

MASTER THESIS

Implementation of a Fiber Beam Finite Element
for the Simulation of Steel-Reinforced
Concrete Beam Under Dynamic Loading

Mahmoud Zidan

Implementation of a Fiber Beam Finite Element for the Simulation of Steel-Reinforced Concrete Beam Under Dynamic Loading

Submitted to the Department of Civil, Geo and Environmental Engineering
in partial fulfillment of the requirements for the degree of
M.Sc.
at the Technical University of Munich.

Supervised by MSc. Long Chen
MSc. Klaus Sautter
Prof. Dr.-Ing. Kai-Uwe Bletzinger
Chair of Structural Analysis

Submitted by Mahmoud Zidan
Biedersteiner Str. 24
80805 München
Student number: 03693907

Submitted on Munich, 21.11.2019

Abstract

The simulation of reinforced concrete elements due to a seismic loading brings down the cost of physical experiments, however involves high non-linearities. Many implementations to tackle these problems have been established through the recent years. This thesis work shows the implementation of a flexibility-based fiber beam-column element to model the response of a reinforced concrete element under a cyclic loading in the post-yielding phase. The theory behind the element is discussed as well as the algorithm to carry on the solution. The element has been implemented in a stand-alone Python code and in the open-source framework KRATOS, which is an open-source FEM framework developed in the chair of Structural Analysis in the Technical University of Munich, Germany, as well as in the International Center for Numerical Methods in Engineering (CIMNE) in Barcelona, Spain.

Keywords

FEM; FEA; Reinforced-Concrete; Beam-Column; Seismic-Loading; Non-linear-FEM; Element Formulation; Computational Earthquake Engineering

Kurzfassung

Die Simulation von Stahlbeton-Strukturen unter seismischen Belastungen erlaubt eine effiziente und kostengünstige alternative Analyse zu herkömmlichen Tests. Dabei müssen jedoch materielle Nichtlinearitäten berücksichtigt werden. Um diese anspruchsvolle Problemstellung zu simulieren wurden in den letzten Jahren verschiedene Algorithmen entwickelt, von denen einer in dieser Arbeit behandelt wird. Ein flexibles Faser-Balkenelement wird implementiert um die Strukturantwort eines Stahlbetonbalkens unter einer zyklischen Belastung zu berechnen. Dabei wird sowohl die zugrunde liegende Theorie als auch der Algorithmus selbst detailliert besprochen. Das Element wurde in einem eigenständigen Python-Code und im Open-Source-Framework KRATOS implementiert, das am Lehrstuhl für Statik in der Technischen Universität München sowie im "International Center for Numerical Methods in Engineering" (CIMNE) in Barcelona, Spanien, entwickelt wird.

Schlüsselwörter

FEM; FEA; Reinforced-Concrete; Beam-Column; Seismic-Loading; Non-linear-FEM; Element Formulation; Computational Earthquake Engineering

Acknowledgements

Firstly, I would like to express my gratitude to both my supervisors, Long Chen and Klaus Sautter, for their continuous support and mentoring throughout this work. Thanks to their many comments, remarks and engagement, this thesis was accomplished. Also, my thanks goes to Philipp Bucher for his help in the KRATOS implementation. Acknowledgements also go to my colleagues and friends in the Computational Mechanics program for the inspiring discussions and ideas.

Moreover, my appreciation goes to Prof Dr-Ing Kai-Uwe Bletzinger and Dr.-Ing. habil. Roland Wüchner for giving me the opportunity to do my Master thesis at the chair of Structural Analysis at the Technical University of Munich.

Lastly and most importantly, I would like to thank my family for their never ending support for my studies. I am in this position because of them standing by my side. My thanks extend to my friends back in Kuwait.

Table of Contents

1	Introduction	1
2	Background Theory	2
2.1	Introduction to Non-Linear Finite Element.....	2
2.1.1	Newton-Raphson	4
2.1.2	Newton-Raphson With Displacement Control Constraint	5
2.1.3	Implementation of Displacement Control in KRATOS	6
2.1.4	Examples On Displacement Control	6
3	The Fiber Beam-Column Element	9
3.1	Brief History of Modeling of Reinforced Concrete	9
3.2	Definition of the Element	10
3.3	Definition of Degrees of Freedom and Forces	11
3.4	Formulation of the Element.....	13
3.5	Uniaxial Fibers.....	15
3.6	Element Solution Algorithm	16
3.7	Detailed Explanation of the Algorithm	21
4	Constitutive Laws	26
4.1	Steel Uniaxial Constitutive Law	26
4.2	Concrete Uniaxial Constitutive Law	28
4.3	Implementation of the Constitutive Laws	29
4.3.1	Algorithm for Menegotto-Pinto	30
4.3.2	Algorithm for Kent-Park.....	32
5	Code Structure	36
6	Numerical Results	38
6.1	Example 1: Moment-Curvature	38
6.2	Example 2: Uni-axial Bending.....	42
6.3	Effect of the Discretization.....	45
7	Conclusions and Outlook	47
A	Appendix	48
A.1	Gauss-Lobatto Quadrature Rule	48

1. Introduction

Motivation

The seismic analysis of the response of structures involves non-linear analysis due to the fact that the loading is cyclic and goes typically beyond the yield point of the elements. The most common structural elements used in building are reinforced concrete elements which consist of concrete reinforced by steel bars, as the concrete is a cheap material that is strong in compression but relatively weak in tension. The element having two types of materials makes it difficult to analyze the structure especially in post-yield response. Historically, tests have been done on reduced scale elements or structure to assess the response and develop hysteretic constitutive laws. On the other hand, computational models allowed the simulation of real scale and more complex problems at a considerably lower cost.

The multi-physics open-source flexible software KRATOS [1] has been used as a simulation tool, as it already contains a framework to simulate non-linear structures. KRATOS is an open-source framework, written in C++ to allow the implementation of numerics to solve engineering problems. It is written in a modular way to include collaboration and development of researchers.

Objective

The objective of this thesis work is to implement the flexibility-based fiber reinforced concrete element to analyze the response of the element to a dynamic loading, changing in time. The application of the load is considered quasi-static, so the non-linearity lies solely in the material laws of the concrete and the steel bars.

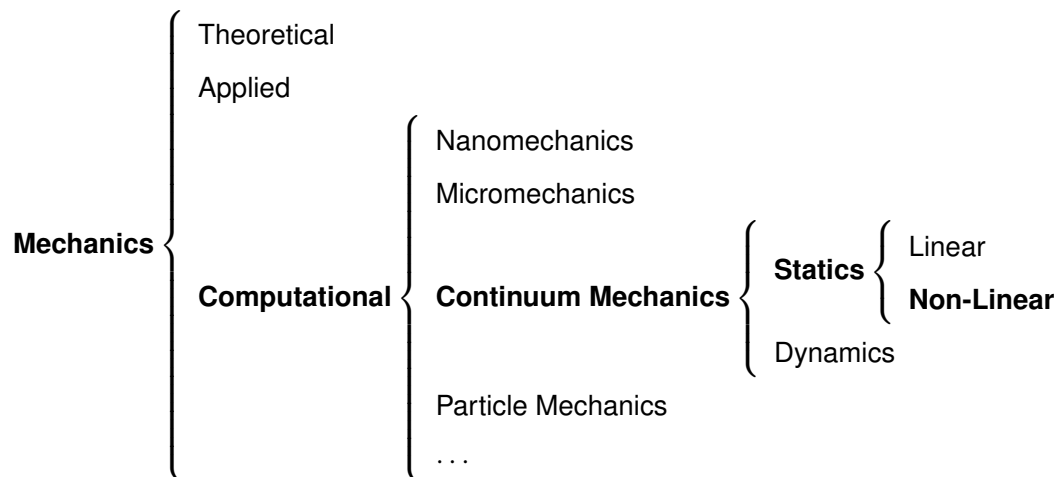
The element is a one dimensional elements that is discretized into sections, each section having fibers. The fibers can have individual uniaxial stress-strain relationships. The element stiffness is derived from the flexibility of the sections. The element implemented is based on the work of Taucer [2], since it is the first original well derived, reliable, and computationally efficient flexibility-based fiber beam-column element.

In chapter 2, the theory of non-linear FEM is introduced to solve the problem in hand, and the Newton-Raphson method is discussed. Also, the implementation in KRATOS is briefly discussed with two problems using non-linear truss elements. In chapter 3, the theory of the flexibility based fiber beam-column element is shown, where the advantages of the current element are discussed. Moreover, the formulation of the element based on the mixed method is derived, and the element solution algorithm is extensively discussed. The constitutive laws used in this element are explained in chapter 4 with their implementation algorithms. Finally, numerical examples are discussed in chapter 6.

2. Background Theory

2.1. Introduction to Non-Linear Finite Element

The Non-Linear Finite Element Method (NFEM) is a method to solve of Computational Mechanics problems. To introduce it, a classification of Mechanics is explained here. Mechanics is the science of Physics that relates physical bodies to forces, displacements, and energy. Mechanics can be subdivided into theoretical, applied or computational, where the problem is solved numerically using a computer in the last one. Computational Mechanics can be subdivided into branches based on the scale of the problem. Here, the interest is Continuum Mechanics, where the material is assumed continuous. Moreover, Continuum Mechanics can be divided into Statics, Quasi-Statics, and Dynamics. Furthermore, the response of the material can be linear or non-linear. The non-linearity of the response could be geometrical (large displacements), physical (material), etc. To solve this non-linear problem, many discretization methods can be applied. Our interest is the Finite Element Method applied to a Continuum Mechanical model of Non-Linear Quasi-Statics. This division of Mechanics is explained in the following: [3]



Essentially, a time invariant problem of Continuum Mechanics can be described by a partial differential equation, which can be solved by means of the Finite Element Method, where the physical space is discretized into elements of the same Parametric Space [4]. One of the methods to derive a Finite Element solution to a problem is the Variational Formulation, which can be solved using the Principal of Virtual Work, where in the weak form,

$$\int_{\Omega} \delta W d\Omega = 0 \quad (2.1)$$

holds, where δW is the variation of the virtual work, and Ω is the solution domain. This generally leads to a linear system of equations,

$$A \cdot u = f, \quad (2.2)$$

where u is the solution to be found. This system of equations can be reorganized to define a residual, r , such that

$$r = A \cdot u - f = 0. \quad (2.3)$$

Using the stiffness-based displacement formulation [3] generally yields

$$\mathbf{r} = \mathbf{K} \cdot \mathbf{u} - \mathbf{f} = \mathbf{0}, \quad (2.4)$$

where \mathbf{r} is the residual vector, \mathbf{K} is the stiffness matrix, \mathbf{u} is the displacement vector, and \mathbf{f} is the force vector. A solution \mathbf{u} is obtained when the residual approaches zero meaning the system is in equilibrium.

The proposed element is solved using Static Non-Linear Finite Element Method (FEM). The non-linearity of the problems lies only in the material law. The concept of solving a non-linear problem using FEM is explained here using the *load-displacement* curve, where the solution is obtained by seeking a point on the equilibrium path. Since the equilibrium path is non-linear, an iterative scheme is used until a point where the residual approaches zero is obtained. A basic example of a non-linear equilibrium path is shown in Figure 1.

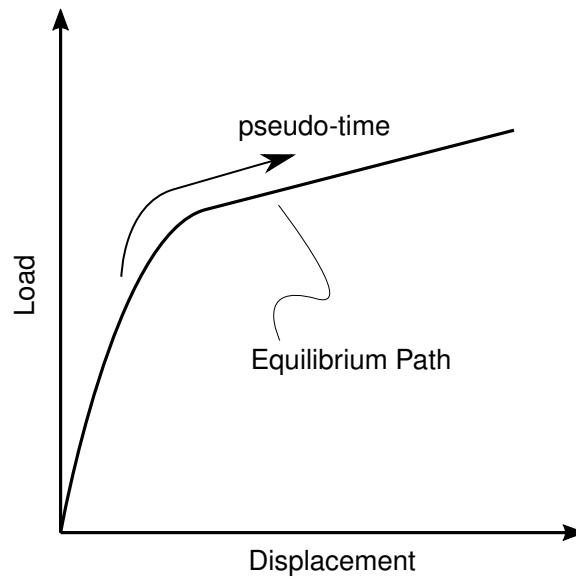


Figure 1 Non-linear load displacement curve example.

2.1.1. Newton-Raphson

The method to find a point on the equilibrium used in this work is the commonly used Newton-Raphson (NR) method, where the problem is linearized and solved iteratively. The methods and the derivations in this chapter are explained in details in the lecture notes by Filippa [3]. Generally, the method is used to solve an equation $f(x) = 0$, where $f(x)$ is a non-linear function. The method linearizes the function using its gradient. In details, the solution can be obtained by iteratively solving the linearized equation $\tilde{f}(x) = 0$ as:

$$\tilde{f}(x) = f(x_k) + (x_{k+1} - x_k) \left. \frac{df}{dx} \right|_{x_k}, \quad (2.5)$$

where k is the iteration number. The only unknown in this equation is x_{k+1} . Hence, the equation is solved iteratively, until $||\Delta x_k||$ is below a certain prescribed tolerance, where $\Delta x_k = x_{k+1} - x_k$.

The Newton-Raphson method can be easily applied to a non-linear Finite Element problem by solving eq. (2.4) iteratively, so that the residual approaches zero. However, since the solution is for an equilibrium point on a non-linear path, it involves solving for displacements as well as loads, so there are more unknowns than equations. Consequently, another equation must be added, namely the constraint equation $C = 0$. Different constraint lead to different approaches. The three main approaches using the constraint are known as *Load Control*, *Displacement Control*, and *Arc-Length Control*. The three approaches solve the equations:

$$\mathbf{r}(\mathbf{u}, \lambda) = 0, \quad (2.6)$$

$$C(\mathbf{u}, \lambda) = 0, \quad (2.7)$$

where \mathbf{u} is the displacement vector and λ is the load factor.

Load Control

The constraint for the load control is:

$$C(\mathbf{u}, \lambda) = \lambda_{k+1} - \hat{\lambda},$$

where $\hat{\lambda}$ is a prescribed load factor.

Displacement Control

The constraint for the displacement control is:

$$C(\mathbf{u}, \lambda) = u_{k+1}^{(i)} - \hat{u}^{(i)}, \quad (2.8)$$

where the superscript is the selected controlled degree of freedom, and $\hat{u}^{(i)}$ is the prescribed displacement of the controlled degree of freedom.

Arc-Length Control

The constraint for the arc-length control is:

$$C(\mathbf{u}, \lambda) = (L_k)^2 - (\hat{L})^2,$$

where \hat{L} is the prescribed arc-length and the arc-length L_k at every iteration is

$$L_k = \sqrt{\Delta \mathbf{u}_k \cdot \Delta \mathbf{u}_k + (\Delta \lambda_k)^2}.$$

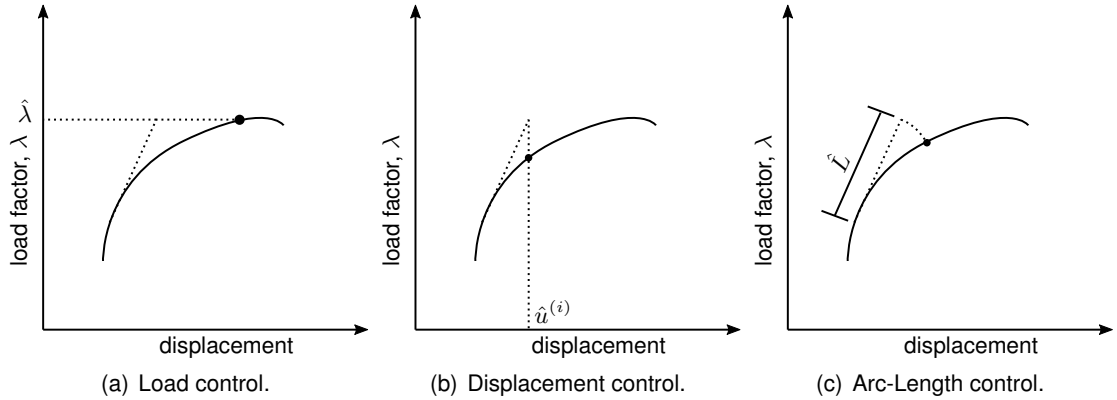


Figure 2 Illustration of the different constraint approaches.

2.1.2. Newton-Raphson With Displacement Control Constraint

The chosen approach for this thesis is the displacement control, since it is able to solve the physics of material softening in concrete fibers, explained later in section 4.2. In this section, the method is derived for non-linear FEM generally. In every iteration, the residual and the constraint are determined on the last iteration values of both the displacements and the load factor, as in:

$$\begin{aligned} \mathbf{r}(\mathbf{u}_k, \lambda) &= \mathbf{K}_k \cdot \mathbf{u}_k - \lambda_k \mathbf{F} = \mathbf{0}, \\ C(\mathbf{u}_k, \lambda_k) &= u_k^{(i)} - \hat{u}^{(i)} = 0. \end{aligned} \quad (2.9)$$

These equation are linearized using eq. (2.5) leading to:

$$\begin{aligned}\tilde{\mathbf{r}} &= \mathbf{r}(\mathbf{u}_k, \lambda_k) + \left. \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right|_k \Delta \mathbf{u}_k + \left. \frac{\partial \mathbf{r}}{\partial \lambda} \right|_k \Delta \lambda_k = \mathbf{0}, \\ \tilde{C} &= C(\mathbf{u}_k, \lambda_k) + \left. \frac{\partial C}{\partial \mathbf{u}} \right|_k \Delta \mathbf{u}_k + \left. \frac{\partial C}{\partial \lambda} \right|_k \Delta \lambda_k = 0,\end{aligned}\tag{2.10}$$

where $\left. \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \right|_k$ is the stiffness matrix at iteration k , \mathbf{K}_k , and $-\left. \frac{\partial \mathbf{r}}{\partial \lambda} \right|_k$ is the external load vector at iteration k , \mathbf{q}_k . Using the displacement control constraint, $\left. \frac{\partial C}{\partial \mathbf{u}} \right|_k$ is a zero vector with 1 at the controlled degree of freedom, and $\left. \frac{\partial C}{\partial \lambda} \right|_k$ is zero. Substituting these values into (2.10) and reorganizing into a linear system of equations at each iteration leads to:

$$\begin{bmatrix} \mathbf{K}_k & -\mathbf{q}_k \\ \left. \frac{\partial C}{\partial \mathbf{u}} \right|_k & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \lambda_k \end{bmatrix} = - \begin{bmatrix} \mathbf{r}(\mathbf{u}_k, \lambda_k) \\ u_k^{(i)} - \hat{u}^{(i)} \end{bmatrix},\tag{2.11}$$

where $\mathbf{u}_{k+1} = \Delta \mathbf{u}_k + \mathbf{u}_k$ and $\lambda_{k+1} = \Delta \lambda_k + \lambda_k$.

2.1.3. Implementation of Displacement Control in KRATOS

The non-linear residual solver that solves eq. (2.4) is already implemented in KRATOS, so the displacement control is implemented as a condition. A degree of freedom (DoF), the load factor λ , is added to the node upon which the displacement control is applied. Then, the local system, the Left Hand Side (LHS) and the Right Hand Side (RHS) of the condition are assembled to the global system. Essentially, the displacement control condition consists of two DoFs, the controlled DoF and the load factor. So, the local system is

$$\text{LHS} = \begin{bmatrix} 0 & -F \\ 1 & 0 \end{bmatrix}, \quad \text{RHS} = \begin{bmatrix} \lambda_k \cdot F \\ \hat{u}^{(i)} - u_k^{(i)} \end{bmatrix},$$

where F is the applied external force. In this local system, the first DoF is the controlled DoF and the second is the load factor on this DoF.

2.1.4. Examples On Displacement Control

The non-linear displacement control is demonstrated on the two snap-through problems, one is a the benchmark case of two trusses snap-through modeled with one bar and a symmetry condition shown in Figure 3(a) [5] and a more complicated case of the two dimensional circular arch of trusses example introduced by [6] shown in Figure 4, where the trusses are geometrically non-linear "Truss3D2N" in KRATOS.

Two Bar Truss System

The system shown in Figure 3(a), is made using a geometrically non-linear truss element with Young's Modulus of 210×10^9 and a cross-sectional area of 0.01. The solution using the load control constraint and the displacement control constraint. The displacement control solution shows the reduction of the stiffness matrix throughout the snap-through, so it describes the effect more accurately.

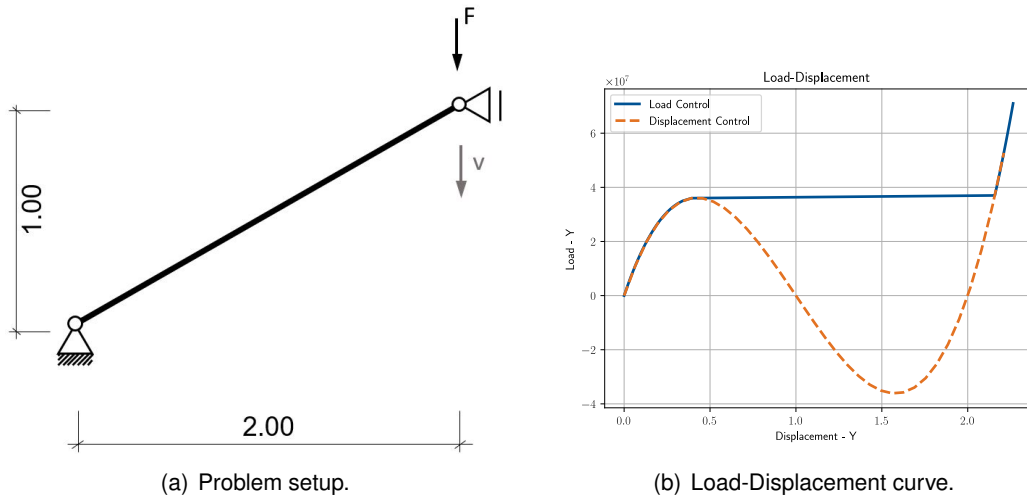


Figure 3 Two trusses snap-through problem. [5]

Two-Dimensional Circular Arch of Trusses

and Young's modulus is 5×10^9 and the area is 0.01. The solution using displacement-control and load control are shown in Figures 6 and 5. In this example, also, the displacement control describes more accurately the snap-through effect and the reduction of the stiffness matrix.

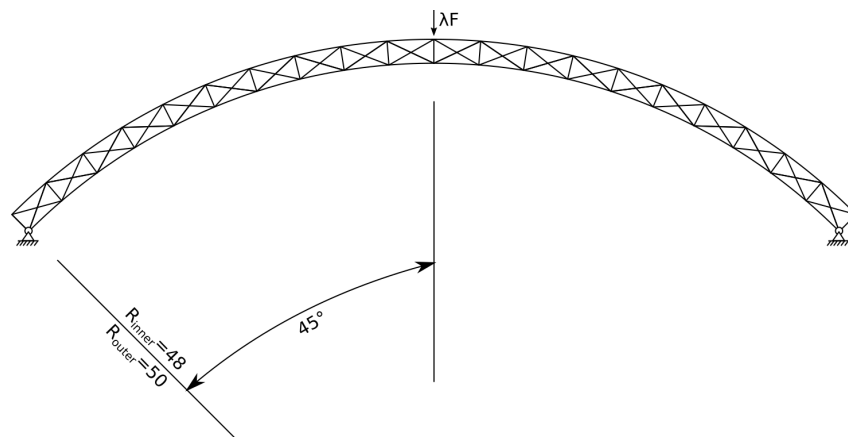


Figure 4 Circular Arch truss system.

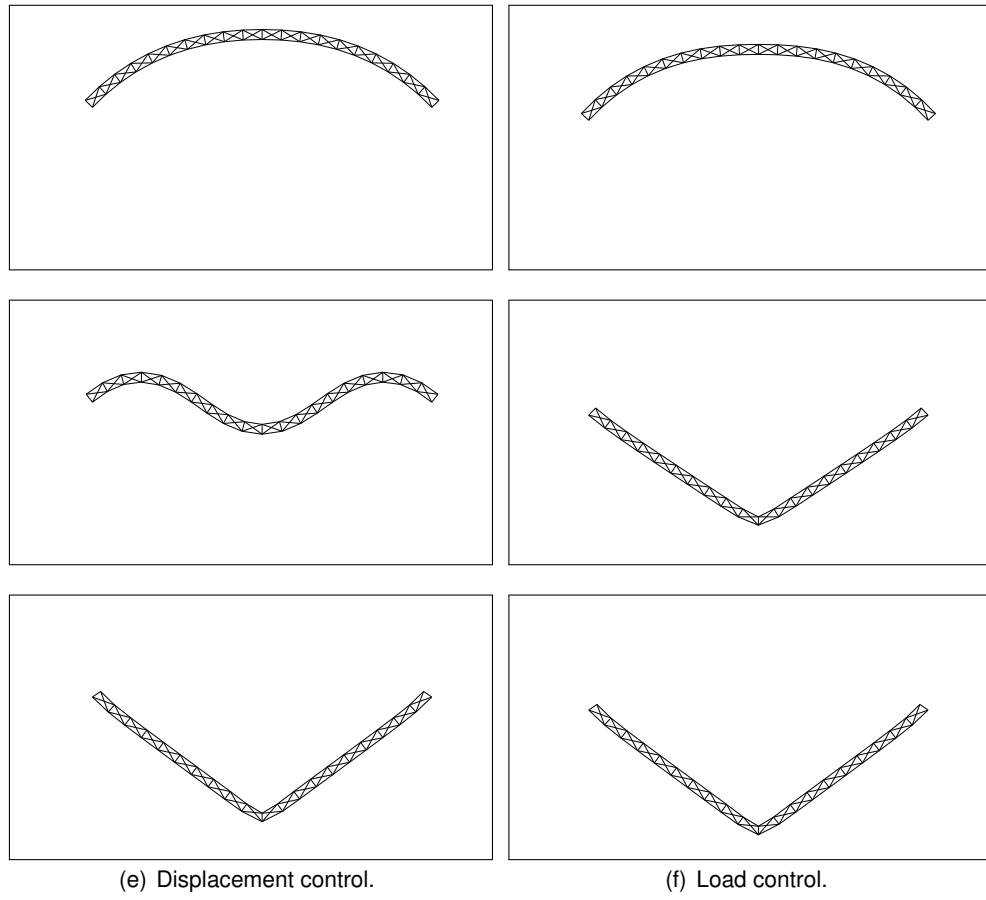


Figure 5 Evolution of the system.

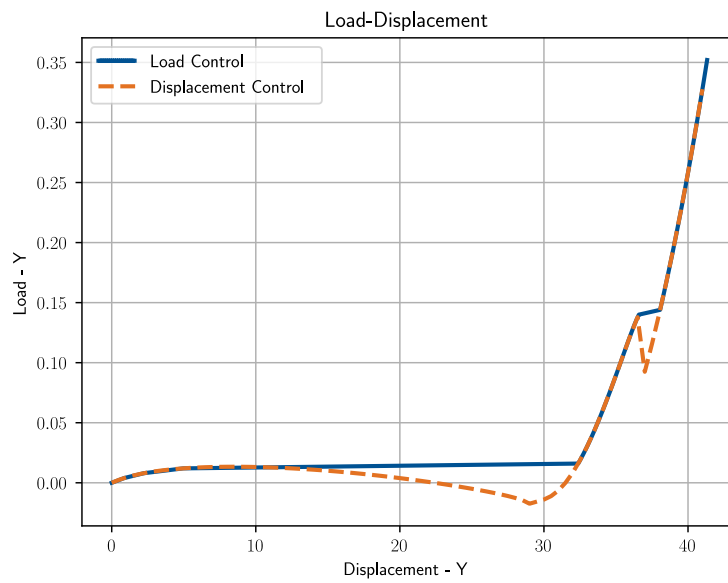


Figure 6 Solution of circular Arch truss system.

3. The Fiber Beam-Column Element

In this chapter, the theory of the flexibility-based fiber beam-column element is discussed. First, the definition of the element, its degrees of freedom, and the formulation are presented, where the necessary equations are derived. Then, the load stepping algorithm is discussed extensively.

3.1. Brief History of Modeling of Reinforced Concrete

Throughout the history of FEM, many studies have been made to model reinforced concrete columns under dynamic loading. Firstly, the **lumped models** were first introduced in [7], where plastic hinges at the ends of the beam element were used to model bilinear moments. Then, the element was improved in [8] to include multi-linear material behavior. Since then, many improvements have been made on the lumped model, however, the inelastic deformation is only modeled at the ends of the element. Secondly, **distributed non-linearity models** were introduced in [9], where the inelastic behavior is more accurately modeled throughout the element, as the element essentially consists of cantilever beams, and the fixed ends have inelastic rotational springs. One assumption is that the axial forces and the bending moments are decoupled in this element. The inelasticity is solely modeled by the inelastic springs. Next, an element that consists of multiple subdivided elements connected by inelastic rotational springs was developed in [10], where static condensation is used to derive the stiffness of the element. Thirdly, the early **flexibility-based fiber model**, depicted in Figure 7, was proposed in [11], where the element consists of longitudinal uni-axial fibers. The element being flexibility-based basically means that it derives its stiffness from the flexibility matrix of the fibers using force interpolation functions. The main advantage of the flexibility-based approach is that the internal forces are exact regardless of the system non-linearities (as shown later in section 3.4), leading to fewer elements needed to represent material non-linearity. Consequently, the softening behavior can be solved in this approach, in contrast to stiffness based elements which are less numerically stable while the material is softening due to negative entries in the stiffness matrix. Moreover, the element-loads application can be direct using the internal force interpolation functions [2]. The first element in [11] models only uni-axial bending moments, which gave accurate results but had problems in convergence and did not model any material softening. The model has been improved in [12] to include biaxial bending moments, where the material softening was described and the non-linearity of the curvature along the element was modeled. However, the element was an improvement to [11] and was rather ad hoc, thus the theory is not general enough for further development. The element proposed by [2] is the first original element to have a clear and a well derived theory, and includes axial forces. Hence, it was chosen to be implemented in this thesis work. In the next sections, the element is described and the solution algorithm is discussed.

3.2. Definition of the Element

The flexibility-based fiber element introduced in [2] is implemented through this thesis in the open-source project KRATOS. This element is implemented to solve deformations that induce static axial loads and bi-axial moments. Hence, the shear forces are beyond the scope of this study. The element is formulated based on the flexibility approach so it handles the non-linear curvatures as well as material softening.

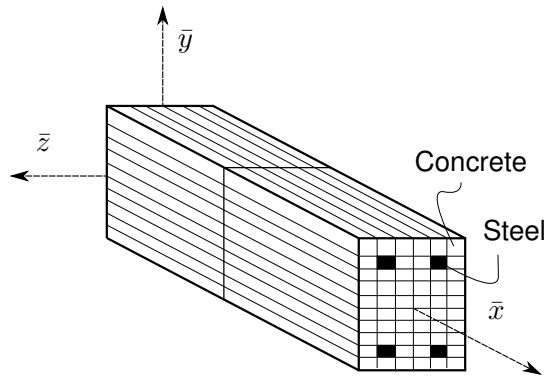


Figure 7 Discretization of the fiber beam-column element.

The discretization of the element is shown in Figure 7. The element is a one dimensional line in a 3D space cut into “control sections.” Each section is discretized in a 2D grid of uni-axial fibers. Each fiber has its own uni-axial constitutive law, and the response of the section is derived from the integration over the fibers. Each section is given a position on the element as well as a weight, in order to integrate over sections to get element properties, based on the Gauss-Lobatto rule. The difference to the conventional Gauss-Legendre quadrature rule, is that the end points are included here, and it is accurate for polynomials of degree $2n - 1$. The advantage to Gauss-Lobatto is that it includes the fixed-ends where the curvature is highly non-linear. More on this quadrature rule is in section A.1 in the appendix.

It is noteworthy to mention the proposed element assumes small deformation and plane sections during the deformation. Moreover, the mixed formulation [13] is used, where both force interpolation functions and deformation interpolation functions are used. However, a special choice of deformation interpolation functions yields a flexibility-based approach, but the derivation is based on the mixed approach to keep it in a general sense for a possibility of further improvements. The following derivations are based on the work of Taucer, Spacone, and Filippou in 1991 [2].

Advantages of the Current Element

The advantages of the element can be summarized in:

- It can model non-linear curvature along the element as well as material softening, because of the use of the flexibility approach and the displacement control non-linear FEM.
- The Gauss-Lobatto includes the end-points where the curvature is highly non-linear.
- The element response is derived from uni-axial fibers that can have a constitutive law independently.
- The generality of its derivation allows further improvements.

3.3. Definition of Degrees of Freedom and Forces

The element has globally 12 degrees of freedom, as it consists of two nodes, each node having three displacements and three rotations. The global forces vector, \mathbf{P} , is defined as the axial forces followed by the moments of each node, and the global degrees of freedom vector, \mathbf{p} , is defined as the displacements followed by the rotations of each node. Locally, the element has five degrees of freedom, one axial, and four bending moments. The local forces of the element are shown in Figure 8.

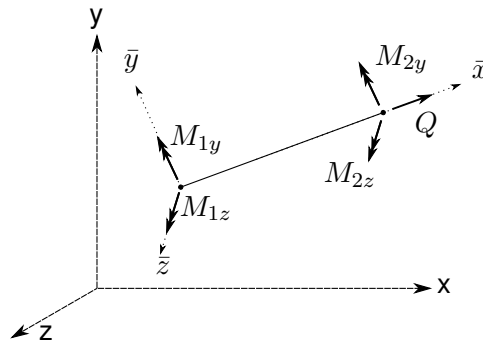


Figure 8 Element local forces.

The local force vector \mathbf{Q} of the element and local displacement vector \mathbf{q} are defined as:

$$\mathbf{Q} = \begin{bmatrix} M_{1z} \\ M_{2z} \\ M_{1y} \\ M_{2y} \\ Q \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} \varphi_{1z} \\ \varphi_{2z} \\ \varphi_{1y} \\ \varphi_{2y} \\ q \end{bmatrix} \quad (3.1)$$

Similarly, the local section force vector \mathbf{D} and local section deformation vector \mathbf{d} are defined as:

$$\mathbf{D} = \begin{bmatrix} M_z \\ M_y \\ N \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \kappa_z \\ \kappa_y \\ d \end{bmatrix} \quad (3.2)$$

where the section parameters can be approximated from the element parameters using the matrix of deformation interpolation \mathbf{a} and the matrix of force interpolation functions \mathbf{b} , as the mixed approach is being used. The approximation is as follows:

$$\mathbf{d} = \mathbf{a} \cdot \mathbf{q}, \quad (3.3)$$

$$\mathbf{D} = \mathbf{b} \cdot \mathbf{Q}. \quad (3.4)$$

The determination of the strains from the section deformations is a statically indeterminate problem, since the section has typically more than two fibers. However, given the assumptions mentioned earlier in section 3.2 of small deformations as well as the assumption of the section remaining plane and normal to the local x axis during the deformation, it can be safely assumed that the relation between the fiber strains and the section deformations is linear, and defined as follows:

$$\varepsilon_f = \mathbf{y}_f \cdot \mathbf{d}, \quad (3.5)$$

where \mathbf{y} is the direction vector of each fiber such that $\mathbf{y} = [-y \ z \ 1]$. The subscript f refers to the fiber. This definition of the direction vector can be extended to include bond-slippage or non-linearity, however that is beyond the scope of this study.

3.4. Formulation of the Element

In this section, the formulation of the element is shown, using the principal of virtual force. Also, the reduction from the mixed approach to the flexibility approach is shown. The fully detailed derivation can be found in [2]. First, a mixed approach is followed such that the equations

$$\Delta \mathbf{d}_i(x) = \mathbf{a}(x) \cdot \Delta \mathbf{q}_i, \quad (3.6)$$

$$\Delta \mathbf{D}_i(x) = \mathbf{b}(x) \cdot \Delta \mathbf{Q}_i \quad (3.7)$$

hold, where Δ denotes increments, and i the non-linear iteration. As described in Zienkiewicz and Taylor's Finite Elements book [13], with the application of principal of virtual force, the weak form of the virtual work can be written as:

$$\int_0^L \delta \mathbf{D}^T(x) \cdot [\Delta \mathbf{d}_i(x) - \mathbf{f}_{i-1}(x) \cdot \Delta \mathbf{D}_i(x)] dx = 0, \quad (3.8)$$

where the domain of integration, Ω , is the element that goes from 0 to the total length L , and the integration is carried upon the aforementioned control sections, where $\mathbf{f}(x)$ is the symmetric flexibility matrix of the section. Inserting equations (3.6) and (3.7) into (3.8) yields:

$$\int_0^L \delta \mathbf{Q}^T \cdot \mathbf{b}^T(x) \cdot [\mathbf{a}(x) \cdot \Delta \mathbf{q}_i - \mathbf{f}_{i-1}(x) \cdot \mathbf{b}(x) \cdot \Delta \mathbf{Q}_i] dx = 0. \quad (3.9)$$

Reorganizing the equation and putting the element variables outside the integral results

$$\delta \mathbf{Q}^T \left(\underbrace{\left[\int_0^L \mathbf{b}^T(x) \cdot \mathbf{a}(x) dx \right]}_{\mathbf{T}} \cdot \Delta \mathbf{q}_i - \underbrace{\left[\int_0^L \mathbf{b}^T(x) \cdot \mathbf{f}(x) \cdot \mathbf{b}(x) dx \right]}_{\mathbf{F}_{i-1}} \cdot \Delta \mathbf{Q}_i \right) = 0, \quad (3.10)$$

where \mathbf{F}_{i-1} is the flexibility matrix of the element in the previous iteration and \mathbf{T} is the matrix that relates the two matrices of the interpolation functions. This equation has to be satisfied with an arbitrary choice of virtual element force vector $\delta \mathbf{Q}$, so it can be omitted from the equation, yielding

$$\mathbf{T} \cdot \Delta \mathbf{q}_i = \mathbf{F}_{i-1} \cdot \Delta \mathbf{Q}_i. \quad (3.11)$$

Similarly, using the principal of virtual displacement, the weak form of the virtual work can be written as:

$$\int_0^L \delta \mathbf{d}^T(x) \cdot [\mathbf{D}_{i-1}(x) + \Delta \mathbf{D}_i(x)] dx - \delta \mathbf{q}^T \cdot \mathbf{Q}_i^E = 0, \quad (3.12)$$

where \mathbf{Q}^E is the external force vector. Inserting equations (3.6) and (3.7) into (3.12) and reorganizing yields

$$\delta \mathbf{q}^T \left(\underbrace{\left[\int_0^L \mathbf{a}^T(x) \cdot \mathbf{b}(x) dx \right]}_{\mathbf{T}^T} \cdot \mathbf{Q}_{i-1} - \underbrace{\left[\int_0^L \mathbf{a}^T(x) \cdot \mathbf{b}(x) dx \right]}_{\mathbf{T}^T} \cdot \Delta \mathbf{Q}_i - \mathbf{Q}_i^E \right) = 0. \quad (3.13)$$

This equation has to be satisfied with an arbitrary choice of virtual element deformation vector $\delta \mathbf{q}$, so it can be omitted from the equation, yielding

$$\mathbf{T}^T \cdot \Delta \mathbf{Q}_i = \mathbf{Q}_i^E - \mathbf{T}^T \cdot \mathbf{Q}_{i-1}. \quad (3.14)$$

Inserting the solution of eq. (3.11) for $\Delta \mathbf{Q}_i$ into eq. (3.14) results

$$\mathbf{T}^T \cdot [\mathbf{F}_{i-1}]^{-1} \cdot \mathbf{T} \cdot \Delta \mathbf{q}_i = \mathbf{Q}_i^E - \mathbf{T}^T \cdot \mathbf{Q}_{i-1}. \quad (3.15)$$

Equation (3.15) is the matrix form of the weak formulation using the mixed approach of non-linear FEM. Nonetheless, for this element, the deformation interpolation functions are selected in a way to reduce mixed approach to the flexibility approach. Choosing $\mathbf{a}(x) = \mathbf{f}_{i-1}(x) \cdot \mathbf{b}(x) \cdot [\mathbf{F}_{i-1}]^{-1}$ would result in the \mathbf{T} to become the identity matrix, hence, reducing eq. (3.15) to

$$[\mathbf{F}_{i-1}]^{-1} \cdot \Delta \mathbf{q}_i = \mathbf{Q}_i^E - \mathbf{Q}_{i-1}, \quad (3.16)$$

where $[\mathbf{F}_{i-1}]^{-1}$ is the stiffness matrix of the previous iteration, and $\mathbf{Q}_i^E - \mathbf{Q}_{i-1}$ is the unbalance force vector, \mathbf{Q}_U^{i-1} . Although (3.16) looks similar the stiffness approach, the stiffness matrix of the element is obtained by inverting the flexibility matrix, which is obtained by the flexibility matrices of the sections. Now, the choice of the force interpolation functions matrix \mathbf{b} is free. In this work, $\mathbf{b}(x)$ is chosen to obtain linear bending moments and constant axial forces across the sections, such that

$$\mathbf{b} = \begin{bmatrix} \frac{x}{L} - 1 & \frac{x}{L} & 0 & 0 & 0 \\ 0 & 0 & \frac{x}{L} - 1 & \frac{x}{L} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.17)$$

This leads to exact forces in case of no elements forces.

Finally, combining eq. (3.16) with eq. (2.11) and generalizing it into the structure level with multiple elements, the following holds

$$\begin{bmatrix} \mathbf{K}_s^{i-1} & -\mathbf{P}_E \\ \left. \frac{\partial C}{\partial \mathbf{p}} \right|_i & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta \mathbf{p}^i \\ \Delta \lambda^i \end{bmatrix} = \begin{bmatrix} \mathbf{P}_U^{i-1} \\ C \end{bmatrix}, \quad (3.18)$$

where C is the displacement control constraint, and \mathbf{K}_s is the stiffness matrix of the global structure.

3.5. Uniaxial Fibers

So far, the element matrices are obtained by the integration over section matrices. In this section, the section matrices are computed using the fiber variables.

The section flexibility matrix \mathbf{f} is derived by inverting the section stiffness matrix, which is in turn derived from and tangent moduli of the section uniaxial fibers. This can be written as:

$$\mathbf{k} = \begin{bmatrix} \sum E_t \cdot A \cdot y^2 & -\sum E_t \cdot A \cdot y \cdot z & -\sum E_t \cdot A \cdot y \\ -\sum E_t \cdot A \cdot y \cdot z & \sum E_t \cdot A \cdot z^2 & \sum E_t \cdot A \cdot z \\ -\sum E_t \cdot A \cdot y & \sum E_t \cdot A \cdot z & \sum E_t \cdot A \end{bmatrix}, \quad (3.19)$$

where the summation is over the fibers. This can be written in short using cross product

$$\mathbf{k} = \sum E_t \cdot A \cdot \mathbf{y}^T \cdot \mathbf{y}, \quad (3.20)$$

where the direction vector of each fiber is $\mathbf{y} = [-y \ z \ 1]$. The tangent modulus of each fiber, E_t , is calculated based on its constitutive law, using the strain value calculated as eq. (3.5).

In a similar way, the section resisting forces \mathbf{D}_R are calculated as:

$$\mathbf{D}_R = \mathbf{y}_f^T \cdot \sigma_f \cdot A, \quad (3.21)$$

where the section resisting forces can be transformed into the element local resisting forces \mathbf{Q}_R using eq. (3.4).

3.6. Element Solution Algorithm

The algorithm for each load step involves two nested loops, a Newton-Raphson iteration loop and an element equilibrium loop. In details, for every load step, Newton-Raphson iterations are performed on the structure level until convergence, and for every Newton-Raphson iteration, an element loop is performed on all elements until convergence.

Hereon, three superscripts are used to refer to each aforementioned loop. k is for the load step, i is for the Newton-Raphson iteration, and j is for the element loop. The load stepping loop for three load steps can be depicted in Figure 9 and Figure 10.

Algorithm 1 shows the outer loops of the load stepping and the NR iteration loop. First, the structure is initialized, where the global stiffness matrix is computed. In each computation of the stiffness matrix, the section stiffness matrices are first obtained using eq. (3.19), where the tangent modulus of each fiber is initialized, then inverted to get the section flexibility matrices, then globalized using the matrix of force interpolation functions matrix \mathbf{b} such that

$$\mathbf{f}_{\text{global}} = \mathbf{b}^T \cdot \mathbf{f} \cdot \mathbf{b}. \quad (3.22)$$

Finally, the element stiffness matrix is the inverse of the element flexibility matrix, which is obtained from integrating over the sections, where the numerical integral is

$$\mathbf{F} = \sum_{\text{sections}} \mathbf{f}_{\text{global}} \cdot w_{\text{sec}} \cdot \det \mathbf{J}, \quad (3.23)$$

where w_{sec} is the numerical weight of the section based on the integration rule and the determinant of the jacobian $\det J$ in this case is simply $0.5 \cdot L$, where L is the reference length.

After the initialization, the load stepping loop starts, where in each NR iteration, the change of increments of displacements $\delta \Delta \mathbf{p}^{k,i}$ is computed using eq. (3.18). Then, the elements change of increments of deformations $\delta \Delta \mathbf{q}^{k,i}$ is computed. Next, the element loop shown in Algorithm 2 is performed till equilibrium of elements. Finally, the global stiffness matrix, resisting forces, and unbalance forces are updated. The norm of the unbalance forces is checked against a tolerance as a convergence check for the NR iteration.

Algorithm 1: Load Stepping Algorithm

```

1 initialize
2  $k \leftarrow 1$ 
   while  $k \leq \text{final load step}$  do
3      $i \leftarrow 1$ ; converged  $\leftarrow$  false;
       while converged == false &  $i \leq \text{max NR iterations}$  do
4         solve eq. (3.18) for  $\delta\Delta\mathbf{p}^{k^i}$  and  $\delta\Delta\lambda^{k^i}$ 
            $\Delta\mathbf{p}^{k^i} \leftarrow \Delta\mathbf{p}^{k^{i-1}} + \delta\Delta\mathbf{p}^{k^i}$  //  $\Delta\mathbf{p}^{k^0} = \mathbf{0}$ 
            $\Delta\lambda^{k^i} \leftarrow \Delta\lambda^{k^{i-1}} + \delta\Delta\lambda^{k^i}$  //  $\Delta\lambda^{k^0} = 0$ 
            $\lambda^{k^i} \leftarrow \lambda^{k^{i-1}} + \Delta\lambda^{k^i}$  //  $\lambda^{k^0} = 0$ 
5         for each element do
            $\delta\Delta\mathbf{q}^{k^i} \leftarrow \mathbf{L} \cdot \Delta\mathbf{p}^{k^i}$ 
6-14        Perform element equilibrium loop (Algorithm 2)
15         $\mathbf{K}_s^{k^i} \leftarrow \sum \mathbf{L}^T \cdot \mathbf{K}^{k^i j} \cdot \mathbf{L}$ 
            $\mathbf{P}_R^{k^i} \leftarrow \sum \mathbf{L} \cdot \mathbf{Q}^{k^i j}$ 
            $\mathbf{P}_U^{k^i} \leftarrow \lambda^{k^i} \mathbf{P}_E^k - \mathbf{P}_R^{k^i}$ 
16        if  $\|\mathbf{P}_U^{k^i}\| < \text{tol}$  then
           converged  $\leftarrow$  true
           finalize load step
        else
            $k \leftarrow k + 1$ 

```

Algorithm 2 shows the element loop. It starts after calculating the change of increments of element deformations, $\delta\Delta\mathbf{q}^{k^i}$, as it is used to compute the change of increments of element forces, $\delta\Delta\mathbf{Q}^{k^i j}$, in the first iteration of the element equilibrium loop. Otherwise the element deformation residuals vector of the previous iteration is used, $\mathbf{s}^{k^i j-1}$, where it is computed at the end of the iteration using the integral over the section deformation residuals.

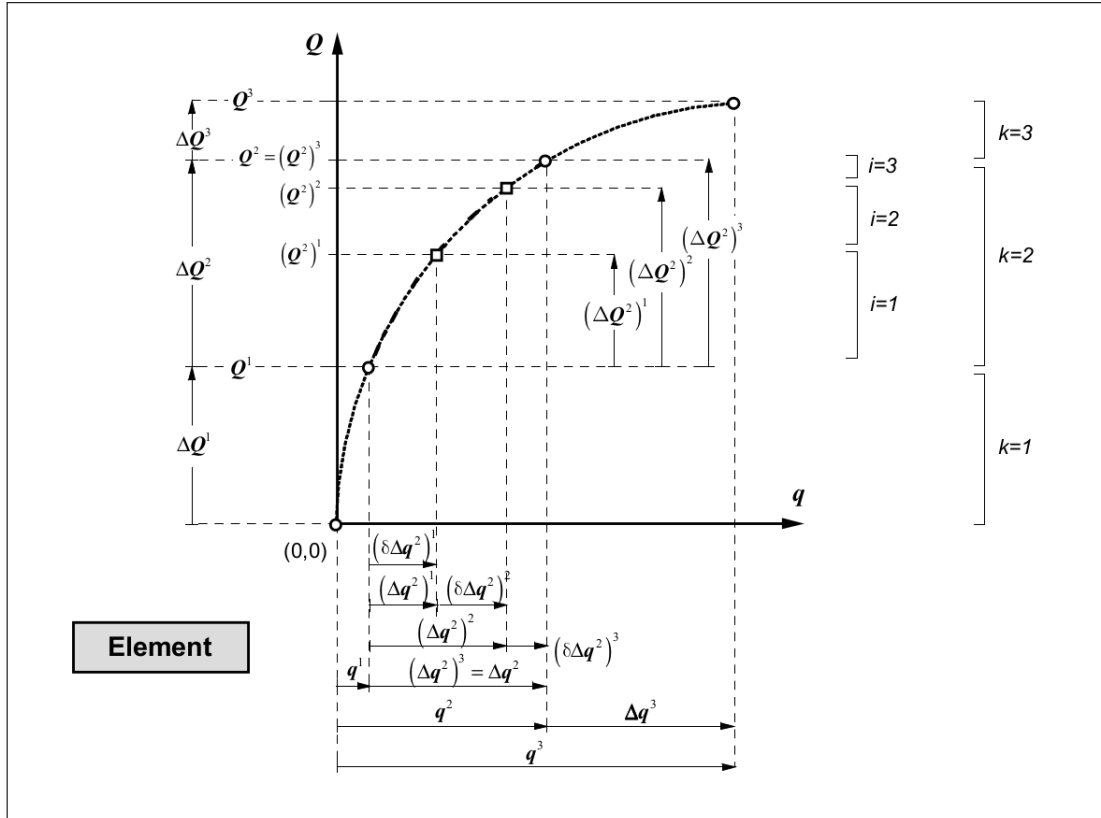
Next, for all the sections in the element, the change of increments of section forces can be computed using eq. (3.4). Then, to compute the change of increments of section deformations, the residual of deformations vector of last iteration is added to the effect of the change of increments of section forces of the current iteration, as in

$$\delta\Delta\mathbf{d}^{k^i j} = \mathbf{r}^{k^i j-1} + \mathbf{f}^{k^i j-1} \cdot \delta\Delta\mathbf{D}^{k^i j}. \quad (3.24)$$

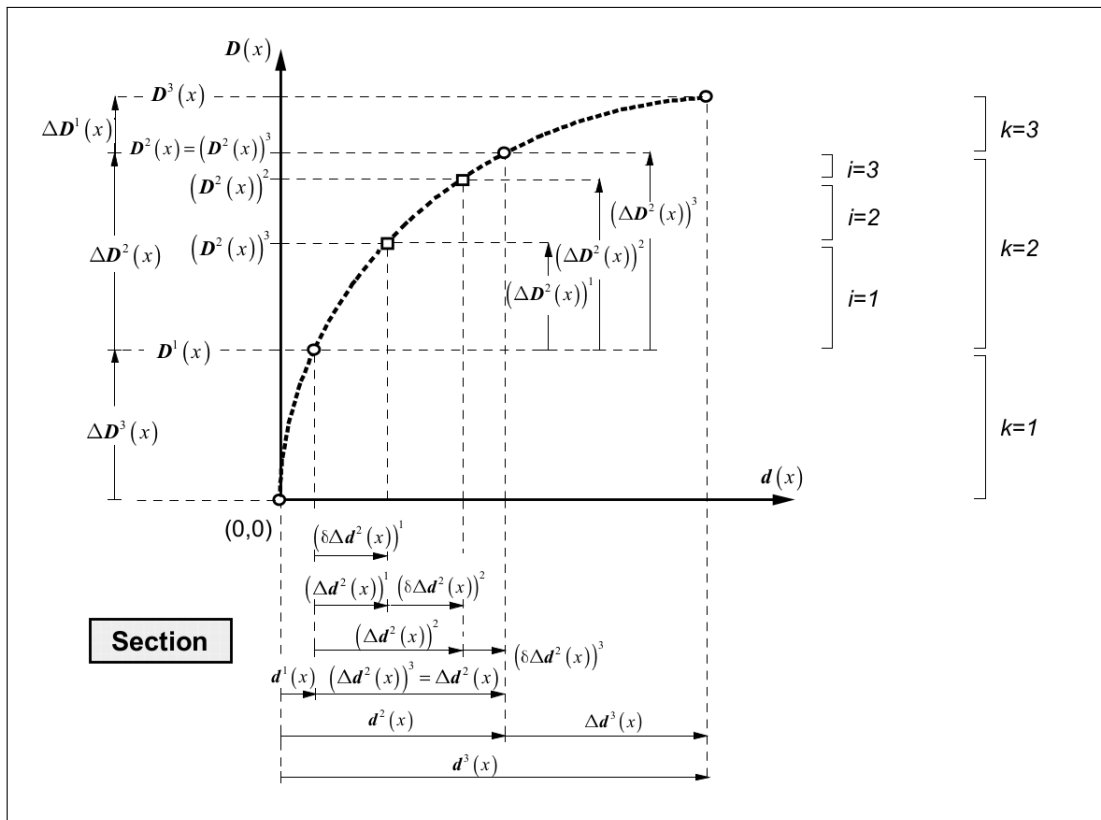
Then, eq. (3.5) is used to compute the strain of each fiber, where the uniaxial constitutive law of the fiber is used to determine the stress as well as the tangent modulus. These parameters are used in eq. (3.20) to compute the section stiffness matrix, and in eq. (3.21) to compute the section resisting forces. Now, the section unbalance forces and deformation residuals can be computed, and the element stiffness matrix and deformation residuals can be computed. Finally, if all sections have reached equilibrium, meaningly the norm of the unbalance force is below a certain tolerance, the element equilibrium loop is concluded.

Algorithm 2: Element Loop Algorithm

```
6  $j \leftarrow 1$ ; elements_converged  $\leftarrow$  false;
   while elements_converged == false &  $j \leq$  max element iterations do
7     for each element do
       if  $j == 1$  then
          $\delta \Delta \mathbf{Q}^{kij} \leftarrow \mathbf{K}^{k^{i-1}} \cdot \delta \Delta \mathbf{q}^{k^i}$ 
       else
          $\delta \Delta \mathbf{Q}^{kij} \leftarrow \mathbf{K}^{k^{ij-1}} \cdot \mathbf{s}^{k^{ij-1}}$ 
          $\Delta \mathbf{Q}^{kij} \leftarrow \Delta \mathbf{Q}^{k^{ij-1}} + \delta \Delta \mathbf{Q}^{kij}$  //  $\Delta \mathbf{Q}^{k^{i0}} = 0$ 
          $\mathbf{Q}^{kij} \leftarrow \mathbf{Q}^{k-1} + \Delta \mathbf{Q}^{kij}$  //  $\mathbf{Q}^0 = 0$ 
       for each section do
8          $\delta \Delta \mathbf{D}^{kij} \leftarrow \mathbf{b} \cdot \delta \Delta \mathbf{Q}^{kij}$ 
          $\Delta \mathbf{D}^{kij} \leftarrow \Delta \mathbf{D}^{k^{ij-1}} + \delta \Delta \mathbf{D}^{kij}$  //  $\Delta \mathbf{D}^{k^{i0}} = 0$ 
          $\mathbf{D}^{kij} \leftarrow \mathbf{D}^{k-1} + \Delta \mathbf{D}^{kij}$  //  $\mathbf{D}^0 = 0$ 
9          $\delta \Delta \mathbf{d}^{kij} \leftarrow \mathbf{r}^{k^{ij-1}} + \mathbf{f}^{k^{ij-1}} \cdot \delta \Delta \mathbf{D}^{kij}$  //  $\mathbf{r}^{k^{i0}} = 0$ 
10        for each fiber do
           $\delta \Delta \varepsilon^{kij} \leftarrow \mathbf{y} \cdot \delta \Delta \mathbf{d}^{kij}$ 
           $\Delta \varepsilon^{kij} \leftarrow \Delta \varepsilon^{k^{ij-1}} + \delta \Delta \varepsilon^{kij}$  //  $\Delta \varepsilon^{k^{i0}} = 0$ 
           $\varepsilon^{kij} \leftarrow \varepsilon^{k-1} + \Delta \varepsilon^{kij}$  //  $\varepsilon^0 = 0$ 
          apply constitutive law and get stress  $\sigma$  and tangent modulus  $E_t$ 
11          $\mathbf{k}^{kij} \leftarrow \sum E_f \cdot A_f \cdot \mathbf{y}_f^T \cdot \mathbf{y}_f$ 
           $\mathbf{f}^{kij} \leftarrow [\mathbf{k}^{kij}]^{-1}$ 
12          $\mathbf{D}_R^{kij} \leftarrow \sum \sigma_f \cdot A_f \cdot \mathbf{y}_f^T$ 
           $\mathbf{D}_U^{kij} \leftarrow \mathbf{D}^{kij} - \mathbf{D}_R^{kij}$ 
           $\mathbf{r}^{kij} \leftarrow \mathbf{f}^{kij} \cdot \mathbf{D}_U^{kij}$ 
13          $\mathbf{F}^{kij} \leftarrow \sum \mathbf{b}^T \cdot \mathbf{f}^{kij} \cdot \mathbf{b} \cdot w_{\text{sec}} \cdot \det J$ 
           $\mathbf{K}^{kij} \leftarrow [\mathbf{F}^{kij}]^{-1}$ 
14         if  $\|\mathbf{D}_U^{kij}\| < \text{element\_tolerance} \forall$  sections then
          | elements_converged  $\leftarrow$  true
          else
          |  $j \leftarrow j + 1$ 
          |  $\mathbf{s}^{kij} \leftarrow \sum \mathbf{b}^T \cdot \mathbf{r}^{kij} \cdot \det J$ 
```



(a) Element level.



(b) Section level.

Figure 9 Illustration of the three nested loops for three load steps on the element and the section level. [2]

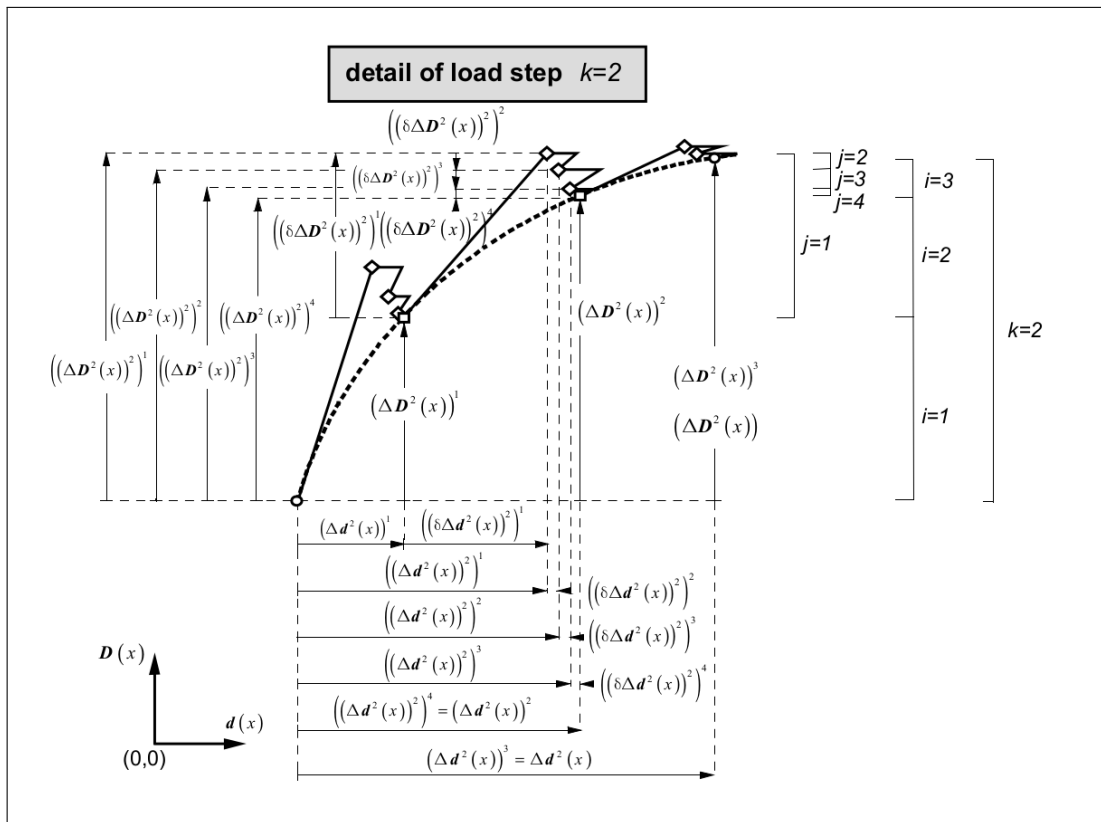
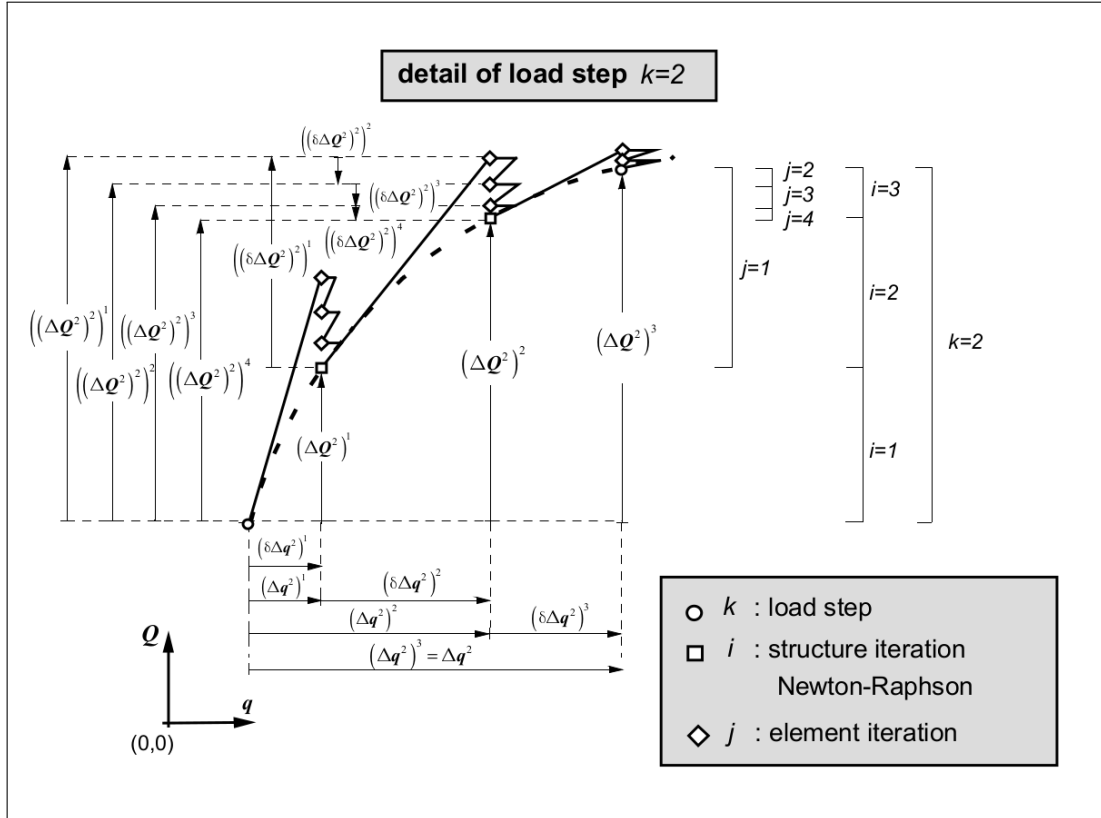


Figure 10 Illustration of the change of parameters in one load step on the element and the section level. [2]

3.7. Detailed Explanation of the Algorithm

The total algorithm to advance in load steps is shown in Algorithm 1 and Algorithm 2. Algorithm 1 shows the loop of load steps as well as the loop of the NR iterations, where the displacements, the stiffness matrix the resisting and unbalance forces are calculated, and Algorithm 2 shows the loop of element equilibrium, where the element forces are calculated based on the element displacements. Based on element forces, the section forces are calculated, followed by section displacements, then, fiber strains are calculated. Given the fiber strains, the uniaxial constitutive laws can be applied to get the material response of every fiber, so that the section flexibility matrix, section resisting forces, section unbalance forces, and section displacement residuals can be updated. Finally the element local stiffness matrix can be computed, and the convergence can be checked with the section unbalance forces. If all sections converged, the structure stiffness matrix, and resisting and unbalance forces can be computed, where the structure unbalance forces are checked for the convergence of the NR iteration. It is noteworthy to mention that the element and the section flexibility matrices are used more than once in every iteration, so they are saved in memory instead of recomputation. Also, the increments of deformations and forces are changed in every iteration until they converge, so they are saved as well, but the changes in increments are calculated and not saved. Here follows the summary and explanation of each step:

1. The goal of the initialization is to compute the initial structure stiffness matrix. First, all fibers are initialized so that the initial tangent modulus E_t is assumed based on the constitutive law. Then, for each section, the stiffness matrix are calculated using the section fibers. Each section stiffness matrix, \mathbf{k} , is computed using eq. (3.20), which defines an integral over the fiber. After \mathbf{k} is computed, the section flexibility matrix, \mathbf{f} , which is the inverse of \mathbf{k} , is computed and stored. Next, the element flexibility matrix, \mathbf{F} , is computed using an integral over section flexibility matrices, which is defined in eq. (3.10). Numerically, the Gauss Lobatto rule is applied, where every section is given a position and a weight, then,

$$\mathbf{F} = \int_0^L \mathbf{b}^T(x) \cdot \mathbf{f}(x) \cdot \mathbf{b}(x) dx \approx \sum_{sections} \mathbf{b}^T \cdot \mathbf{f} \cdot \mathbf{b} \cdot w_{sec} \cdot \frac{L}{2}, \quad (3.25)$$

where \mathbf{f} is the local section flexibility matrix and \mathbf{F} is the element flexibility matrix. The element local stiffness matrix, \mathbf{K} which is simply the inverse of \mathbf{F} , is computed and stored. Finally, the structure tangent stiffness matrix, \mathbf{K}_s , can be computed by assembling the global element stiffness matrix. The global element stiffness matrix, \mathbf{K}_{global} is calculated as

$$\mathbf{K}_{global} = \mathbf{L} \cdot \mathbf{K} \cdot \mathbf{L}^T, \quad (3.26)$$

where \mathbf{L} is a transformation of coordinates matrix. Since the element includes only axial loading and biaxial bending, the 12×5 matrix \mathbf{L} is defined as

$$\mathbf{L} = \begin{bmatrix} \frac{\mathbf{e}_2}{L} & \frac{\mathbf{e}_2}{L} & -\frac{\mathbf{e}_3}{L} & -\frac{\mathbf{e}_3}{L} & -\mathbf{e}_1 \\ \mathbf{e}_3 & \mathbf{0} & \mathbf{e}_2 & \mathbf{0} & \mathbf{0} \\ -\frac{\mathbf{e}_2}{L} & -\frac{\mathbf{e}_2}{L} & -\frac{\mathbf{e}_3}{L} & -\frac{\mathbf{e}_3}{L} & -\mathbf{e}_1 \\ \mathbf{0} & \mathbf{e}_3 & \mathbf{0} & \mathbf{e}_2 & \mathbf{0} \end{bmatrix}, \quad (3.27)$$

where \mathbf{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3 are the unit vectors that describe the element's local coordinates system expressed as 3×1 column vectors, and $\mathbf{0}$ is a 3×1 column vector of zeros. The first three rows of \mathbf{L} are to transform the forces of the first node, the second three rows are for the moments of the first nodes, the third three rows are for the forces of the second node, and the final three rows are for the moment of the second node. L is the undeformed length of the element.

2. The index of the load steps, k , is set to one, and the load stepping loop starts.
3. The index of the NR iterations, i is set to one, and the NR iteration loop starts.
4. The linear system of equations derived in eq. (3.18), which can be written in this loop as

$$\begin{bmatrix} \mathbf{K}_s^{k^{i-1}} & -\mathbf{P}_E^k \\ \left. \frac{\partial C}{\partial \mathbf{p}} \right|_i^k & 0 \end{bmatrix} \cdot \begin{bmatrix} \delta \Delta \mathbf{p}^{k^i} \\ \delta \Delta \lambda^{k^i} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_U^{k^{i-1}} \\ C^k \end{bmatrix}, \quad (3.28)$$

where \mathbf{K}_s is the structure stiffness matrix after applying the Dirichlet boundary conditions and $\mathbf{K}_s^{k^0}$ is the initialized matrix in step 1, \mathbf{P}_E is the external load vector, $\mathbf{P}_U^{k^{i-1}}$ is the unbalance force vector and $\mathbf{P}_U^{k^0}$ is a zero vector, C is the displacement control constraint defined in eq. (2.8), and $\left. \frac{\partial C}{\partial \mathbf{p}} \right|_i^k$ is simply a zero vector with a value of one at the controlled DoF. The linear system of equations is solved for the change of increments of displacements, $\delta \Delta \mathbf{p}^{k^i}$, and the change of increments of the load factor, $\delta \Delta \lambda^{k^i}$. Then, the increments are calculated as in

$$\Delta \mathbf{p}^{k^i} = \Delta \mathbf{p}^{k^{i-1}} + \delta \Delta \mathbf{p}^{k^i}, \quad (3.29)$$

$$\Delta \lambda^{k^i} = \Delta \lambda^{k^{i-1}} + \delta \Delta \lambda^{k^i}, \quad (3.30)$$

where $\Delta \mathbf{p}^{k^0} = \mathbf{0}$ and $\Delta \lambda^{k^0} = 0$, and the load factor is incremented, $\lambda^{k^i} = \lambda^k + \Delta \lambda^{k^i}$, where $\lambda^{k^0} = 0$.

5. The change of increments of the element local deformations is then calculated from the structure using the transformation matrix, \mathbf{L} ,

$$\delta \Delta \mathbf{q}^{k^i} = \mathbf{L} \cdot \Delta \mathbf{p}^{k^i}. \quad (3.31)$$

$\delta\Delta\mathbf{q}^{k^i}$ is updated only once after every NR iteration since it will be used only once in the first element equilibrium iteration. Also, $\Delta\mathbf{q}^{k^i}$ and \mathbf{q}^{k^i} are not needed.

6. The index of the element equilibrium iterations, j is set to one, and the element equilibrium loop starts.
7. For every element, the change of increments of element forces is calculated as

$$\delta\Delta\mathbf{Q}^{k^{i1}} = \mathbf{K}^{k^{i-1}} \cdot \delta\Delta\mathbf{q}^{k^i} \quad (3.32)$$

if $j = 1$. Otherwise, if $j > 1$, then

$$\delta\Delta\mathbf{Q}^{k^{ij}} = \mathbf{K}^{k^{ij-1}} \cdot \mathbf{s}^{k^{ij-1}}, \quad (3.33)$$

where \mathbf{s} is the displacement residual, since the change of increments of element deformations is not updated in the element equilibrium iterations. Then, the section increments of forces, $\Delta\mathbf{Q}^{k^{ij}}$, and the section forces, $\mathbf{Q}^{k^{ij}}$, are calculated as

$$\Delta\mathbf{Q}^{k^{ij}} = \Delta\mathbf{Q}^{k^{ij-1}} + \delta\Delta\mathbf{Q}^{k^{ij}}, \quad (3.34)$$

$$\mathbf{Q}^{k^{ij}} = \mathbf{Q}^{k-1} + \Delta\mathbf{Q}^{k^{ij}}, \quad (3.35)$$

where $\Delta\mathbf{Q}^{k^{i0}} = \mathbf{0}$ and $\mathbf{Q}^0 = \mathbf{0}$.

8. For every section in the element, the section forces, $\mathbf{D}^{k^{ij}}$, are calculated from the element change in increments of forces, $\delta\Delta\mathbf{Q}^{k^{ij}}$, using the force interpolation functions matrix, \mathbf{b} , using

$$\delta\Delta\mathbf{D}^{k^{ij}} = \mathbf{b} \cdot \delta\Delta\mathbf{Q}^{k^{ij}}, \quad (3.36)$$

$$\Delta\mathbf{D}^{k^{ij}} = \Delta\mathbf{D}^{k^{ij-1}} + \delta\Delta\mathbf{D}^{k^{ij}}, \quad (3.37)$$

$$\text{and } \mathbf{D}^{k^{ij}} = \mathbf{D}^{k-1} + \Delta\mathbf{D}^{k^{ij}}, \quad (3.38)$$

where $\Delta\mathbf{D}^{k^{i0}} = \mathbf{0}$ and $\mathbf{D}^0 = \mathbf{0}$.

9. Using the stored section flexibility matrix of the last iteration, $\mathbf{f}^{k^{ij-1}}$, and the residual of section displacements of the last iteration, $\mathbf{r}^{k^{ij-1}}$, the change of increments of section deformations can be calculated as

$$\delta\Delta\mathbf{d}^{k^{ij}} = \mathbf{r}^{k^{ij-1}} + \mathbf{f}^{k^{ij-1}} \cdot \delta\Delta\mathbf{D}^{k^{ij}}, \quad (3.39)$$

where $\mathbf{r}^{k^{i0}} = \mathbf{0}$, $\Delta\mathbf{d}^{k^{i0}} = \mathbf{0}$, and $\mathbf{d}^0 = \mathbf{0}$.

10. For every fiber in the section, the strain, $\varepsilon^{k^{ij}}$, can be directly calculated as in eq. (3.5).

11.

$$\delta\Delta\varepsilon^{kij} = \mathbf{y}_f \cdot \delta\Delta\mathbf{d}^{kij}, \quad (3.40)$$

$$\Delta\varepsilon^{kij} = \Delta\varepsilon^{kij-1} + \delta\Delta\varepsilon^{kij}, \quad (3.41)$$

$$\text{and } \varepsilon^{kij} = \varepsilon^{k-1} + \Delta\varepsilon^{kij}, \quad (3.42)$$

where $\Delta\varepsilon^{k^i0} = 0$ and $\varepsilon^0 = 0$. Then, the uniaxial material law is applied on the fiber to obtain the tangent modulus, E_t and the stress σ .

12. The section stiffness matrix, \mathbf{k}^{kij} is then updated using eq. (3.20), and the section flexibility matrix \mathbf{f}^{kij} is computed and restored.
13. The resisting forces can be calculated from the fiber stresses as in

$$\mathbf{D}_R^{kij} = \sum \sigma_f \cdot A_f \cdot \mathbf{y}_f^T \quad (3.43)$$

from eq. (3.21). Then, the unbalance forces is simply the difference between the section forces calculated in step 7 and the resisting forces,

$$\mathbf{D}_U^{kij} = \mathbf{D}^{kij} - \mathbf{D}_R^{kij}. \quad (3.44)$$

Finally, the section displacement residuals is calculated based on the updated flexibility matrix and the unbalance forces as in

$$\mathbf{r}^{kij} = \mathbf{f}^{kij} \cdot \mathbf{D}_U^{kij} \quad (3.45)$$

14. Now, the element matrices can be updated. The element flexibility, \mathbf{F}^{kij} , is updated using eq. (3.25) then inverted to obtain the element local stiffness matrix, which is stored.
15. Element equilibrium is checked by calculating the L2 norm of the section unbalance forces $\|\mathbf{D}_U^{kij}\|$. If $\|\mathbf{D}_U^{kij}\| < \text{tolerance}$ for all of the sections of the element, then equilibrium is achieved. Otherwise, iterations are needed, so the element deformation residuals, \mathbf{s}^{kij} , is computed as a numerical integral over the section deformation residuals

$$\mathbf{s}^{kij} = \int_0^L \mathbf{b}^T(x) \cdot \mathbf{r}^{kij}(x) dx \approx \sum_{\text{sections}} \mathbf{b}^T \cdot \mathbf{r}^{kij} \cdot w_{\text{sec}} \cdot \det \mathbf{J}, \quad (3.46)$$

j is incremented by one, and steps 7 through 15 are repeated.

16. Now, the structure stiffness matrix, \mathbf{K}_s can be updated, using an assembly of the element global stiffness matrices as in eq. (3.26)

$$\mathbf{K}_s^{k^i} = \sum_{elements} \mathbf{L} \cdot \mathbf{K} \cdot \mathbf{L}^T. \quad (3.47)$$

The structure resisting forces vector, \mathbf{P}_R , is an assembly of the element resisting forces such that

$$\mathbf{P}_R^{k^i} = \sum_{elements} \mathbf{L} \cdot \mathbf{Q}^{k^i j}. \quad (3.48)$$

Finally, the structure unbalance forces, $\mathbf{P}_U^{k^i}$, is calculated as

$$\mathbf{P}_U^{k^i} = \lambda^{k^i} \mathbf{P}_E^k - \mathbf{P}_R^{k^i}. \quad (3.49)$$

17. NR convergence is checked by calculating the L2 norm of the structure unbalance forces $\|\mathbf{P}_U^{k^i}\|$. If $\|\mathbf{P}_U^{k^i}\| > \text{tolerance}$, then convergence is NOT achieved and more NR iterations are needed, so i is incremented by one and steps 4 through 17 are repeated. Otherwise, convergence is achieved, so the following is set:

$$\Delta \mathbf{p}^{k^i} = \mathbf{0}$$

$$\Delta \lambda^{k^i} = 0$$

$$\mathbf{p}^k = \mathbf{p}^{k^{i-1}} + \Delta \mathbf{p}^{k^i}$$

$$\Delta \mathbf{Q}^{k^i j} = \mathbf{0} \quad \forall \text{ elements}$$

$$\mathbf{Q}^k = \mathbf{Q}^{k^i j} \quad \forall \text{ elements}$$

$$\Delta \mathbf{D}^{k^i j} = \mathbf{0} \quad \forall \text{ sections}$$

$$\mathbf{D}^k = \mathbf{D}^{k^i j} \quad \forall \text{ sections}$$

$$\Delta \varepsilon^{k^i j} = 0 \quad \forall \text{ fibers}$$

$$\varepsilon^k = \varepsilon^{k^i j} \quad \forall \text{ fibers}$$

current material parameters are set to converged $\forall \text{ fibers}$

Finally, k is incremented by one and the new displacement control constraint is calculate. If $k < \text{maximum load steps}$, steps 3 through 17 are repeated, otherwise, the solution is concluded.

4. Constitutive Laws

As per discussed in last section, explicit uniaxial material laws can be used, as the only needed part is the uniaxial stress-strain relationship. Exactly, two material laws are implemented in this work and explained in this chapter, one for the concrete fibers and one for the steel fibers. As mentioned in section 3.3, based on the definition of the fiber direction vector, the interaction between the fibers is out of the scope of this study. Consequently, the constitutive laws can be defined independently.

4.1. Steel Uniaxial Constitutive Law

Menegotto-Pinto [14] is the material law used for the steel fibers. It models kinematic hardening, i.e. the reduction of compressive stiffness in tensile hardening and vice versa (translation of yield line in the stress space) [15], where the hardening ratio b is defined by:

$$b = \frac{E_{\infty}}{E}, \quad (4.1)$$

where E is Young's modulus and E_{∞} is the plastic modulus, shown in figure 11.

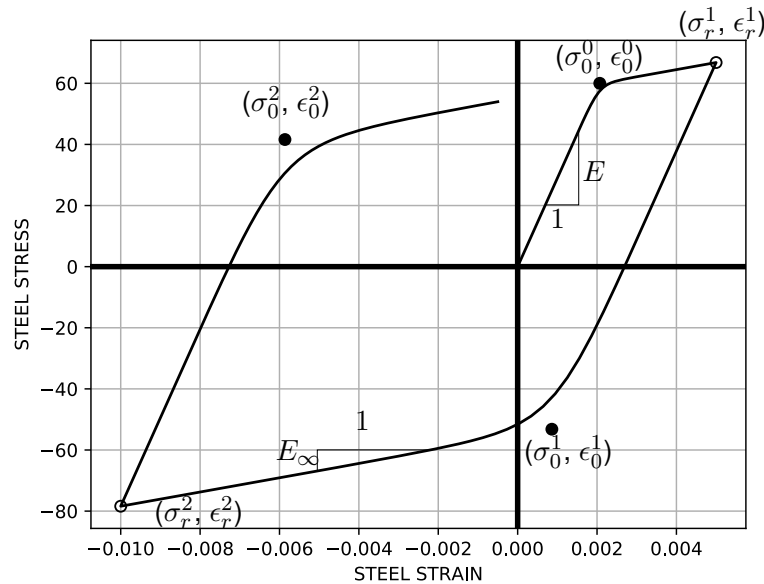


Figure 11 Stress-strain relationship of steel fibers.

The stress σ and strain ε are calculated using the normalized stress σ^* and the normalized strain ε^* , which are defined as

$$\sigma^* = \frac{\sigma - \sigma_r}{\sigma_0 - \sigma_r}, \quad (4.2)$$

$$\varepsilon^* = \frac{\varepsilon - \varepsilon_r}{\varepsilon_0 - \varepsilon_r}, \quad (4.3)$$

where σ_r and ε_r are the stress and strain at the last load reversal, respectively, and $(\sigma_0, \varepsilon_0)$ is the intersection point of the two asymptotes E and E_∞ . To calculate the stress at any point of loading/unloading, (4.2) is used, where:

$$\sigma^* = b \cdot \varepsilon^* + \frac{(1-b) \cdot \varepsilon^*}{(1 + (\varepsilon^*)^R)^{1/R}}, \quad (4.4)$$

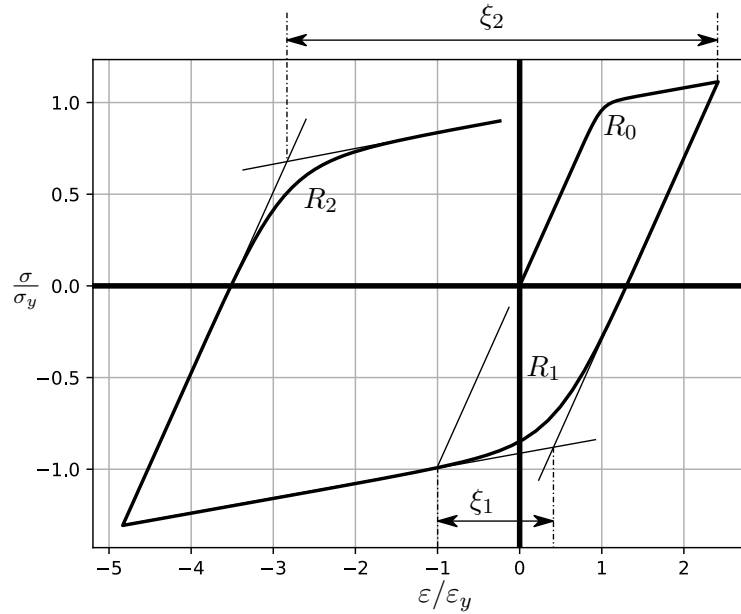


Figure 12 The Bauschinger effect in the steel fibers described by R and ξ .

where R represents the transition in between the moduli, that changes after every material reversal. It represents the Bauschinger effect. R is calculated as:

$$R(\xi) = R_0 - \frac{a_1 \cdot \xi}{a_2 + \xi}, \quad (4.5)$$

where R_0 is the initial transition variable, a_1 and a_2 are experimentally determined parameters, and ξ is the difference, normalized by the yield strain, in the strain of the current moduli intersection point and the previous max. or min. strain. The value of ξ and the effect of R on the curvature of the transition zone are shown in Figure 12, where $R_2 < R_1 < R_0$.

Finally, the stress σ can be simply calculated based on (4.2) as:

$$\sigma = \sigma^* \cdot (\sigma_0 - \sigma_r) + \sigma_r \quad (4.6)$$

The tangent modulus E_t then can be calculated as:

$$E_t = \left(\frac{\sigma_0 - \sigma_r}{\varepsilon_0 - \varepsilon_r} \right) \cdot \left\{ b + (1 - b) \cdot \left[\left(1 + (\varepsilon^*)^R \right)^{-1/R} - (\varepsilon^*)^R \cdot \left(1 + (\varepsilon^*)^R \right)^{-1 - \frac{1}{R}} \right] \right\} \quad (4.7)$$

4.2. Concrete Uniaxial Constitutive Law

Kent-Park [16] is the material law used to model the concrete fibers. It models uni-axial concrete only under compression. The stress σ is defined as:

$$\sigma = \begin{cases} 0 & \text{if } \varepsilon > 0 \\ f'_c \cdot \left[2 \cdot \left(\frac{\varepsilon}{\varepsilon_0} \right) - \left(\frac{\varepsilon}{\varepsilon_0} \right)^2 \right] & \text{if } 0 > \varepsilon \geq \varepsilon_0 \\ f'_c \cdot [1 - Z \cdot (\varepsilon - \varepsilon_0)] & \text{if } \varepsilon_0 > \varepsilon \geq \varepsilon_u \\ 0.2 \cdot f'_c & \text{if } \varepsilon_u > \varepsilon \end{cases} \quad (4.8)$$

where f'_c is the concrete compressive cylinder strength, ε_0 is the strain at the maximum stress, ε_u is the ultimate strain, and Z is the strain softening slope. Z can be determined as: and K and Z can be calculated as:

$$Z = \frac{0.5}{\frac{3}{4} \rho_s \sqrt{\frac{h'}{s_h}} + \frac{3 + 0.002 f'_c}{f'_c - 1000} - \varepsilon_0}$$

where ρ_s is the ratio of the volume of hoop reinforcement to the volume of concrete core measured to outside of stirrups, h' is the width of concrete core measured to outside of stirrups, and s_h is the center to center spacing of stirrups or hoop sets. In this work, f'_c , ε_0 , and ε_u are assumed, as they are sufficient for the model to calculate the stress-strain relationship.

The hysteretic behavior is then described by a straight line connecting the unloading strain ε_r (strain reversal) to the plastic strain ε_p following:

$$\frac{\varepsilon_p}{\varepsilon_r} = \begin{cases} 0.145 \cdot \left(\frac{\varepsilon_r}{\varepsilon_0}\right)^2 + 0.13 \cdot \frac{\varepsilon_r}{\varepsilon_0} & \text{if } \frac{\varepsilon_r}{\varepsilon_0} < 2 \\ 0.707 \cdot \left(\frac{\varepsilon_r}{\varepsilon_0} - 2\right) + 0.834 & \text{if } \frac{\varepsilon_r}{\varepsilon_0} \geq 2 \end{cases} \quad (4.9)$$

Equations (4.8) and (4.9) are depicted in Figure 13. In this model, ε_p models a compressive crushing phenomenon. Once the material is loaded beyond its yield strength, ε_p starts increasing. The unloading of the material is assumed linear until the strain is at ε_p . Any strain below ε_p resembles an open crack in the material, hence, zero stress and zero stiffness.

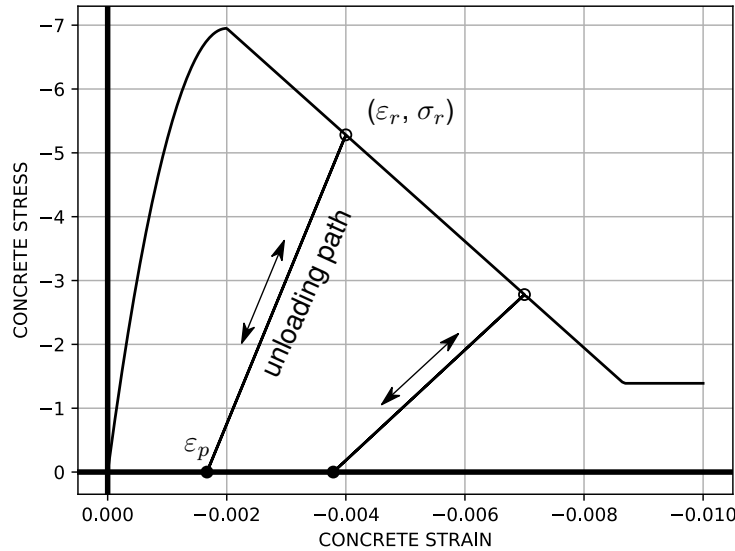


Figure 13 Stress-strain relationship of concrete fibers.

4.3. Implementation of the Constitutive Laws

As mentioned in section 3.6, in each iteration of the element equilibrium loop in Algorithm 2, the strain of each fiber is computed. After which, the tangent modulus as well as the stress can be calculated based on the previous sections in this chapter. After every change in the strain in every iteration, $\varepsilon^{k,i,j}$, the constitutive law checks for load reversal and calculates the tangent modulus and the stress accordingly. The implementation of the constitutive laws followed is based on a previous implementation in the “Open System for Earthquake Engineering Simulation” software, OpenSees. [17]

4.3.1. Algorithm for Menegotto-Pinto

For the steel material law, three extra history material variables are used, namely, ε_{max} , ε_{min} , and ε_p . Another history variable is defined as the loading index, I , which is 0 at the start, 1 at the strain increase, 2 at the strain decrease, and 3 at the strain being unchanged. To define the material, six variables are needed, namely, Young's Modulus E , the yield strength σ_y , the hardening ratio b , the initial transition parameter R_0 , and the two transition parameters a_1 and a_2 . The yield strain can then be calculated as $\varepsilon_y = \sigma_y/E$.

Algorithm 3: Steel Material Response

```

1  $\Delta\varepsilon \leftarrow \varepsilon^{kij} - \varepsilon^{k-1}$ 
2 current history variables  $\leftarrow$  converged history variables
3 if  $I^{kij} == 0$  or  $I^{kij} == 3$  then
    if  $\Delta\varepsilon \approx 0$  then
        return;
    else
         $\varepsilon_{max}^{kij} \leftarrow \varepsilon_y$ ;  $\varepsilon_{min}^{kij} \leftarrow -\varepsilon_y$ 
        if  $\Delta\varepsilon < 0$  then
             $I^{kij} \leftarrow 2$ ;  $\varepsilon_0^{kij} \leftarrow \varepsilon_{min}^{kij}$ ;  $\sigma_0^{kij} \leftarrow -\sigma_y$ ;  $\varepsilon_p^{kij} \leftarrow \varepsilon_{min}$ 
        else
             $I^{kij} \leftarrow 12$ ;  $\varepsilon_0^{kij} \leftarrow \varepsilon_{max}^{kij}$ ;  $\sigma_0^{kij} \leftarrow \sigma_y$ ;  $\varepsilon_p^{kij} \leftarrow \varepsilon_{max}$ 
4 if  $I^{kij} == 2$  and  $\Delta\varepsilon > 0$  // Load reversal
    then
         $I^{kij} \leftarrow 1$ ;  $\varepsilon_r^{kij} \leftarrow \varepsilon_r^{k-1}$ ;  $\sigma_r^{kij} \leftarrow \sigma_r^{k-1}$ 
         $\varepsilon_{min}^{kij} \leftarrow \min(\varepsilon_{min}^{kij}, \varepsilon^{k-1})$ ;  $\varepsilon_p^{kij} \leftarrow \varepsilon_{max}^{kij}$ 
         $\varepsilon_0^{kij} \leftarrow \frac{\sigma_y - E_\infty - \sigma_r^{kij} + E \cdot \varepsilon_r^{kij}}{E - E_\infty}$ 
         $\sigma_0^{kij} \leftarrow \sigma_y + E_\infty \cdot (\varepsilon_0^{kij} - \varepsilon_y)$ 
    if  $I^{kij} == 1$  and  $\Delta\varepsilon < 0$  // Load reversal
    then
         $I^{kij} \leftarrow 2$ ;  $\varepsilon_r^{kij} \leftarrow \varepsilon_r^{k-1}$ ;  $\sigma_r^{kij} \leftarrow \sigma_r^{k-1}$ 
         $\varepsilon_{max}^{kij} \leftarrow \max(\varepsilon_{max}^{kij}, \varepsilon^{k-1})$ ;  $\varepsilon_p^{kij} \leftarrow \varepsilon_{min}^{kij}$ 
         $\varepsilon_0^{kij} \leftarrow \frac{-\sigma_y + E_\infty - \sigma_r^{kij} + E \cdot \varepsilon_r^{kij}}{E - E_\infty}$ 
         $\sigma_0^{kij} \leftarrow -\sigma_y + E_\infty \cdot (\varepsilon_0^{kij} + \varepsilon_y)$ 
 $\xi \leftarrow \left| \frac{\varepsilon_{plastic} - \varepsilon_0}{\varepsilon_y} \right|$ 
Compute  $R$  as in eq. (4.5)
Compute  $\varepsilon^*$  as in eq. (4.3)
Compute  $\sigma^*$  as in eq. (4.4)
Compute  $\sigma$  from eq. (4.2)
Compute  $E_t$  from eq. (4.7)

```

To explain Algorithm 3 in the context of the nested loops mentioned discussed in section 3.7, the material response determination step comes in step 10 in the element equilibrium iteration loop in section 3.7 above. So, at every iteration, after setting the strain of the fiber, ε , the following takes place:

1. The strain difference, $\Delta\varepsilon$ is calculated as the difference between the current strain, ε^{kij} , and the converged strain from last load step, ε^{k-1} .
2. The current history variables are reset with the values of the converged variables, meaning:

$$\begin{aligned}\varepsilon_{max}^{kij} &= \varepsilon_{max}^{k-1}; & \varepsilon_{min}^{kij} &= \varepsilon_{min}^{k-1} \\ \varepsilon_p^{kij} &= \varepsilon_p^{k-1}; & \varepsilon_0^{kij} &= \varepsilon_0^{k-1} \\ \varepsilon_r^{kij} &= \varepsilon_r^{k-1}; & \varepsilon_0^{kij} &= \varepsilon_0^{k-1} \\ \varepsilon_r^{kij} &= \varepsilon_r^{k-1}; & \sigma_0^{kij} &= \sigma_0^{k-1} \\ \sigma_r^{kij} &= \sigma_r^{k-1}; & I^{kij} &= I^{k-1}\end{aligned}$$

3. In case the material is still in the initialization phase or the strain is unchanged, $I = 0$ or $I = 3$, the following happens, otherwise, the next step is followed. If the strain difference, $\Delta\varepsilon$ is nearly zero, the strain is considered not changed and the material response is **concluded**. If not, the current loading index, I^{kij} , and the intersection point of the elastic and plastic moduli, determined by ε_0^{kij} and σ_0^{kij} , and ε_p^{kij} are calculated. The values of max and min strains are used which are set as the positive and negative values of the yield strain, ε_y , respectively.

- If strain is decreasing ($\Delta\varepsilon < 0$):

$$I^{kij} = 2; \quad \varepsilon_0^{kij} = \varepsilon_{min}^{kij}; \quad \sigma_0^{kij} = -\sigma_y; \quad \varepsilon_p^{kij} = \varepsilon_{min}^{kij};$$

- If strain is increasing ($\Delta\varepsilon > 0$):

$$I^{kij} = 1; \quad \varepsilon_0^{kij} = \varepsilon_{max}^{kij}; \quad \sigma_0^{kij} = \sigma_y; \quad \varepsilon_p^{kij} = \varepsilon_{max}^{kij};$$

4. In this step, the load reversal is checked. There are two cases of load reversal, either the strain was increasing and started decreasing or vice versa. If so, the reversal strain and stress, the intersection point of the elastic and plastic moduli, and ε_p^{kij} are updated.

- If the strain was decreasing and starting increasing ($I^{kij} = 2$ and $\Delta\varepsilon < 0^{kij}$):

$$I^{kij} = 1; \quad \varepsilon_r^{kij} = \varepsilon^{k-1}; \quad \sigma_r^{kij} = \sigma^{k-1};$$

$$\varepsilon_{min}^{kij} \leftarrow \min(\varepsilon_{min}^{kij}, \varepsilon^{k-1}); \quad \varepsilon_p^{kij} = \varepsilon_{max}^{kij};$$

$$\varepsilon_0^{kij} = \frac{\sigma_y - E_\infty - \sigma_r^{kij} + E \cdot \varepsilon_r^{kij}}{E - E_\infty}; \quad \sigma_0^{kij} = \sigma_y + E_\infty \cdot (\varepsilon_0^{kij} - \varepsilon_y)$$

- If the strain was increasing and starting decreasing ($I^{kij} = 1$ and $\Delta\varepsilon > 0^{kij}$):

$$I^{kij} = 2; \quad \varepsilon_r^{kij} = \varepsilon^{k-1}; \quad \sigma_r^{kij} = \sigma^{k-1};$$

$$\varepsilon_{max}^{kij} \leftarrow \max(\varepsilon_{max}^{kij}, \varepsilon^{k-1}); \quad \varepsilon_p^{kij} = \varepsilon_{max}^{kij};$$

$$\varepsilon_0^{kij} = \frac{-\sigma_y + E_\infty - \sigma_r^{kij} + E \cdot \varepsilon_r^{kij}}{E - E_\infty}; \quad \sigma_0^{kij} = -\sigma_y + E_\infty \cdot (\varepsilon_0^{kij} + \varepsilon_y)$$

5. The value ξ is calculated as

$$\xi = \left| \frac{\varepsilon_p - \varepsilon_0}{\varepsilon_y} \right|.$$

6. The tangent modulus and the stress are updated here. Equations (4.5), (4.3), (4.4), (4.2), and (4.7) are used consecutively to compute the transition parameter R , the normalized strain, ε^* , the normalized stress, σ^* , the current stress σ^{kij} and the current tangent modulus E_t^{kij} , respectively.

Finally, if the structure has converged as in step 17 in section 3.7, all the converged history variables are set as the current variables, meaning:

$$\varepsilon_{max}^k = \varepsilon_{max}^{kij}; \quad \varepsilon_{min}^k = \varepsilon_{min}^{kij}$$

$$\varepsilon_{pl}^k = \varepsilon_{pl}^{kij}; \quad \varepsilon_0^k = \varepsilon_0^{kij}$$

$$\varepsilon_r^k = \varepsilon_r^{kij}; \quad \varepsilon_0^k = \varepsilon_0^{kij}$$

$$\varepsilon_r^k = \varepsilon_r^{kij}; \quad \sigma_0^k = \sigma_0^{kij}$$

$$\sigma_r^k = \sigma_r^{kij}; \quad I^k = I^{kij}$$

4.3.2. Algorithm for Kent-Park

For the concrete material law, one extra history material variables are added, namely, the unload slope S . To define the material, four variables are needed, namely, the compressive yield strength f_c' , the yield strain ε_0 , the ultimate strain ε_u , and the confinement factor K . The algorithm for the material response is shown in Algorithm 4.

Algorithm 4: Concrete Material Response

```
1  $\Delta\varepsilon \leftarrow \varepsilon^{kij} - \varepsilon^{k-1}$ 
2 current history variables  $\leftarrow$  converged history variables
3 if  $\Delta\varepsilon \approx 0$  then return;
   if  $\varepsilon > 0$  then  $\sigma \leftarrow 0$ ;  $E_t \leftarrow 0$ ; return;
4 if  $\varepsilon^{kij} < \varepsilon^{k-1}$  then
5a | if  $\varepsilon^{kij} < \varepsilon_r^{kij}$  then
   |    $\varepsilon_r^{kij} \leftarrow \varepsilon^{kij}$ 
   |   Compute  $\sigma^{kij}$  as in eq. (4.8)
   |   Compute  $E_t^{kij}$ 
   |    $\varepsilon_{temp} \leftarrow \min(\varepsilon_r^{kij}, \varepsilon_u)$ 
   |   Compute  $\varepsilon_p^{kij}$  as in eq. (4.9) using  $\varepsilon_{temp}$  as  $\varepsilon_r$ 
   |    $S^{kij} \leftarrow \sigma^{kij} / (\varepsilon_r^{kij} - \varepsilon_p^{kij})$ 
5b | else if  $\varepsilon^{kij} < \varepsilon_p^{kij}$  then
   |    $\sigma^{kij} \leftarrow S^{kij} \cdot (\varepsilon_r^{kij} - \varepsilon_p^{kij})$ 
   |    $E_t^{kij} \leftarrow S^{kij}$ 
5c | else
   |    $\sigma^{kij} \leftarrow 0$ 
   |    $E_t^{kij} \leftarrow 0$ 
6 else if  $\varepsilon^{kij} < \varepsilon_p^{kij}$  then
   |    $\sigma^{kij} \leftarrow S^{kij} \cdot (\varepsilon_r^{kij} - \varepsilon_p^{kij})$ 
   |    $E_t^{kij} \leftarrow S^{kij}$ 
   else
   |    $\sigma^{kij} \leftarrow 0$ 
   |    $E_t^{kij} \leftarrow 0$ 
```

To explain Algorithm 4 in the context of the nested loops mentioned discussed in section 3.7, the material response determination step comes in step 10 in the element equilibrium iteration loop in section 3.7 above. So, at every iteration, after setting the strain of the fiber, ε , the following takes place:

1. The strain difference, $\Delta\varepsilon$ is calculated as the difference between the current strain, ε^{kij} , and the converged strain from last load step, ε^{k-1} .

2. The current history variables are reset with the values of the converged variables, meaning:

$$\begin{aligned}\varepsilon_r^{kij} &= \varepsilon_r^{k-1}; & \varepsilon_p^{kij} &= \varepsilon_p^{k-1} \\ \sigma^{kij} &= \sigma^{k-1}; & \varepsilon^{kij} &= \varepsilon^{k-1} \\ S^{kij} &= S^{k-1}; & E_t^{kij} &= E_t^{k-1}\end{aligned}$$

3. If the strain difference, $\Delta\varepsilon$ is nearly zero, the strain is considered not changed and the material response is **concluded**. Also, if the new strain, ε^{kij} , is positive, the stress and the tangent modulus are set to zero, because it is assumed that the concrete material has zero tensile stiffness. Also the material response would be **concluded**.
4. There are three cases, the material goes further into compression, goes towards tension, or goes further into tension. This is checked with the values of the current strain, ε^{kij} , and the current plastic strain, ε_p^{kij} .
5. If the current strain is less than the converged strain, $\varepsilon^{kij} < \varepsilon^{k-1}$, the material is going further into compression. The material going into compression can also be subdivided into three different cases, loading, reloading, or cracked.
- a. If the current strain is less than the current reversal strain, $\varepsilon^{kij} < \varepsilon_r^{kij}$, **loading** is happening, which could be on the open-crack path, the reloading path or the loading path (cases 1, 2, and 3 in Figure 14). So, the reversal strain is set as the current strain, $\varepsilon_r^{kij} \leftarrow \varepsilon^{kij}$, then as in eq. (4.8):

- if $0 > \varepsilon^{kij} > \varepsilon_0 \implies E_t^{kij} = \frac{2 \cdot f'_c}{\varepsilon_0 - \varepsilon^{kij}} \cdot \left[1 - \left(\frac{\varepsilon}{\varepsilon_0} \right) \right]$
and $\sigma^{kij} = f'_c \cdot \left[2 \cdot \left(\frac{\varepsilon}{\varepsilon_0} \right) - \left(\frac{\varepsilon}{\varepsilon_0} \right)^2 \right]$
- if $\varepsilon_0 > \varepsilon^{kij} \geq \varepsilon_u \implies E_t^{kij} = \frac{f'_c - 0.2f'_c}{\varepsilon_0 - \varepsilon_u}$
and $\sigma^{kij} = f'_c + E_t^{kij} \cdot (\varepsilon^{kij} - \varepsilon_0)$
- if $\varepsilon_u > \varepsilon^{kij} \implies E_t^{kij} = 0$ and $\sigma^{kij} = 0.2f'_c$

Then, the unloading slope, S^{kij} , and the plastic strain, ε_p^{kij} , are updated. Eq. (4.9) is used to update the plastic strain, ε_p^{kij} . The unloading slope is calculated as

$$S^{kij} = \frac{\sigma^{kij}}{\varepsilon_r^{kij} - \varepsilon_p^{kij}}$$

5. Code Structure

The code used to implement the element uses the objective oriented programming method, in which, the code is divided into classes. Each class has its own attributes and methods, to allow a modular way of programming and to allow further improvements. It also uses inheritance, where the fiber beam-column element could inherit from a base class *element* to use its features. Also, the uniaxial constitutive laws could also have a base class of shared methods and attributes.

In general, the **element** consists of an object of a class. This object has a list of the sections as an attribute, because it derives its properties from them. Also, it has the methods to get and update the matrices of the element, and to iterate until the equilibrium of the element. Similarly, the object **section** has a list of the fibers as an attribute, because it derives its properties from them. It also has methods to get and update the section matrices from the fibers. Last but not least, each object of the class **fiber** has an object of the **uniaxial material** that defines the stress-strain relationship that could be either **concrete material** or **steel material**. These classes are shown in Figure 15.

The Python as a stand-alone code allows more flexibility in the implementation, hence it was chosen at the beginning to develop an element that gives accurate results, where performance is not a priority. This code is made in a way that can be easily extendible for multiple elements, however, only one elements test cases have been carried on. In the Python code, the main file is to construct the structure with all its elements, section, fibers and material laws, prescribe the displacement in the current load step, and to advance in load steps using the NR algorithm. Structure is a class to hold the nodes, elements, Dirichlet and Neumann boundary conditions, the system stiffness matrix and resisting forces, the unbalance forces, and the increments and the values of the displacements and the load factor. It has methods to add the elements and the boundary conditions.

The method **Initialize()** initializes all the elements, which in turn initializes all the sections, fibers and materials, to update the stiffness matrix of the structure. Then, the method **solve_NR_iteration()** is called, which calls the **state_determination()** in each class consecutively. If the NR iteration has converged, the method **finalize_load_step()** is called in each function.

The code structure for KRATOS, however, does not include the structure class, as it already contains what generates the global system and relates the elements and condition. It also contains the NR iteration algorithm. The difference is that the state determination function in the element is called **FinalizeNonLinearIteration()**, to be called after every iteration.



Figure 15 Code Structure UML diagram

6. Numerical Results

6.1. Example 1: Moment-Curvature

The Fiber Beam-Column Element was implemented in a stand-alone Python code [18] as well as in KRATOS. A test case has been done to validate the element based on an experiment carried out by Kent in [16] and simulated as well in [2]. The problem is a cantilever beam of 9 ft. span clamped from one edge and is free from the other edge. The reinforcement consists of two bars at the top and two bars at the bottom of $\frac{1}{2}$ " diameter with a concrete cover of 1". The beam is loaded statically and cyclically to failure. The beam cross-section and discretization can be shown in Figure 16, while the beam is discretized into four sections. The material parameters are summarized in Table 1. The concrete fibers inside the perimeter of the steel fibers are considered confined, and the outer ones are non-confined.

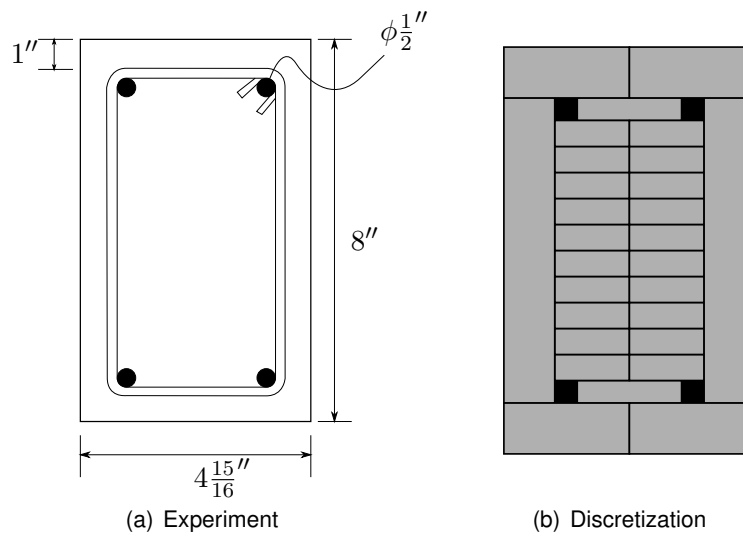


Figure 16 Beam cross-section of Example 1.

The experimental moment-curvature result from the article by Kent is shown in Figure 17(a) and the result using KRATOS is shown in Figure 17(b). The load step size used here is 0.4 and the tolerance value used for convergence is 10^{-9} .

As depicted in Figure 17(b), the Python code results are close to the experimental data. However, the results of the element implemented in KRATOS are erroneous due to a convergence issue. The element with the explained constitutive laws, Menegotto-Pinto and Kent-Park, did not converge even with a high tolerance value of the element equilibrium loop. The problem could be in the implementation of the constitutive laws as the solution would start diverging when the load starts reversing in the fibers. The converged result obtained and shown

(a) Confined Concrete

f'_c [ksi]	ε_0	ε_u
6.95	0.0027	0.0381

(b) Unconfined Concrete

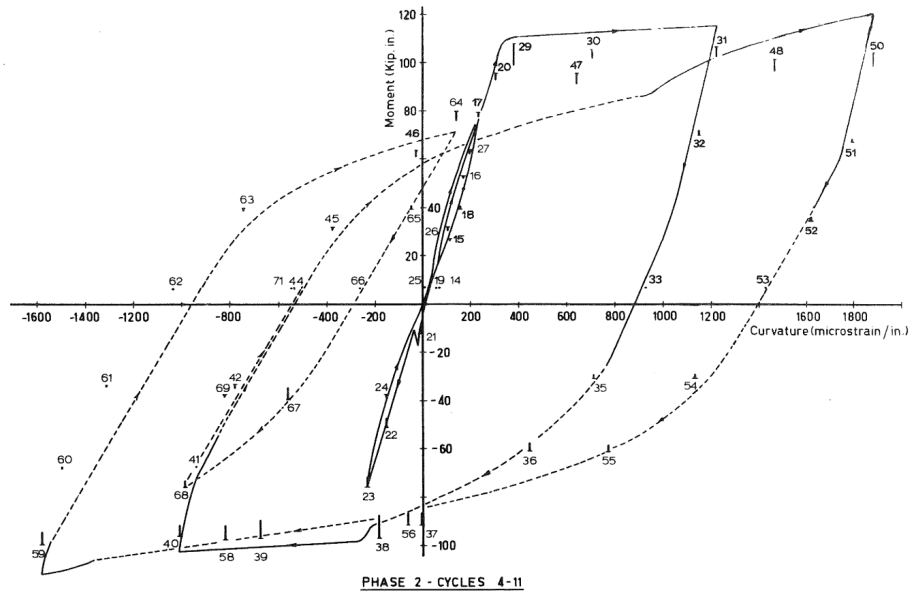
f'_c [ksi]	ε_0	ε_u
6.95	0.0027	0.00292

(c) Steel

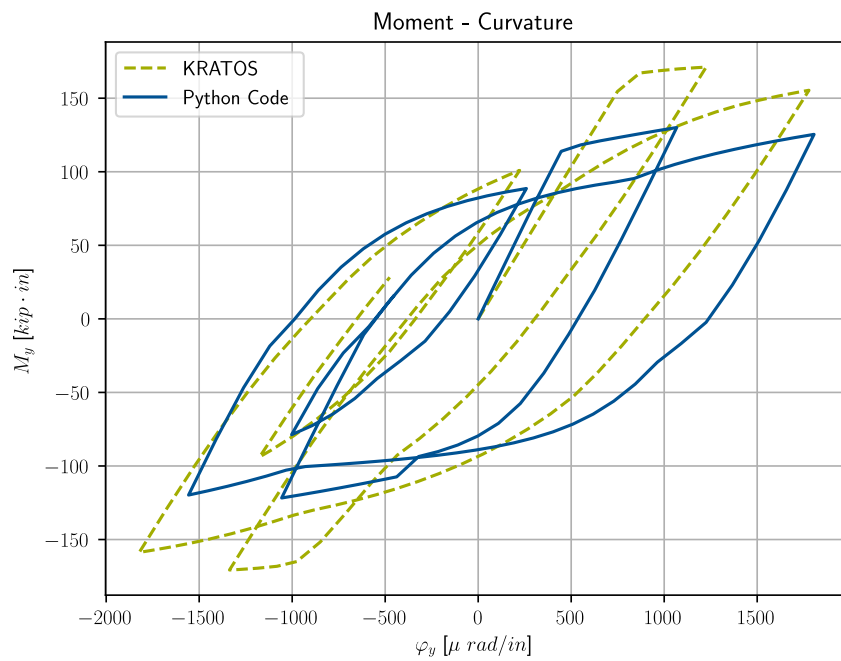
E	b	σ_y [ksi]	R	a_1	a_2
29000	0.0042	48.4	20	18.5	0.0002

Table 1 Material parameters of Example 1.

in Figure 17(b) is with a simplified concrete constitutive law where the softening effect is dismissed and the concrete is assumed to be elastic and fully plastic in compression, and has a zero stiffness in tension. This would result in reduction of the whole element tangent stiffness as can be seen in the figure. Nonetheless, the Python code results are more accurate and therefore validates the implementation for this test case. Moreover, a test for the step size and number of sections is shown in Figure 18, where the x-axis is the controlled displacement of the free end and the y-axis is the applied load factor, where the load is 1 *kip*. One can notice that refining the step size would lead to a better solution but the pattern does not change. However, increasing the number of sections leads to a more accurate solution. In the simulation with three sections, the solutions differs to the experiment, and the hardening of the steel fibers is not well modeled. With increasing number of sections, the solution converges, but the computational complexity increases as smaller step sizes are needed to reach convergence of the element equilibrium. Here, for 10 sections and 20 sections, a step size of 0.05 was used.

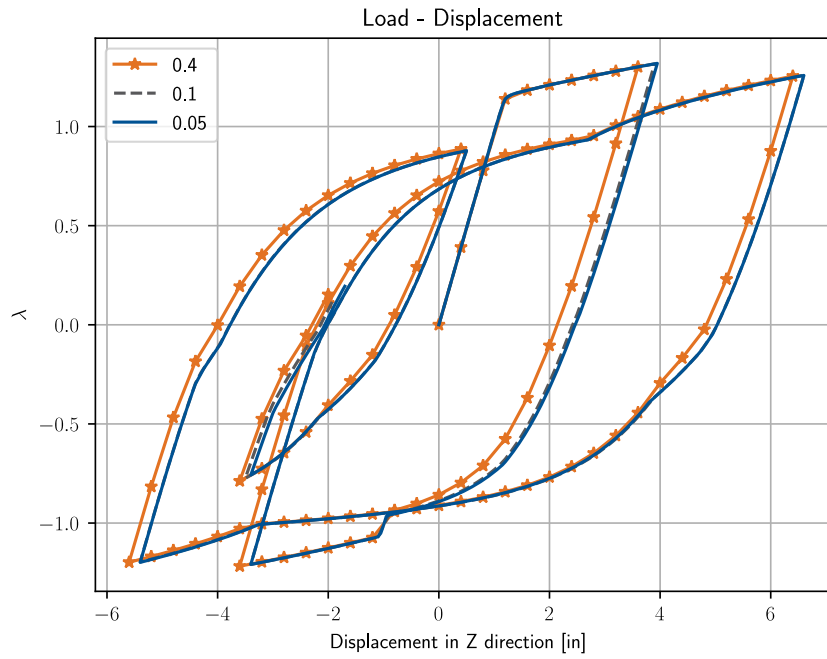


(a) Experiment. [16]

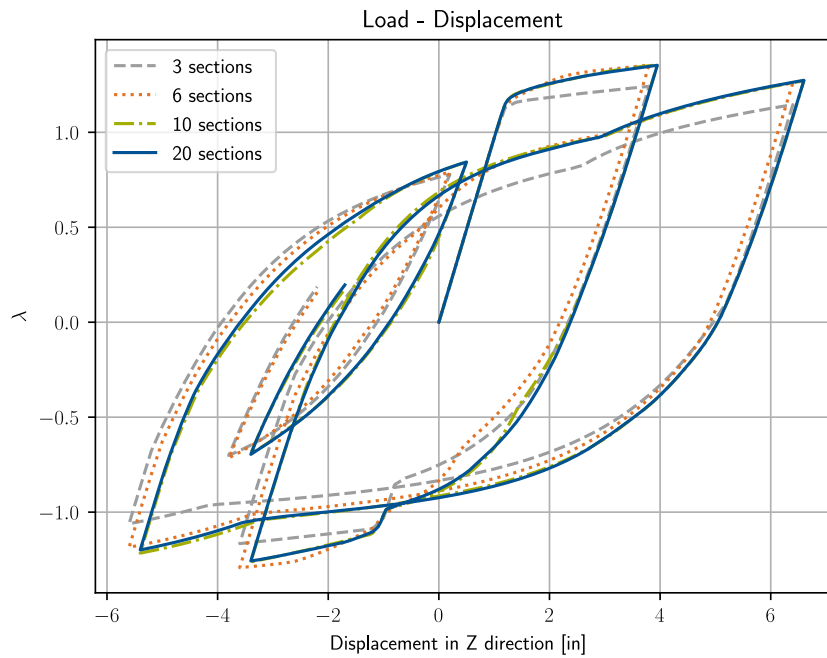


(b) Simulation using the Python code.

Figure 17 Moment on the clamped-end against Curvature on the free end of Example 1.



(a) Different load step sizes.



(b) Different number of sections.

Figure 18 Load-Displacement plot of Example 1 using the Python code.

6.2. Example 2: Uni-axial Bending

This example simulates the uniaxial bending effect of a cantilever beam. The beam spans 71", and is clamped from one end and free from the other. An axial load is statically applied on the free end and repeated cyclically until failure. The experiment is explained as the R-1 cantilever beam in the paper by Ma et. al. [19], where the experiment result is shown here in Figure 20. The beam cross-section as described in the paper is shown in Figure 19(a), where the cross-section shown is not doubly symmetric as in Example 1. The concrete cover is 1" from the bottom, 2" from the top, and 0.75" from the sides. The beam is 16" deep and 9" wide. The steel reinforcement consists of three #5 (0.625" diameter) in the bottom and #6 (0.75" diameter) in the top ASTM A615M Grade 60 rebars. The discrete beam element is sliced into 4 section, where each section is discretized as in 19(b), where it consists of 15 confined concrete fibers and 4 unconfined fibers. The material parameters used for the steel and the concrete constitutive laws are shown in Table 2.

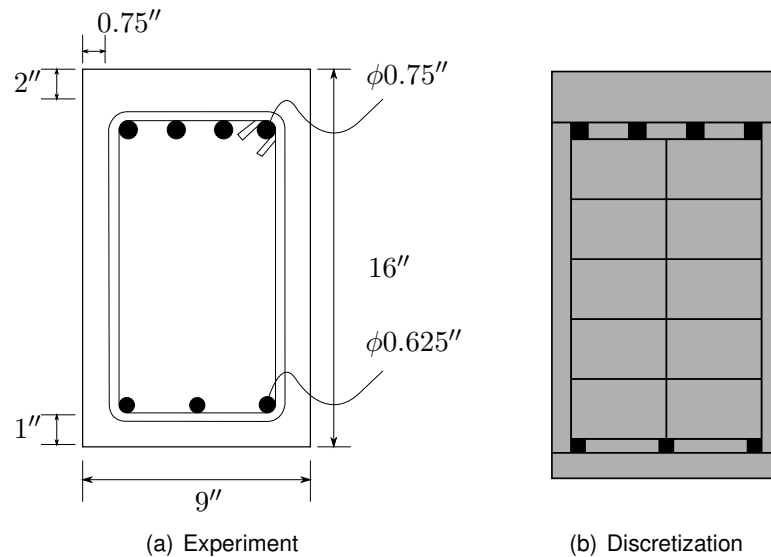


Figure 19 Beam cross-section of Example 2.

The result of the experiment is shown in Figure 20, where the result of the simulations by the paper by Taucer [2] and the result of the Python code are in Figure 21. Here, the results of the simulation differ from the experiment due to the effect of bars pull-out and shear deformations, which are not accounted for in this model. This shows in the deformation of the cycles after yield of the fibers.

(a) Confined Concrete

f_c' [ksi]	ϵ_0	ϵ_u
5.43	0.00214	0.069

(b) Unconfined Concrete

f_c' [ksi]	ϵ_0	ϵ_u
5.07	0.002	0.003

(c) Steel

E	b	σ_y [ksi]	R	a_1	a_2
29000	0.0085	66.5	20	18.5	0.0002

Table 2 Material parameters of Example 2.

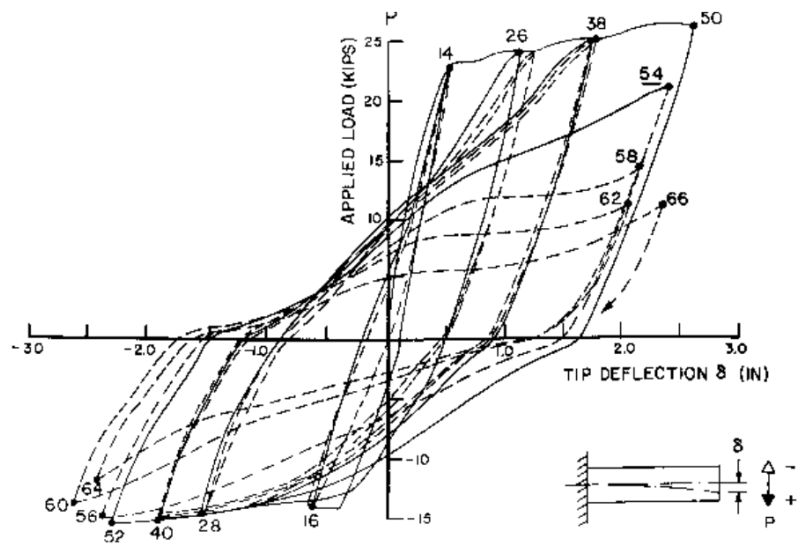
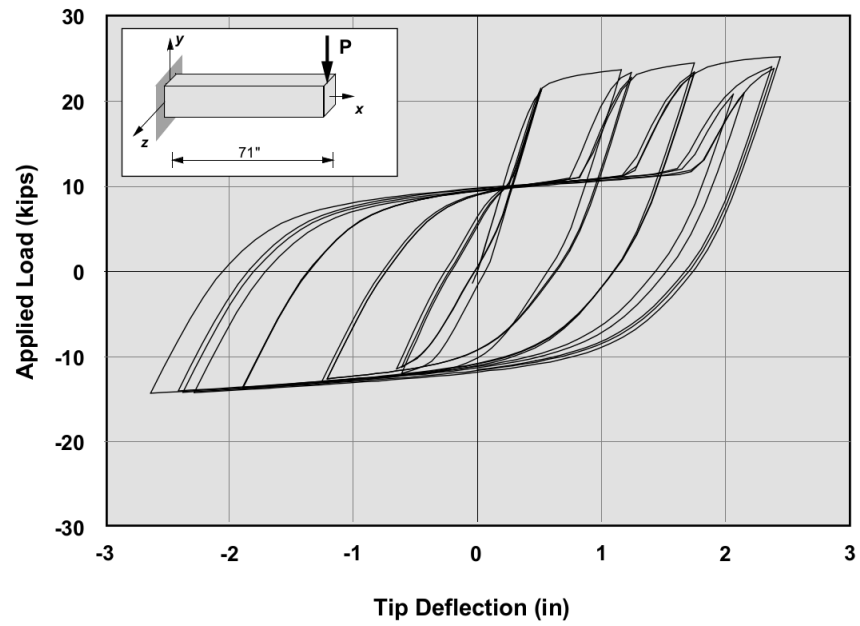
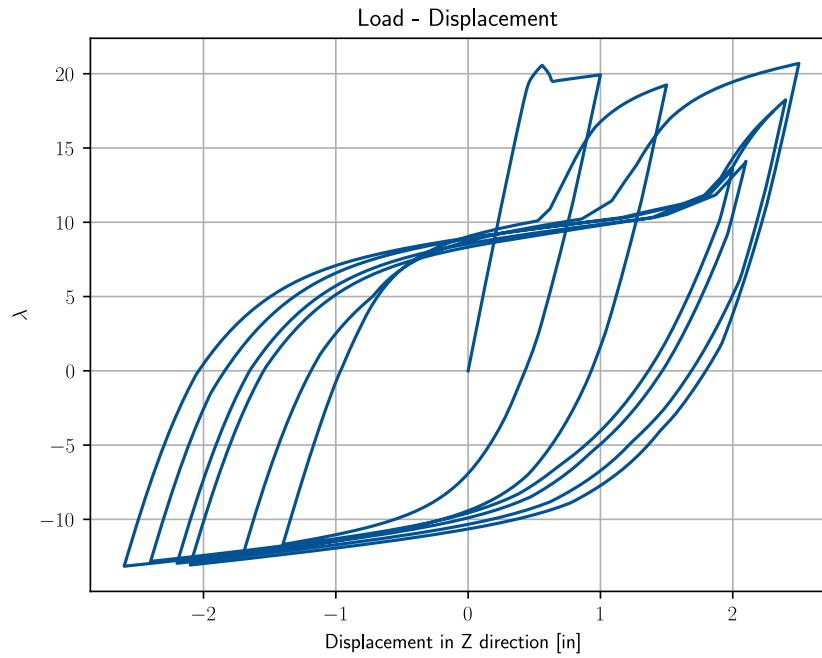


Figure 20 Experimental Load-Displacement plot of Example 2. [19]



(a) Original section.



(b) Coarse section.

Figure 21 Load-Displacement plot of Example 2.

6.3. Effect of the Discretization

The effect of the section discretization using uniaxial fiber can be shown in Figure 22 where the simulation of Example 1 was carried on using two different discretization schemes, one with 10 confined concrete fibers and one with two fibers. The results of the simulation are very similar with minor differences. This can be explained due to the fact that the concrete fibers are loaded beyond their yield point, so they have high plastic strains and contribute considerably much less to the element stiffness. However, using more fibers would result in having to use smaller time steps to reach convergence.

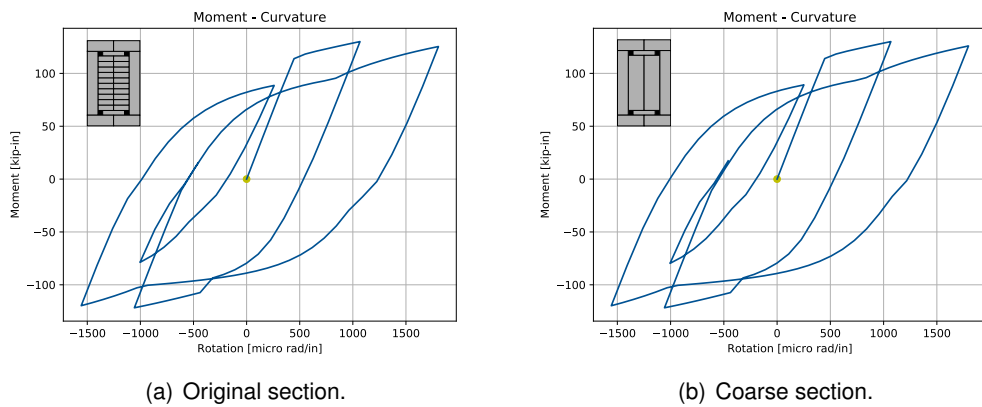
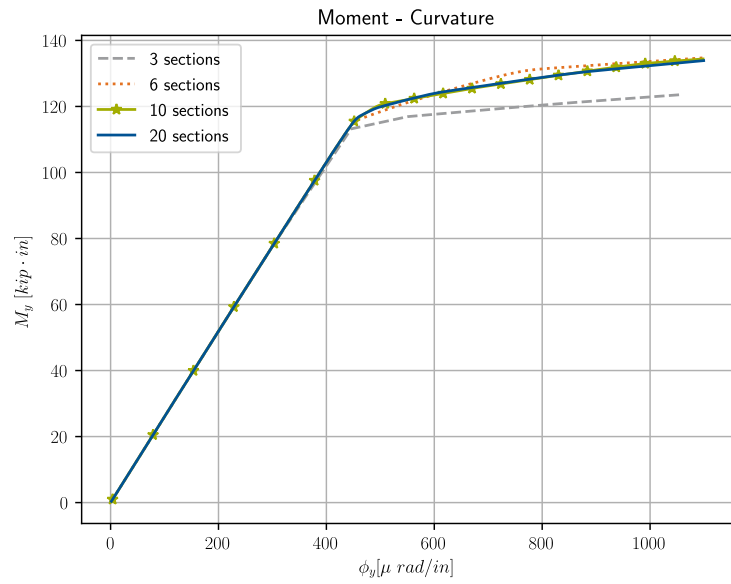
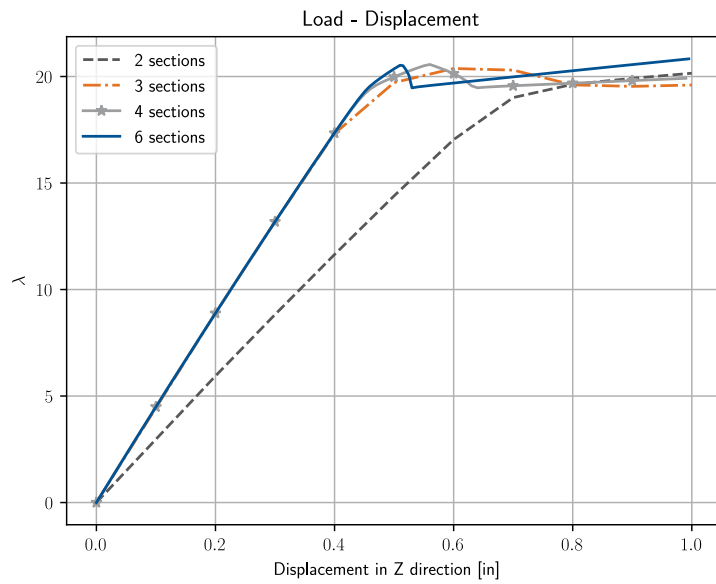


Figure 22 Moment-Curvature plot with different fiber discretization of Example 1.

On the other hand, choosing the number of sections, hence the number of integration points, plays a great role in the solution. Figure 23 shows the solution with different number of sections. In the linear phase, the number of sections is insignificant because the effect of inelasticity are not there. However, in the post-yielding loading, the flexibility matrix of the element is better represented with more sections, since the curvature along the element becomes non-linear. Also here, choosing more sections would require smaller load step size.



(a) Original section.



(b) Coarse section.

Figure 23 Results with different number of sections.

7. Conclusions and Outlook

The objective of this work is to implement an element to model Reinforced Concrete under dynamic loading to model seismic effects on a beam-column and response by axial forces and biaxial moments. The element implemented in a Python stand-alone code [18] and in the open-source Multi-physics software KRATOS [1], is an element derived by Taucer [2], as it is the first original well derived, reliable, and computationally efficient flexibility-based fiber beam-column element.

The element derives its stiffness from integrating the flexibility over sections, which in turn derive their flexibility from individual uniaxial fibers, each having an individual uniaxial stress-strain relationship. Therefore, any uniaxial constitutive law can be implemented on the fiber level. However, the interaction between fibers leading to bond-slip is not modeled. Also, it assumes small deformations and sections normal to the longitudinal axis. The element being flexibility based and solved using displacement-controlled non-linear FEM makes it represent the internal forces exactly regardless of the non-linearities in the system, and models the element softening.

The theory behind the element as well as the implementation is shown in this Thesis. The Python code delivers accurate converging results. However, the KRATOS code suffers from convergence problems. Also, the KRATOS implementation is done on a *developer* version, so it lacks the graphical user interface to apply the section discretization.

Further work to this Thesis could include:

- Fixing the implementation in KRATOS code.
- Improving the code to solve multiple elements, such as a frame of beams and columns.
- Studying the effect of the 2D section discretization of fibers.
- Studying the effects of more complicated constitutive laws to include damage of concrete due to cyclic loading and steel buckling.
- Adding non-linear geometric stiffness to model larger deformations.
- Adding dynamics to model the impact of mass and damping to the system.
- Adding shear forces to model a wider range of problems in which the shear forces affect the behavior of the element.
- Implementing the graphical user interface to discretize the sections.

A. Appendix

A.1. Gauss-Lobatto Quadrature Rule

Gaussian quadrature is an approximate integral using a discrete set of points in the range $[-1, 1]$, where the integrand is evaluated on the points. By definition

$$\int_{-1}^1 f(x) dx \approx \sum_i^n f(x_i) \cdot w_i, \quad (\text{A.1})$$

where w_i is the weight of each point.

The commonly used Gauss-Legendre quadrature rule is exact to integrate polynomials up to degree $2n - 1$. However, this method does not include the end-point of the Gaussian domain, 1 and -1 . Another rule, namely Gauss-Lobatto quadrature or Radau quadrature, includes the end-points and the integral is exact for polynomials up to a degree of $2n - 3$. The flexibility-based fiber beam-column element has highly non-linear curvatures at the fixed ends of the beam, which makes the Gauss-Lobatto rule a more suitable method.

n	x_i	w_i
2	± 1	1
3	0 ± 1	4/3 1/3
4	$\pm \sqrt{1/5}$ ± 1	5/6 1/6

Table 3 Example of Gauss-Lobatto quadrature rule points and weights.

Table 3 shows examples of Gauss-Lobatto points and weights. In general, the x values of an arbitrary number of points are the roots of P'_{n-1} , where P_n is the Legendre polynomial for n number of points, and the weight of each point is

$$w_i = \frac{2}{n \cdot (n - 1) \cdot [P_{n-1}(x_i)]^2}. \quad (\text{A.2})$$

List of Figures

Figure 1	Non-linear load displacement curve example.....	3
Figure 2	Illustration of the different constraint approaches.....	5
Figure 3	Two trusses snap-through problem. [5].....	7
Figure 4	Circular Arch truss system.	7
Figure 5	Evolution of the system.....	8
Figure 6	Solution of circular Arch truss system.	8
Figure 7	Discretization of the fiber beam-column element.	10
Figure 8	Element local forces.	11
Figure 9	Illustration of the three nested loops for three load steps on the element and the section level. [2]	19
Figure 10	Illustration of the change of parameters in one load step on the element and the section level. [2]	20
Figure 11	Stress-strain relationship of steel fibers.....	26
Figure 12	The Bauschinger effect in the steel fibers described by R and ξ	27
Figure 13	Stress-strain relationship of concrete fibers.	29
Figure 14	Different cases of material state determination of concrete fibers.	35
Figure 15	Code Structure UML diagram.....	37
Figure 16	Beam cross-section of Example 1.	38
Figure 17	Moment on the clamped-end against Curvature on the free end of Example 1.	40
Figure 18	Load-Displacement plot of Example 1 using the Python code.	41
Figure 19	Beam cross-section of Example 2.	42
Figure 20	Experimental Load-Displacement plot of Example 2. [19]	43
Figure 21	Load-Displacement plot of Example 2.	44
Figure 22	Moment-Curvature plot with different fiber discretization of Example 1.....	45
Figure 23	Results with different number of sections.	46

List of Tables

Table 1	Material parameters of Example 1.....	39
Table 2	Material parameters of Example 2.....	43
Table 3	Example of Gauss-Lobatto quadrature rule points and weights.	48

Bibliography

- [1] Kratos. <http://www.cimne.com/kratos/>.
- [2] Fabio Taucer, Enrico Spacone, and Filip Filippou. A fiber beam-column element for seismic response analysis of reinforced concrete structures. 01 1991.
- [3] Carlos Filippa. Lecture notes in nonlinear finite element methods, 2004.
- [4] Junuthula Narasimha Reddy. *Introduction to the finite element method*. McGraw-Hill Education, 2019.
- [5] Steen Krenk. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University Press, 2009.
- [6] M. A. Crisfield. *Non-linear finite element analysis of solids and structures*, volume 2. Wiley, 1997.
- [7] Ray W. Clough and Kalmar L. Benuska. Nonlinear earthquake behavior of tall buildings. *Journal of the Engineering Mechanics Division*, 93(3):129–146, 1967.
- [8] "notes on some basic problems in inelastic analysis of planar rc structures.". *Transactions of the Architectural Institute of Japan*, 1967.
- [9] Shunsuke Otani. Inelastic analysis of r/c frame structures. *Journal of the Structural Division*, 100(7):1433–1449, 1974.
- [10] Takehira Takayanagi and William C. Schnobrich. Non-linear analysis of coupled wall systems. *Earthquake Engineering & Structural Dynamics*, 7(1):1–22, 1979.
- [11] Said A. Kaba and Stephan A. Mahin. *Refined Modeling of Reinforced Concrete Columns for Seismic Analysis*. EARTHQUAKE ENGINEERING RESEARCH CENTER, 1984.
- [12] Christos A. Zis and Stephen A. Mahin. Behavior of reinforced concrete structures subjected to biaxial excitation. *Journal of Structural Engineering*, 117(9):2657–2673, 1991.
- [13] O. C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, Butterworth-Heinemann, 2013.
- [14] M. Menegotto and P. E. Pinto. *Method of Analysis for Cyclic Loaded R. C. Plane Frame*

Including Changes in Geometry and Non-Elastic Behaviour of Elements under Combined Normal Force and Bending, volume 11, pages 15–22. 1973.

- [15] PA Kelly. Lecture notes: Solid mechanics part ii: Engineering solid mechanics – small strain.
- [16] D. C. Kent. *Inelastic Behavior of Reinforced Concrete Members with Cyclic Loading*. PhD thesis, University of Canterbury, 1969.
- [17] Opensees. <https://github.com/OpenSees/OpenSees>.
- [18] Stand-alone python implementation. https://github.com/zidanmahmoud/fiber_beam_column.
- [19] S. Ma, V. Bertero, and E. Popov. *Experimental and Analytical Studies on the Hysteretic Behavior of Reinforced Concrete Rectangular and T-Beams*. EARTHQUAKE ENGINEERING RESEARCH CENTER, 1976.

Declaration

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. In addition, I declare that I make the present work available to the Chair of Structural Analysis for academic purposes and in this connection also approve of dissemination for academic purposes.

Munich, 21.11.2019, Signature