areas. For an example Voronoi cell with triangular subcells, see Figure 4.3(b). The formula for area of an arbitrary triangle in space is $A = (s(s-a)(s-b)(s-c))^{0.5}$, where a, b, c are the three triangle side lengths and $s = 0.5(a+b+c)$.

7. The coordinates for the centroid of every Voronoi cell is calculated. This is accomplished by the following method. Each Voronoi cell has N vertices. Let (x_i, y_i) be the coordinates for vertex i. Upon defining the vertex coordinates (x_N, y_N)=(x_0, y_0), the x and y coordinates of the centroid are found by the following expressions

$$C_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \quad C_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \tag{G.1}$$

   where A is the Voronoi cell area.

8. The Voronoi diagram edges are written to the file “voronoi.txt”. If necessary a matlab script is used to plot the diagram to verify it is generated correctly.

9. The Voronoi cell centroid coordinates are also written to the file “voronoicent.txt”.

With the exception of Voronoi cell centroid, all of these items associated with the generated Voronoi diagram are required for nodal integration. It took a significant amount of work to generate the information indicated above. A significant and difficult amount of computational geometry is associated with matching Voronoi edges to nodes and constructing the list of edges in counter-clockwise order for each node. It was also very challenging to clip the Voronoi edges that crossed the boundary and augment the edge list with boundary edges. It does not appear that a Voronoi diagram generator exists that already provides all of this information in an organized form.

Constructing a neighbor list for every node in the domain is beneficial for bookkeeping, computational efficiency and reduction of memory use. If such a list is known for every node only those nodes need be used during construction of the stiffness matrix. Furthermore, only the b_{Ia} values in the neighbor list associated with the nodes need be stored in memory. Thus a neighbor list and number of neighbors for each node allows for a helpful bookkeeping scheme. The scheme implemented for constructing the neighbor list, for a given node L, is as follows. First, determine the distance, d_{fv}, from node L to its farthest Voronoi cell vertex. Second, check every node M in the domain and include node M as a neighbor if the support radius, ρ_M plus d_{fv} is greater than the distance, d_{LM}, from node L to node M (see Figure G.3). This scheme is guaranteed to work. Certain nodes, that are not true neighbors, are likely included in this scheme because of the liberal use of the distance to the *farthest* Voronoi cell vertex. However, these extra "neighbors" are few, have minimal effect on computational efficiency and have zero contribution when assembling the stiffness matrix.

The $b_{Ia}(\boldsymbol{x})$ values are generated and stored for every node in the domain based on integration around the Voronoi cells. These are used every time the stiffness matrix is generated during the analysis. The $b^c_{Ia}(\boldsymbol{x})$ values are generated and stored for every triangle in every node's Voronoi cell. These values are used to stabilize the stiffness matrix every time it is generated.

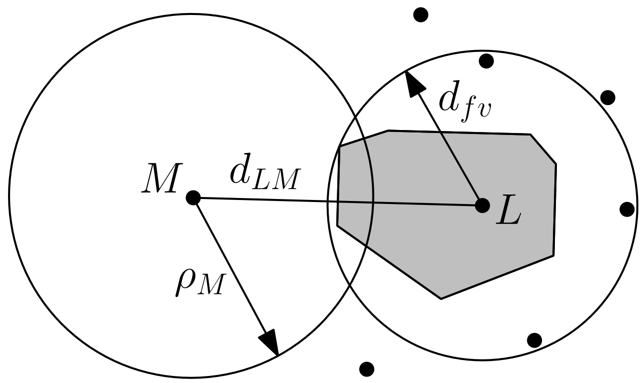


Figure G.3: Geometry used to determine list of neighbors for node L.

The total force vector is created. This vector contains the forces due to the tractions if the entire load P_{max} is applied to the structure. In fact, for the load control scheme all traction magnitudes are referred to P_{max} and are scaled as necessary by the traction load factor. For the displacement control scheme the total force vector is still constructed, but the final load is dependent on the specified displacement control at the chosen degree of freedom of a selected node.

G.1.3 Analysis

The nonlinear analysis takes place within the analysis phase. After initialization of the necessary variables the analysis begins with the loop over each load step. The basic algorithm is equivalent to that given previously in Section 5.2.5 and hence is not repeated here. Essentially the analysis provides an implicit Newton-Raphson scheme at the global level and an implicit Newton-Raphson scheme with radial return at the constitutive level when inelastic material behavior is included. During the analysis information is printed to screen to indicate progress. Information written to screen sometimes provides clues to the problem if the program crashes. During the nonlinear analysis the current load step number is printed to screen followed by the global equilibrium residual during each iteration. After the iterations for a given load increment satisfy the equilibrium tolerance the number of iterations required is printed to screen. This allows the user to observe the rate of convergence and observe when convergence problems possibly arise. The time required for the analysis is output to screen at the end. Until the user hits return, it is possible to scroll back up in the console window and look over various phases of the analysis.

G.1.4 Postprocessing

The following standard output of results is currently part of the analysis program. The load versus displacement response is written to the file "pvsu.txt". Stresses and strains at the end of the analysis are given in the files "sigmaxx.txt" and "strainxx.txt". The final x and y nodal displacements are written to the file "disp.txt".

Given the pre and post processing output the following information is plotted by using matlab scripts.

1. Load versus displacement (nplot.m)
2. Structure deflected shape (eplot.m)
3. Stresses, σ_{xx}, σ_{yy} and σ_{xy} (splot.m)
4. Strains, ε_{xx}, ε_{yy} and ε_{xy} (explot.m)
5. Voronoi diagram (vplot.m)
6. Thickness plot (thplot.m)