

The Pennsylvania State University

The Graduate School

**A GLOBAL PARAMETRIC NONLINEAR PROGRAMMING FRAMEWORK FOR A  
GEOMETRICALLY EXACT INELASTIC BEAM FINITE ELEMENT**

A Dissertation

in Civil Engineering

by

Charilaos Lyritsakis

© 2022 Charilaos Lyritsakis

Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Doctor of Philosophy

December 2022

The dissertation of Charilaos Lyritsakis was reviewed and approved by the following:

Kostas Papakonstantinou  
Associate Professor of Civil & Environmental Engineering  
Dissertation Advisor  
Chair of Committee

Gordon Warn  
Professor of Civil & Environmental Engineering

Pinlei Chen  
Assistant Professor of Civil & Environmental Engineering

Tong Qiu  
Professor of Civil & Environmental Engineering

Reuben Kraft  
Associate Professor of Mechanical Engineering

Patrick Fox  
Professor of Civil & Environmental Engineering  
Head of Department of Civil & Environmental Engineering

# Abstract

A novel framework for the structural analysis of planar frames is developed, which combines tools from finite element analysis, mathematical programming and homotopy theory. The core building block in this work is a versatile, high-performance, hybrid beam-column element which is formulated based on nonlinear programming principles. The geometrically-exact kinematic assumptions are adopted, thereby allowing for arbitrarily large displacements and rotations, while inelastic behavior is modeled by assuming a discretization of selected cross-sections into layers or fibers, which can incorporate a multiaxial constitutive law but act independently of neighboring layers. The element is capable of capturing both geometric and material nonlinearities with just one element per structural member while maintaining all the numerically attractive properties pertaining to the structure of the resulting global stiffness matrix.

The interaction between axial, shear and flexural effects, especially during inelastic deformation, is accounted for by incorporating a multiaxial constitutive law at the level of cross-section fibers. In particular, a fast and robust return-mapping algorithm is developed which is utilizing the zero transverse normal stress assumption in order to arrive at a reduced stress space formulation. This allows for a more efficient stress update both in terms of memory and computation costs. The implementation assumes a  $J_2$  von Mises material with combined isotropic and kinematic hardening. Such reformulation of general three-dimensional or axisymmetric stress update procedures is crucial for beam elements that rely on scarce meshes but higher order quadratures in order to achieve accuracy, since elastoplastic analyses by fibre-discretized elements increases the computational cost considerably.

The proposed beam element is embedded in a parametric nonlinear programming framework which is developed to facilitate analysis considerations beyond response to mechanical loading. In particular, we take advantage of the optimization reformulation of the underlying variational structure of the mechanics problem and develop a naturally parameterized nonlinear programming framework which can easily handle parametric investigations, such as design

optimization or sensitivity analysis, while also providing the theoretical and numerical tools that ensure global convergence characteristics, provided certain regularity conditions hold. A predictor-corrector type numerical continuation algorithm suited for this framework is also developed which can account for any parameterization and derivative discontinuities along the solution path.

To improve the performance of the aforementioned numerical continuation algorithm, a reliable and efficient prediction scheme is proposed that is considerably faster than conventional approaches that utilize second derivative information. Instead, a weighted least squares fitting is carried out at the start of an incremental step, provided that a number of previously converged solution points are stored and are available in memory. The curve generated by this fitting process is capable to emulate the local geometry of the actual solution curve much better since an interpolating condition is not enforced. In addition, by using an appropriate weighting function, we can assign increased weights to solutions closer to the current step. This leads to a versatile scheme that can provide a suite of additional options, as far as general predictor-corrector algorithms are concerned and, more importantly, it can be applied for any problem which can be solved by such procedures. It can be particularly attractive for problems described by dense, non-symmetric Jacobians, especially if coupled with iterative correction schemes such as conjugate gradients.

The present work is partitioned into four parts: the core hybrid beam element, the multiaxial constitutive model and return mapping algorithm, the naturally parameterized framework for the hybrid element and, finally, the weighted least squares predictor scheme. To validate the efficacy and accuracy of the procedures associated with each individual part developed in this work, several numerical tests are presented in the thesis.

# Table of Contents

<b>List of Figures . . . . .</b>	viii
<b>List of Tables . . . . .</b>	xiv
<b>Nomenclature . . . . .</b>	xv
<b>Acronyms . . . . .</b>	xx
<b>Acknowledgements . . . . .</b>	xxii
<b>Chapter 1: Introduction and Background . . . . .</b>	1
1.1 Literature Review . . . . .	1
1.1.1 Beam Element Technologies . . . . .	1
1.1.2 Modelling Inelastic Behavior in Beams . . . . .	3
1.1.3 Local and Global Solution Methods . . . . .	4
1.2 Research Objectives . . . . .	6
1.3 Research Scope . . . . .	8
1.4 Organization of the Manuscript . . . . .	10
<b>Chapter 2: Hybrid Beam Element Formulation . . . . .</b>	12
2.1 Introduction . . . . .	12
2.2 Geometrically-Exact Model . . . . .	13
2.2.1 Beam Kinematics . . . . .	13
2.2.2 Strain Measures, Strain-Displacement Relations . . . . .	15
2.3 Nonlinear Programming Formulation . . . . .	17
2.3.1 NLP Framework . . . . .	17
2.3.2 Total Potential Energy . . . . .	18
2.3.3 Strain-Displacement Constraints . . . . .	19
2.3.4 Lagrangian Function . . . . .	21
2.4 Implementation . . . . .	22
2.4.1 Equilibrium . . . . .	24
2.4.2 Hessian Matrix . . . . .	25
2.4.3 Solution Scheme . . . . .	25
2.4.4 Cross-Section State Determination . . . . .	30
2.5 Numerical Examples . . . . .	34
2.5.1 Cantilever with Tip Load . . . . .	35
2.5.2 Lee's Frame . . . . .	39

2.5.3	Two-Storey Frame . . . . .	43
2.6	Summary . . . . .	45
<b>Chapter 3: Material Constitutive Model</b>	. . . . .	<b>47</b>
3.1	Introduction . . . . .	47
3.2	Three-dimensional Constitutive Model . . . . .	48
3.3	The $J_2$ Formulation for Fiber Stress State . . . . .	50
3.3.1	Mapping on the Constrained Stress Space . . . . .	50
3.3.2	Continuous Tangent Modulus . . . . .	54
3.4	Implicit Integration of Rate Equations . . . . .	54
3.4.1	Elastic Predictor - Plastic Corrector algorithm . . . . .	55
3.4.2	Consistent Tangent Modulus . . . . .	57
3.4.3	Consistent Section Stiffness . . . . .	58
3.5	Numerical Examples . . . . .	60
3.5.1	Accuracy of the Proposed Algorithm . . . . .	60
3.5.2	Performance Against Non-Monotonic Strain Histories . . . . .	62
3.5.3	Plastic Collapse of a Doubly Clamped Beam . . . . .	63
3.5.4	Cyclic Loading of Shear Link . . . . .	69
3.5.5	Elastoplastic Post-Buckling Behavior of Two-Beam Structure . . . . .	72
3.5.6	Three Story Frame With Shear Links . . . . .	75
3.6	Summary . . . . .	76
<b>Chapter 4: Naturally Parameterized NLP framework</b>	. . . . .	<b>78</b>
4.1	Introduction . . . . .	78
4.2	Homotopy Continuation . . . . .	80
4.2.1	Main idea . . . . .	80
4.2.2	Regularity and Existence of Homotopy Paths . . . . .	83
4.2.3	Parametric Nonlinear Programming . . . . .	85
4.2.4	Global Optimization . . . . .	89
4.3	Naturally Parameterized NLP Framework for Mechanics . . . . .	91
4.3.1	Existence and Uniqueness of NPNLP Path . . . . .	93
4.4	Implementation . . . . .	96
4.5	Numerical Examples . . . . .	99
4.5.1	Example 1 - Global Optimization . . . . .	99
4.5.2	Example 2 - Global Sensitivity . . . . .	102
4.5.3	Example 3 - Frictionless Contact . . . . .	105
4.5.4	Example 4 - Nonlinear Analysis of Shallow Arch . . . . .	108
4.6	Summary . . . . .	109
<b>Chapter 5: Weighted least squares predictor for numerical continuation</b>	. . . . .	<b>111</b>
5.1	Introduction . . . . .	111
5.2	Numerical Continuation . . . . .	112
5.3	Weighted Least Squares Predictor . . . . .	115
5.3.1	Matrix Formulation of WLS Approximation . . . . .	118
5.3.2	WLS Prediction Variants . . . . .	120

5.3.3	Arc-length Stabilization . . . . .	121
5.3.4	The Weight Function . . . . .	122
5.4	Implementation . . . . .	123
5.4.1	Algorithm . . . . .	123
5.4.2	Arc-Length Measurement . . . . .	124
5.4.3	Discussion . . . . .	125
5.5	Numerical Examples . . . . .	127
5.5.1	Numerical Example 1 . . . . .	128
5.5.2	Numerical Example 2 . . . . .	129
5.5.3	Numerical Example 3 . . . . .	131
5.5.4	Numerical Example 4 . . . . .	133
5.6	Summary . . . . .	135
<b>Chapter 6: Conclusions and Suggestions for Future Research</b>	. . . . .	<b>136</b>
6.1	Research Summary and Conclusions . . . . .	136
6.2	Limitations . . . . .	136
6.3	Future Directions . . . . .	137
<b>Appendix A: Notation</b>	. . . . .	<b>141</b>
<b>Appendix B: Gradient of Hybrid NLP Lagrangian</b>	. . . . .	<b>145</b>
<b>Appendix C: Hessian of Hybrid NLP Lagrangian</b>	. . . . .	<b>150</b>
<b>Appendix D: Iterative Corrections for Plastic Parameter</b>	. . . . .	<b>153</b>
<b>Appendix E: Analytical Derivation of Consistent Tangent Modulus</b>	. . . . .	<b>155</b>
<b>Appendix F: Numerical Path-following</b>	. . . . .	<b>157</b>
<b>Appendix G: A Simple GUI for the Hybrid NLP Element</b>	. . . . .	<b>160</b>
<b>References</b>	. . . . .	<b>205</b>

## List of Figures

<b>Figure 2.1:</b>	Undeformed and deformed configurations of the beam.	14
<b>Figure 2.2:</b>	Integration of curvature shape functions.	21
<b>Figure 2.3:</b>	Typical element representing one member.	23
<b>Figure 2.4:</b>	Solution process stages. Quantities $\mathbf{K}^e$ and $\vec{\mathbf{F}}_{int}$ stand for stiffness matrix and internal nodal force vector for element $e$ respectively.	33
<b>Figure 2.5:</b>	Geometrically nonlinear analyses of cantilever.	37
<b>Figure 2.6:</b>	Moment and curvature distributions for the nonlinear analyses of the cantilever - one element with $n = 6$ quadrature points.	37
<b>Figure 2.7:</b>	Influence of $L/h$ on the tip deflection and comparison with analytical solution.	38
<b>Figure 2.8:</b>	Geometrically nonlinear analyses of Lee's frame with pinned supports.	40
<b>Figure 2.9:</b>	Deformation profiles for Lee's frame with pinned supports.	40
<b>Figure 2.10:</b>	Effect of shear flexibility on Lee's frame response.	41
<b>Figure 2.11:</b>	Deformation profiles for Lee's frame with fixed supports.	42
<b>Figure 2.12:</b>	Geometrically nonlinear analyses of Lee's frame with fixed supports.	43
<b>Figure 2.13:</b>	Two-storey portal frame discretization.	44
<b>Figure 2.14:</b>	Geometrically nonlinear analyses of the two-storey frame.	44
<b>Figure 2.15:</b>	Deformation profiles for two-storey frame.	45
<b>Figure 3.1:</b>	Feasible space and yield locus for the planar fiber constrained $J_2$ model. At the ellipse boundary we have $\sqrt{3J_2} = q$ .	51

<b>Figure 3.2:</b> Points of interest on yield surface and direction of load increments. . . . .	61
<b>Figure 3.3:</b> Iso-error maps for a perfectly plastic material. . . . .	61
<b>Figure 3.4:</b> Iso-error maps for combined linear isotropic and kinematic hardening. .	62
<b>Figure 3.5:</b> The four different strain histories imposed, with the unstresses config- uration as initial point. . . . .	63
<b>Figure 3.6:</b> Geometry, initial and final configuration of clamped beam, along with cross-section dimensions. . . . .	66
<b>Figure 3.7:</b> Load-deflection plots for different slenderness ratios. Influence of quadra- ture type and number of points. . . . .	67
<b>Figure 3.8:</b> Case $L/h = 4$ . Spread of Plastic Zone (P.Z.) and stress distribution for von Mises, axial and shear stress at three different load levels. . . . .	68
<b>Figure 3.9:</b> Geometry of the specimen and displacement-controlled loading history.	70
<b>Figure 3.10:</b> Experimental and numerical results for the specimen. . . . .	71
<b>Figure 3.11:</b> Performance of the three different return-mapping algorithms applied for the section state determination of the shear link under the imposed displacement history. . . . .	71
<b>Figure 3.12:</b> Geometry and loading of two-beam structure. Comparison of load vs deflection curves for different cases. . . . .	73
<b>Figure 3.13:</b> Influence of slenderness and constitutive model on the response for elastic and elastoplastic behavior. . . . .	74
<b>Figure 3.14:</b> Convergence rates for two representative steps when using the consis- tent and continuous tangents at fibers. . . . .	74
<b>Figure 3.15:</b> Geometry of the steel frame and 3.15b response curves for three differ- ent models. . . . .	76
<b>Figure 4.1:</b> A homotopy path connecting the trivial solution of $G = 0$ to a root of $F = 0$ for a single scalar equation dependent on $x$ . . . . .	81

<b>Figure 4.2:</b> Homotopy paths for (a) $f(x) = x^3 - 6x^2 + 21x - 26 = 0$ , with starting point $x_0 = 0$ and convergence at root $x_1 = 2$ when $t = 1$ and (b) truss, with initial state the undeformed configuration( $x_0 = 0$ ) and target load level $P = 0.03$ . . . . .	82
<b>Figure 4.3:</b> Components of $\vec{H}^{-1}(\vec{0})$ in $\mathbb{R}^{n+1}$ . . . . .	84
<b>Figure 4.4:</b> Piecewise continuously differentiable paths and loops of $H_{KKT}^{-1}(0)$ in $\mathbb{R}^{n+l+p+1}$ . . . . .	90
<b>Figure 4.5:</b> Transition from cell $\tau^1$ to $\tau^2$ during numerical continuation. Projection of $\vec{H}(\vec{z}, t) = \vec{0}$ on $\vec{d} - t$ space. . . . .	98
<b>Figure 4.6:</b> (a) Paths $\Gamma_1$ , $\Gamma_2$ and $\Gamma_3$ in $\mathbb{R}^3$ , (b) interpretations of paths as trajectories on the target TPE function (4.32). . . . .	101
<b>Figure 4.7:</b> (a) Truss structure, (b) equilibrium paths for $A_2 = \{4A_1, A_1/3, A_1/5\}$ projected on the $u_1 - P$ plane. . . . .	103
<b>Figure 4.8:</b> (a) projection of components $\Gamma_1$ & $\Gamma_2$ on the $u_1 - E_2 A_2(t)/L$ plane, (b) paths $\Gamma_1$ , $\Gamma_2$ and $\Gamma_P$ projected on the TPE of the target NLP $\mathcal{P}_{A_2}(1)$ . . . . .	105
<b>Figure 4.9:</b> Initial geometry for Example 3. . . . .	106
<b>Figure 4.10:</b> Projection of path (a) on $u_2 - t$ plane and (b) $\mu - t$ plane. . . . .	107
<b>Figure 4.11:</b> Configuration profiles during deformation. . . . .	107
<b>Figure 4.12:</b> (a) Shallow arch structure and loading, (b) equilibrium path using the NPNLP framework. . . . .	109
<b>Figure 5.1:</b> Advancing from step $N$ to step $N + 1$ using a PC approach. . . . .	113
<b>Figure 5.2:</b> Two types of predictors: (a) interpolatory and (b) Euler predictor. . . . .	115
<b>Figure 5.3:</b> Geometric interpretation of predictors (i)WLSE (blue), (ii) WLSIT (light blue, $z = 1$ ), and (iii) WLST (red, $z = 0$ ). . . . .	121
<b>Figure 5.4:</b> Weight function $w(s)$ of the WLS predictor. . . . .	123
<b>Figure 5.5:</b> Solution path of Eq. (5.5.3). . . . .	132
<b>Figure 5.6:</b> (a) Shallow arch structure and loading, (b) equilibrium path. . . . .	133

<b>Figure F.1:</b> Two correction algorithms: Normal Flow and Normal Plane. . . . .	159
<b>Figure G.1:</b> The <i>Mesh</i> tab with relevant facilities A-M marked in red circles and enumerated with capital english letters. . . . .	163
<b>Figure G.2:</b> Error when attempting to add a node with coordinates already defined for an existing node. . . . .	163
<b>Figure G.3:</b> A .txt file with the required format for nodal input. First column is X coordinate, second is Y coordinate. The $k$ -th row represents Node $k$ . . .	164
<b>Figure G.4:</b> A .txt file with the required format for element input. First column is start node $i$ , second is end node $j$ . The $k$ -th row represents Element $k$ . .	164
<b>Figure G.5:</b> The <i>Mesh</i> tab with a completed mesh. . . . .	165
<b>Figure G.6:</b> The <i>Materials</i> tab with relevant facilities A-M marked in red circles and enumerated with capital english letters. . . . .	167
<b>Figure G.7:</b> Multiple materials defined without a specified material name. . . . .	167
<b>Figure G.8:</b> Potting of uniaxial stress-strain law of material <b>MAT_1</b> . . . . .	168
<b>Figure G.9:</b> Available predefined materials from the <i>Standard materials</i> drop-down list. . . . .	168
<b>Figure G.10:</b> Error when trying to add a standard material that was already added previously. . . . .	168
<b>Figure G.11:</b> The <i>Sections</i> tab with all essential parts marked in red cycles and enumerated. . . . .	170
<b>Figure G.12:</b> Typical section definition process. A previously defined material <b>MAT_1</b> is assigned to section <b>Rectangular_Section</b> . . . . .	171
<b>Figure G.13:</b> Successful section assignment for <b>Element 1</b> . . . . .	171
<b>Figure G.14:</b> Indication that <b>Element 2</b> still needs to be assigned a cross-section. . .	172
<b>Figure G.15:</b> The <i>Boundary Conditions</i> tab with all essential parts marked in red cycles and enumerated. . . . .	173
<b>Figure G.16:</b> Input example for external nodal loading specification. . . . .	173

<b>Figure G.17:</b> Input example for nodal support specification. . . . .	174
<b>Figure G.18:</b> The <i>Analysis Controls</i> tab with all essential parts marked in red cycles and enumerated. . . . .	177
<b>Figure G.19:</b> The <i>Analysis Controls</i> tab with the advanced settings enabled. . . . .	178
<b>Figure G.20:</b> The <i>Analysis Controls</i> tab with the advanced settings enabled. . . . .	178
<b>Figure G.21:</b> The <i>Solve Job</i> tab. . . . .	179
<b>Figure G.22:</b> The <i>Post-processing</i> tab and the <i>Deformation history</i> panel. . . . .	181
<b>Figure G.23:</b> The <i>Solve Job</i> tab and the <i>Deformation history</i> . . . . .	182
<b>Figure G.24:</b> Equilibrium path plot. . . . .	183
<b>Figure G.25:</b> Deformation history plot of the structure. . . . .	183
<b>Figure G.26:</b> Load problem using the input file. . . . .	184
<b>Figure G.27:</b> How NODE part is typeset in the ASCII file and how it is stored in a cell data structure by MATLAB. . . . .	186
<b>Figure G.28:</b> How ELEMENTS part is typeset in the ASCII file. . . . .	186
<b>Figure G.29:</b> How MATERIALS part is typeset in the ASCII file, with multiple materials defined. . . . .	187
<b>Figure G.30:</b> How SECTIONS part is typeset in the ASCII file, with multiple cross-sections defined. . . . .	188
<b>Figure G.31:</b> The ELEMENT-SECTION ASSIGNMENT input block. . . . .	189
<b>Figure G.32:</b> Defining load and displacement boundary conditions in the BOUNDARY CONDITIONS block for the ASCII input file. . . . .	189
<b>Figure G.33:</b> The ANALYSIS CONTROLS template block for the ASCII input file. .	190
<b>Figure G.34:</b> The equilibrium plot for the elastoplastic cantilever, using the GUI post-processor utility. . . . .	191
<b>Figure G.35:</b> The input file for a simple cantilever problem. . . . .	192
<b>Figure G.36:</b> The <b>main.m</b> file of the hybrid code. . . . .	194
<b>Figure G.37:</b> The geometry of the frame used as input. . . . .	194

<b>Figure G.38:</b> First portion of Matlab input file. . . . .	195
<b>Figure G.39:</b> Second portion of Matlab input file. . . . .	196
<b>Figure G.40:</b> Third portion of Matlab input file. . . . .	197
<b>Figure G.41:</b> Fourth and final portion of Matlab input file. . . . .	198
<b>Figure G.42:</b> The <b>snode</b> data structure. . . . .	200
<b>Figure G.43:</b> The <b>element</b> data structure. . . . .	201
<b>Figure G.44:</b> The <b>anlsData</b> data structure. . . . .	204

## List of Tables

<b>Table 2.1:</b>	Comparison of tip normalized displacements for different load levels for elastic response. . . . .	36
<b>Table 3.1:</b>	Results for Scenario 1, perfect plasticity. . . . .	64
<b>Table 3.2:</b>	Results for Scenario 2, perfect plasticity. . . . .	64
<b>Table 3.3:</b>	Results for Scenario 3, perfect plasticity. . . . .	64
<b>Table 3.4:</b>	Results for Scenario 4, five cycles, perfect plasticity. One cycle: A-B-C-A. .	64
<b>Table 3.5:</b>	Collapse moments and loads for clamped beam. . . . .	67
<b>Table 3.6:</b>	Material properties for Example 3.5.4. . . . .	70
<b>Table 3.7:</b>	Member geometric and material data. . . . .	72
<b>Table 3.8:</b>	Convergence in residual norm during steps 5, 15, 25 35. $\epsilon_{tol} = 10^{-10}$ . . .	73
<b>Table 3.9:</b>	Convergence in energy norm during steps 5, 15, 25, 35. $\epsilon_{tol} = 10^{-10}$ . . .	74
<b>Table 3.10:</b>	Vertical gravity loads at nodes. . . . .	75
<b>Table 4.1:</b>	Stationary solutions of Eq. (4.32). . . . .	101
<b>Table 4.2:</b>	Initial States for paths $\Gamma_1$ , $\Gamma_2$ and Target States reached. . . . .	104
<b>Table 5.1:</b>	Results for Example 1, $\epsilon_{tol} = 10^{-6}$ , $t_0 = 0$ , $t_{max} = 1$ . . . . .	129
<b>Table 5.2:</b>	Results for Example 2, $\epsilon_{tol} = 10^{-6}$ , $t_0 = 0$ , $t_{max} = 1.2$ . . . . .	130
<b>Table 5.3:</b>	Stationary points of Eq. (5.5.2). . . . .	130
<b>Table 5.4:</b>	Results for Example 3, $\epsilon_{tol} = 10^{-6}$ , $t_0 = 0$ , $t_{max} = 1$ , Normal Flow. . .	132
<b>Table 5.5:</b>	Results for Example 4, $\epsilon_{tol} = 10^{-6}$ , $t_0 = 0$ , $t_{max} = 1$ , $\alpha = 0.2$ . . . . .	134

## Nomenclature

$\vec{E}_i$	Spatial basis unit vectors, $i = 1, 2, 3$
$\vec{e}_i$	Material basis unit vectors, $i = 1, 2, 3$
$X_i$	Initial configuration coordinates, $i = 1, 2, 3$
$\vec{X}$	Position vector of a material point in the initial configuration
$x_i$	Current configuration coordinates, $i = 1, 2, 3$
$\vec{x}$	Position vector of a material point in the current configuration
$\vec{r}_o$	Position vector of a material point on the centerline in the initial configuration
$\vec{r}$	Position vector of a material point on the centerline in the current configuration
$\vec{u}$	Displacement vector of points on the centerline with respect to the initial configuration.
$\vec{t}$	Position vector of a material point on the local system of a cross-section.
$\mathbf{f}_c$	Forcing term, counterpart of $\vec{f}$ , expressed in the shifted basis $\vec{c}$
$w(s)$	Weight function associated with the weighted least squared predictor
$\mathbf{W}$	Diagonal matrix of weights of solutions points with respect to the current one
$\mathcal{B}$	Coefficient matrix associated with the weighted least squares predictor
$\vec{t}_z$	Unit tangent vector on the weight least squares curve
$\widetilde{\Delta s}$	Stabilized step-length associated with weighted least squares extrapolation
$D_s$	Approximation of segment on the solution curve between two converged points
$h_m$	Average mesh spacing in a given spatial direction
$d$	Spatial dimension of the problem
$N_{nod}$	Total number of (structural )nodes in the system
$\phi$	Angle between $\vec{E}_i$ and $\vec{e}_i$

$\mathbf{R}(\phi)$  Rotation matrix

$\kappa$	Curvature of the beam centerline.
$\epsilon$	Axial strain of the beam centerline
$\gamma$	Shear strain of the beam centerline
$\vec{s}$	Vector containing $\epsilon$ and $\gamma$ but not $\kappa$
$u$	Displacement component along $X_1$
$v$	Displacement component along $X_2$
$\vec{d}$	Displacement vector containing $u, v$ and $\phi$
$\vec{q}$	Strain vector containing $\epsilon, \gamma$ and $\kappa$
$f$	Generic objective function
$\vec{h}$	Vector of discretized constraints
$\mathcal{S}$	Constrained space for $\vec{x}$
$\mathcal{L}$	Lagrangian function associated with the NLP
$\vec{\lambda}$	Vector of Lagrange multipliers associated with equality constraints
$\Pi$	Total potential energy function
$U$	Strain energy function
$W$	Potential energy of external loads
$\mathcal{W}_s$	Cross-section strain energy density
$\vec{P}$	Vector of external loads
$\vec{F}_{sec}$	Cross-section stress resultants
$N$	Section axial force
$V$	Section shear force
$M$	Section internal moment
$E$	Young's modulus
$G$	Shear modulus
$I$	Cross-section moment of inertia
$A$	Cross-section area

$A_s$	Effective area of the cross-section
$\ell$	Element length
$n_{nod}$	Number of non-restrained nodes
$n_q$	Number of quadrature points in the element
$w_i$	Quadrature weight for the $i$ -th quadrature point
$\xi$	Normalized coordinate associated with $X_1$
$L_i$	Lagrange cardinal function
$\mathbf{T}$	Curvature mapping matrix
$\Theta$	Vandermonde matrix
$\vec{g}$	Internal field variable vector, associated with unknowns at quadrature points
$\Lambda$	Element orientation matrix
$\theta$	Angle of element centerline in the initial configuration with $\vec{E}_1$
$\mathcal{E}^e$	Energy contribution of element $e$ in the Lagrangian function
$\mathbf{H}$	Hessian matrix
$\vec{z}$	Vector of all unknown state variables of the system
$\mathbf{k}_s$	Section stiffness matrix
$\mathbf{K}$	Stiffness matrix
$\vec{\mathbf{F}}_{int}$	Internal nodal force vector
$\Delta s$	Incremental arc-length parameter
$\vec{\sigma}_f$	Vector of active stress components on a fiber
$\vec{\epsilon}_f$	Strain components corresponding to $\vec{\sigma}_f$
$\mathbf{N}_s$	Cross-section shape function matrix
$\varphi$	Shear strain distribution function along the cross-section height
$\mathbf{C}$	Material tangent modulus
$\vec{d}_{l,r}$	Element nodal displacement vector for left ( $l$ ) or right ( $r$ ) elememtn edge
$\vec{P}_{l,r}$	Element externally applied nodal force vector for left ( $l$ ) or right ( $r$ ) elememtn edge
$\boldsymbol{\sigma}$	Cauchy stress tensor

$\epsilon$	Strain tensor
$a$	Back stress tensor
$q$	Current uniaxial yield stress
$e^{pl}$	Equivalent plastic strain
$\lambda$	Plastic parameter
$\Phi$	Yield function
$\sigma_{eq}$	Equivalent stress associated with the appropriate yield criterion
$H_{kin}$	Kinematic hardening modulus
$H_{iso}$	Isotropic hardening modulus
$\mathbb{C}, \mathbf{C}$	Fourth order constitutive tensor and its matrix representation
$J_2$	Second invariant of the deviatoric stress tensor
$\sigma_y$	Yield stress of the virgin material
$\zeta$	Effective stress tensor
$\mathbf{w}^d$	Deviatoric part of a tensor $\mathbf{w}$
$\mathbf{C}_c^{pl}$	Consistent elastoplastic tangent modulus
$\vec{\mathbf{H}}$	Homotopy (vector) function
$t$	Pseudo-time or homotopy parameter
$C^k$	Space of functions with their first $k$ derivatives continuous
$\Gamma$	A (connected) component, path or loop, in $\vec{\mathbf{H}}^{-1}(\vec{0})$
$\Omega$	Feasible space for an NLP or PNLP program
$I_h$	Index set of equality constraints
$I_g$	Index set of inequality constraints
$\vec{\mu}$	Vector of Lagrange multipliers associated with inequality constraints
$\bar{I}_g$	Index set of active inequality constraints
$\mathcal{D}$	Subdivision of an interval
$\tau(\bar{I}_g)$	Cell associated with subdivision of $\Omega$
$W^q$	Subinterval associated with subdivision $\mathcal{D}$

$\vec{J}$	Jacobian of a vector function
$\mathcal{P}_m$	Space of polynomials up to degree $m$
$f$	Forcing term associated with the least squares moment equations
$\mathbf{A}$	Moment matrix associated with a least squares process
$\vec{p}$	Polynomial basis vector associated with the weighted least squares predictor
$\vec{c}$	Shifted polynomial basis vector associated with the weighted least squares predictor
$\Phi$	Transformation matrix between $\vec{p}$ and $\vec{c}$
$\mathbf{A}_c$	Moment matrix, counterpart of $\mathbf{A}$ , expressed in the shifted basis $\vec{c}$

## **Acronyms**

**AB2** Two-step Adams-Bashforth.

**AB3** Three-step Adams-Bashforth.

**DOF** Degree of Freedom.

**FEM** Finite Element Method.

**GUI** Graphical User Interface.

**IS** Initial State.

**IVP** Initial Value Problem.

**KKT** Karush-Kuhn-Tucker.

**LICQ** Linear Independence Constraint Qualification.

**NLP** Non-Linear Programming.

**NPNLP** Naturally Parameterized Non-Linear Programming.

**PC** Predictor-Corrector.

**PNLP** Parametric Non-Linear Programming.

**SCC** Strict Complementarity Condition.

**TPE** Total Potential Energy.

**TS** Target State.

**WLS** Weighted Least Squares.

**WLSE** Weighted Least Squares Extrapolation.

**WLSIT** Weighted Least Squares Implicit Tangent.

**WLST** Weighted Least Squares Tangent.

## Acknowledgements

I would like to express my sincere gratitude to Dr. Kostas Papakonstantinou, my academic advisor and chair of my doctoral committee, for his guidance, encouragement and valuable assistance throughout all these years. Without his guidance, this work would not have come into fruition and for that I am deeply grateful.

Special thanks are owed to the rest members of the dissertation committee, Dr. Gordon Warn, Dr. Pinley Chen, Dr. Tong Qiu, Dr. Reuben Kraft, and the former member Dr. Michael Hillman, for their participation, advice and suggestions in completing this work. I would also like to thank my friend and colleague Dr. Charalampos Andriotis for all his helpful suggestions and important contributions early on in this endeavor.

I cannot conclude this acknowledgement without expressing my deep gratitude to my parents, Claire and Michael, to whom this dissertation is dedicated. They have always been by my side with their encouragement, love and unwavering support, without which none of this would have been possible.

The material presented herein is based upon work supported by the U.S. National Science Foundation under Award No. 1634575. The support is greatly acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the U.S. National Science Foundation.

*To my parents*

# CHAPTER 1

## INTRODUCTION AND BACKGROUND

### 1.1 Literature Review

#### 1.1.1 Beam Element Technologies

Engineering applications involving structures undergoing large displacements, rotations and elastic or inelastic deformations are quite common in several fields and industries. As far as rod-like structures are concerned, computational extensions of the classical Euler-Bernoulli theory are required for the accurate prediction of the aforementioned effects, but also to account for shear flexibility considerations. In small displacement theory, the undeformed and deformed configurations are regarded as identical and, therefore, only one configuration acts as referential. Yet, when large displacements are considered, the description of beam kinematics and equilibrium have to take into account that the deformed geometry of the beam can be drastically different from the undeformed one. Within the Lagrangian framework, one chooses a body configuration to refer to, giving rise to the notion of total and updated Lagrangian formulations [1]. In the former, the initial (or undeformed) configuration is chosen as a reference, whereas in the latter, typically associated with path-dependent problems, one refers to the current (or deformed) one.

Several methods have been employed over the years to tackle geometrically nonlinear problems of beams within the context of finite elements. One approach is to introduce a local and rigid Cartesian coordinate frame associated with each element and rotating with it. Small displacement assumptions can then be applied locally and computed quantities are referred back to the global frame by the associated element transformation matrix. Such approaches, as typically encountered in the treatment of large displacements of beams and shells, are often termed co-rotational [2, 3, 4, 5], are associated with small strain assumptions and ordinarily require

fine discretization. Early application of such approaches are found in the works of Argyris et.al. [6] and Wempner [7]. For a survey on co-rotational methods the interested reader may refer to Felippa and Haugen [8].

Another approach, which we follow in this work, considers orthogonal coordinate frames attached to material points along the beam centerline. By choosing a reference configuration the task then is to trace the current position vectors of such frames. Since beam cross-sections are assigned to material points of the centerline, these frames completely describe the current beam configuration once the transformation of the frames from the current to the reference configuration is established. Formulations adhering to this geometric description have often been termed geometrically-exact and have been considered by Reissner [9], and further extended in the works of several authors, such as Simo [10], Simo and Vu-Quoc [11, 12], Cardona and Geradin [13], Ibrahimbegovic [14], Romero and Armero [15], among others. The geometrically-exact formulation is particularly attractive because the kinematic description is expressed directly in a global inertial frame for all elements, as opposed to local element-specific frames in the co-rotational method. For three-dimensional problems, the description of finite rotations requires special treatment (e.g., see [16, 17]) due to their non-commutative algebraic character. There are, however, methods that are able to tackle large rotation problems without considering nodal rotational variables (e.g., see Romero [18], Shabana and Yakoub [19]). For a comparison between co-rotational and geometrically-exact approaches, see Mathisen et al. [20]. More recent developments of geometrically-exact models in the context of finite element analysis include viscous damping extensions[21], incorporation of the Green-Lagrange tensor as strain measure [22], a mixed formulation based on complementary energy principles [23], and modeling of fiber-based materials[24], to name but a few. Marino [25] and Tasora et al. [26] also implemented geometrically-exact formulations in the framework of isogeometric analysis [27], while Salehi and Sideris[28], following Reissner's theory, developed a large-strain force-based formulation capable of tackling material softening.

Relevant approaches where strain measures are included as nodal variables are often termed

strain or deformation-based and, among others, have been investigated by Planinc et al. [29] for the plane case of a geometrically-exact beam, in order to properly account for the effect of local instabilities on the tangent stiffness matrix due to inelastic response. Extensions to two-dimensional dynamic cases [30] and to three-dimensional models [31] handling issues related to strain-objectivity have also been achieved. A similar formulation has also been presented by Saje [32], where the exact kinematic relations are incorporated using the Hu-Washizu variational principle and nodal unknowns comprise only rotational variables. Bayo et al. [33] introduced a penalty formulation in order to take into account kinematic and motion constraints. This approach avoids the use of additional Lagrange multiplier unknowns, yet the penalized constraints are not satisfied exactly and criteria for the proper adjustment of the penalty coefficients are difficult to establish.

### 1.1.2 Modelling Inelastic Behavior in Beams

Modelling the inelastic response of structural beams is of considerable interest in many fields of engineering practice. While accuracy of numerical approximations is retained even in nonlinear elastic simulations, this is not the case when inelastic response is expected. The localized and complicated nature of processes at the material level becomes more difficult to capture with numerical models developed at a higher abstraction level, such as beams. Since candidate locations for plastification in frame structures are usually known *a priori*, analytical models have been developed based on macroscopic, geometric and constitutive, assumptions that can account for the localized character of elastoplastic response. These phenomenological formulations are commonly known as concentrated or lumped plasticity models and have enjoyed widespread use, especially for reinforced concrete applications, while also offering low computational costs.

An alternative approach is based on discretization of the beam cross-section into a number of fibers or layers. This allows for the employment of constitutive laws directly at the material level. The element stiffness and internal nodal forces are then determined by integrating the elastoplastic rate equations over selected cross-sections in the element interior, which are encap-

sulated by the quadrature points. In addition, this enables the modelling of plastic zone spread over the element, thus allowing for a more faithful representation of the response. Elements formulated this way are called fibre or distributed plasticity beam elements[34, 35, 36, 37, 38] and have seen active development for almost 30 years. In the simplest case where only uniaxial constitutive law is considered at the level of fibers, these elements can adequately reproduce inelastic axial-flexural interactions in slender members.

While incorporation of shear effects into the fiber constitutive law is straightforward, reproducing an accurate macroscopic behavior by integrating a multiaxial law over the section fibers has proven intricate in general. One reason for this is the nonlocal effect of shear, which depends on structure-level characteristics, such loading and boundary conditions[39]. Additional difficulties are induced when modelling complex cross-section configurations comprised of composite materials, such as reinforced concrete (RC) sections. Restricting ourselves to a review of methodologies developed in the context of fibre elements, early studies[40, 41] modelled shear-axial-flexure interaction in planar beams by iteratively enforcing static equilibrium for each fiber between two controlling sections. More recently this was extended to 3D beams[42]. Other works introduce a stress-resultant model for shear and retain an independent representation axial-flexure coupling through fiber discretization[43, 44, 45]. Taylor et al.[46] and more recently Lyritsakis et al.[47] considered uniaxial laws on each fiber for shear and axial stresses separately, thereby ignoring shear-axial coupling. A more physically rigorous approach is adopted by a number of authors[48, 49, 50, 51, 52], whereby a multiaxial model is utilized at the fiber level. However, this requires additional corrective iterations per fiber[53, 54, 55] during a plastic step in order to enforce the zero transverse normal stress condition,  $\sigma_{yy} = \sigma_{zz} = 0$ , thereby introducing additional overhead.

### 1.1.3 Local and Global Solution Methods

When dealing with problems in optimization, some important questions that arise are concerned with the objective function properties and whether a particular algorithm can converge to a sta-

tionary point, given an initial guess. In a similar fashion, the problem of solving systems of nonlinear algebraic equations, which is encountered in virtually all branches of engineering practice, poses similar challenges. For problems where an underlying variational structure exists, the two cases above can be viewed as equivalent. For mechanics applications in particular, the solution of the equilibrium equations implies the stationarity of the total potential energy of the system. For such applications, it is often the case that more solutions exist for a particular external input, whether it is an applied load or a design parameter. In such cases, it is important for the underlying mathematical formulation to provide these tools that would allow the analysis to reveal all such solutions.

One of the most common solution methods used for such problems is the Newton-Raphson method[56], which converges with quadratic rates if the initial guess is in the proximity of the solution. In this sense, the method is said to be local. It is also well known that it fails at singular points because the Jacobian becomes non-invertible[57]. A family of algorithms known as quasi-Newton methods[58, 59, 60], which are variants of the Newton-Raphson method, attempt to bypass the singularity issue by approximating the Jacobian according to certain criteria while also offering substantial advantages as far as computational cost is concerned. Such methods have been utilized extensively in computational mechanics literature[61, 62], with the BFGS[63] variant being the most widely used in implementations.

To overcome the shortcomings of Newton-type iterative schemes, globally convergent methods have been an active field of research for many decades[64]. These are often termed continuation, imbedding or homotopy methods[65] and aim to converge to one or more solutions, irrespectively of how good the initial guess is, by tracing a smooth set of points(or path), with both the solution and the initial guess being members of it. Modern homotopy continuation literature within the applied mathematics community is quite rich[66, 67, 68, 69, 70, 71, 72, 73, 74, 75] and its origins can be traced back to the works of Davidenko[76, 77], Klopfenstein[78], Haselgrove[79] and Yakovlev[80], among others. The main idea is to imbed the problem in a higher dimensional space using a homotopy function and, if certain regularity conditions hold,

the implicit function theorem guarantees that a smooth parameterizable path exists that connects an arbitrary initial point to at least one solution. By construction, the Jacobian in the augmented space is always of full rank and turning points now pose no difficulty during the numerical treatment of the problem.

In engineering applications, and more specifically in nonlinear elasticity, the various so-called incremental methods which were developed independently to tackle problems of elastic stability and to trace the load-displacement curve past critical points[81, 82, 83, 84, 85, 86, 87] are closely related to the homotopy continuation framework[74]. Dynamic Relaxation, which is another family of methods that has been applied to the solution of nonlinear quasi-static problems[88, 89, 90, 91, 92], can also be regarded as a numerical application of an underlying homotopy imbedding[93]. Explicit use of the homotopy framework in structural mechanics investigations can be found in Watson et al[94, 95], Rheinboldt[96]. Other disciplines that have employed numerical continuation in applications include optimization[97, 70, 98, 99, 100], chemical engineering[101, 102, 103], transfer problems in electrical circuits[104] and, more recently, problems with variational structure and energy minimization[105, 106, 107], structural mechanics[108, 109, 110, 111, 112] and topology optimization[113], constitutive modelling[114], fluid mechanics[115, 116], economics & operations research[117, 118, 119] and numerical quadrature investigations for isogeometric analysis[120, 121], among others. A survey on the application of homotopy continuation to engineering problems is provided by Seydel and Hlavacek[122].

## 1.2 Research Objectives

The objective of this research is to develop a high-performance, geometrically-exact, hybrid beam element that is formulated in terms of nonlinear programming principles, whereby the total potential energy of the structure is treated as the objective function. The formulation will offer the capability to model a structural member with just one element in both linear and non-linear response spectrum. Since the element formulation is recast using optimization concepts,

we aim to also develop a parametric nonlinear programming framework for the hybrid beam element, which would provide the necessary tools for general purpose parametric analyses with global convergence characteristics. In addition, we aim to develop a fast and robust prediction scheme to be used during the numerical solution of the parametric nonlinear program.

More specifically, this study proposes:

- The development of a beam element that utilizes exact kinematics and is based on nonlinear programming principles. The element would be able to capture arbitrarily large displacements and rotations, as well as curvature localization during inelastic response, with just one element per member. This is a unique capability which, to our knowledge, no beam implementation possesses in commercial and academic software. In addition, the element incorporates shear deformation effects without suffering from locking behavior. The element hybridization is carried out by incorporating the exact kinematic equations in the minimization statement of the potential energy by introducing a set of Lagrange multipliers.
- The formulation of a fast and robust return-mapping algorithm for the numerical integration of fiber-discretized beam cross-sections. The proposed algorithm is an essential part of the section state determination procedure as it provides axial-shear-flexure interaction capabilities and is tailored for the planar hybrid beam element. Because it does not incorporate non-active components of the stress tensor, there is no need to enforce the zero transverse stress constraint at the fiber level, thereby, avoids a nested local Newton iteration. In addition, it requires less memory storage for the state variables. Thus, the stress-update algorithm is considerably faster than current formulations which are based on fibre elements and multiaxial material constitutive models.
- The development of a parametric nonlinear programming framework for the hybrid beam element that will provide the capability to perform analyses on a structural system that are of general consideration. Such analyses include but are not limited to: static response

to load history, design and parameter optimization, contact, global sensitivity analysis. It essentially provides an application interface for the study of parametric dependence of the structural system configuration and how these states change under large variations of the designated parameter. In addition, it establishes the conditions that ensure global convergence properties and the numerical tools to implement such capabilities. By utilizing principles from homotopy theory, we highlight what specific regularity conditions need to hold so that a unique and smooth path exists which connects the initial configuration state, which is considered known, with at least one equilibrium state of a specified potential.

- Incorporation of a fast and reliable prediction scheme in the global solution algorithm used for solving the nonlinear system of equations. Numerical continuation is utilized in approximating stationary points (or, equivalently, solutions) of a system if the initial guess is not in their proximity. The class of algorithms most commonly used for this task are the so-called predictor-corrector algorithms and are used to numerically track paths generated by homotopy or parametric programming imbeddings. Prediction is a crucial and, usually, computationally costly phase, as it affects the quality of the first estimate of the solution we seek. A predictor based on a weighted least squares process is proposed, whereby the next solution estimate is determined from a locally fitted curve. It bypasses the need to rely on second derivative information, as is usually the case, and avoids the penalizing constraints imposed by interpolatory prediction schemes. In addition, it provides reliable initial estimates of sought solutions at a fraction of computational cost compared to most other options. Lastly, if an iterative solver is used for the correction phase, such as MINRES or the more general GMRES, then the numerical continuation can, in principle, be further accelerated.

### 1.3 Research Scope

The scope of this research, based on the aforementioned objectives, is listed below:

- Develop a nonlinear programming framework for a beam element, using the total potential energy as the objective function. The element kinematics are determined from the strain-displacement equations, which are regarded as constraints acting on the optimization program. The Simo-Reissner or geometrically-exact kinematic relations are used in this study, which do not introduce any approximation in the magnitude of nodal displacements or rotations. Shear deformation is fully accounted for but cross-section warping effects are ignored. The formulation pertains to quasi-static problems.
- Develop a multiaxial constitutive model for the stress-update procedure at the cross-section fiber level. The formulation assumes infinitesimal strains, rate-independent, associative elastoplasticity. The von Mises yield criterion is adopted in the present study, where linear kinematic and general isotropic hardening are incorporated in the derivations. The numerical integration of rate constitutive equations of elastoplasticity is carried out using the fully implicit backward Euler method. The consistent tangent modulus for the proposed model is also derived.
- The parametric nonlinear programming framework assumes the total potential energy of the structure and the kinematic (or other) constraints are sufficiently smooth functions of the state variables. In addition, certain regularity conditions, which are explained in detail in the relevant sections, are assumed that guarantee no bifurcating paths occur during the solution process. Furthermore, in the presence of inequality constraints, we assume that the linear independence constraint qualification and the strict complementarity condition hold for the whole domain, with the exception of maybe a finite number of points. This allows us to establish the existence of piecewise smooth paths that lead to stationary points and a numerical solution algorithm to track such paths is developed. The derivation of the framework heavily relies on homotopy continuation principles and optimization in order to establish criteria for a truly global method.
- The development of the weighted least squares predictor assumes an underlying polyno-

mial basis and that a number of previously converged solution points is stored in memory. The proposed algorithm is general and can be applied for arbitrary, general problems where the task is to solve systems of nonlinear equations by numerical continuation using a predictor-corrector approach.

## 1.4 Organization of the Manuscript

The remainder of the manuscript is organized as follows:

Chapter 2 introduces the formulation of the hybrid beam element. The exact kinematics utilized are briefly outlined, followed by the constrained, nonlinear programming statement for the element. Next, we present detailed implementation details, along with a block elimination technique for the Hessian that allows for solving only for nodal displacement state variables. The chapter concludes with a set of numerical examples where we demonstrate the capabilities of the proposed element.

Chapter 3 presents the multiaxial elastoplastic constitutive model adjusted for the stress state of a beam fiber. The chapter begins with the general form of the rate constitutive equations for elastoplasticity, as expressed for the three-dimensional problem. It then proceeds with developing a mapping procedure from that space to the constrained fiber stress space. The equivalent set of rate equations is presented here with the assumption of a von Mises yield criterion. Then, implementation details pertaining to the application of the fully implicit Euler scheme are detailed and the elastic predictor-plastic correct return mapping algorithm is outlined for the constrained stress-update problem. The Chapter closes with a set of numerical examples that test the accuracy of both the algorithm, as well as its performance when used on a number of elastoplastic simulation examples.

Chapter 4 is concerned with the development of the parametric nonlinear programming framework for the hybrid beam element. The Chapter starts with a brief but detailed outline of homotopy continuation theory as applied to solving systems of equations, as well as its extension treating parametric and global optimization problems. The next section introduces

a naturally parameterized nonlinear programming framework suited for structural mechanics applications that use the proposed beam element. The conditions that guarantee global convergence are stated and an appropriate predictor-corrector continuation algorithm is proposed that can track piecewise smooth paths. The Chapter concludes with a set of examples involving optimization of a function, a study of global sensitivity of a structural system to changes in axial stiffness of a member, as well as a frictionless contact problem using a nonlinear elastic constitutive law.

Chapter 5 develops the weighted least squares predictor for use in the predictor-corrector algorithm proposed in the previous chapter. The Chapter first provides a summary on the use of numerical continuation and, specifically, the role of the prediction phase in it. After that the main idea of a locally weighted least squares fitting using a user-defined number of previously converged solution points is presented, along with various additional features provided by this prediction scheme, such as the weighting function, stabilization and the different predictor variants that arise from the proposed algorithm. The following section is concerned with implementation details and, finally, the Chapter concludes with a set of numerical examples that test the performance of the proposed predictor with i) the Euler predictor and ii) two-step and three-step Adams-Bashforth predictors.

Lastly, Chapter 6 contains a summary of the present research work, a discussion of its findings, advantages, and its limitations and suggestions regarding potential future research directions.

Unless stated otherwise, boldface lowercase letters indicate second order tensor. Wherever the same letters appear with an overhead arrow, this indicates the vector notation for the tensor represented with that letter. In addition, any boldface letter, lowercase or uppercase, with an overhead arrow indicates a vector object. In contrast, matrices are represented exclusively by uppercase, non-italic or calligraphic, boldface letters, whereas normal font letters indicate scalar quantities. For more details regarding the convention used, see Appendix A.

## CHAPTER 2

### HYBRID BEAM ELEMENT FORMULATION

#### 2.1 Introduction

In this chapter we present the formulation of a novel hybrid beam element which is based on NLP principles. The kinematic assumptions adopted fall under the category of geometrically exact or Simo-Reissner beam theory[9, 10, 11, 12], whereby no simplifying approximation is made with respect to the strain-displacement equations. This allows for capturing arbitrarily large displacements and rotations, as well as accounting for the effect of shear deformation at the section level.

As opposed to deriving the system equations from the Galerkin form,, we recast the problem in an NLP framework by utilizing the underlying variational structure. The total potential energy (TPE) functional is augmented with all relevant conditions that enforce the exact kinematics, by introducing a set of Lagrange multipliers that act as conjugate force quantities. The resulting modified functional is then approximated by employing a Gauss-Legendre quadrature rule, which yields the objective function to be minimized. With this particular approach, the primary variables in the element interior contributing to the elastic strain energy are the generalized strain measures of the centroid, which are the unknown quantities at the quadrature points. Displacement measures at the edge nodes of the element, namely, the translations along coordinate axes and the rotation of cross sections, are only associated with the external work. Kinematic consistency between the rotational measures of displacement and strains is enforced by using a Lagrange interpolation scheme for the curvature field, similar to the one used by Neuenhofer and Filippou [123] and Schulz & Filippou [124] for force-based elements. In this work, the points used for the interpolation coincide with the integration points of the quadrature rule used for approximating the energy functional. To avoid ill-conditioning issues of the lin-

earized operator, we also outline a block-elimination procedure for the linear system involving the Hessian matrix so that the reduced system includes only displacement components. The corresponding stiffness operator is well-conditioned, sparse, banded and symmetric. This renders standard FEM routines from existing codes reusable and, in conjunction with the capability for accuracy even with crude discretization, provides an attractive formulation for fast and accurate computation. Additionally, accuracy and locking free performance are guaranteed with just one element per structural member, even in the presence of arbitrarily large displacements and rotations. Accordingly, the element can also capture high curvature gradients due to plastic hinge formation, in the case of inelastic analysis.

This chapter starts with a brief outline of the beam geometric description while adhering to the derivations by Cardona & Geradin[125], followed by the hybrid NLP formulation for the element. The third section is devoted to implementation details, where special attention is given to the block elimination technique used for the reduction of the linear system involving the Hessian matrix. Finally, the chapter concludes with a set of numerical examples where the focus is the Hybrid NLP element performance with respect to coarse mesh discretizations in nonlinear analysis.

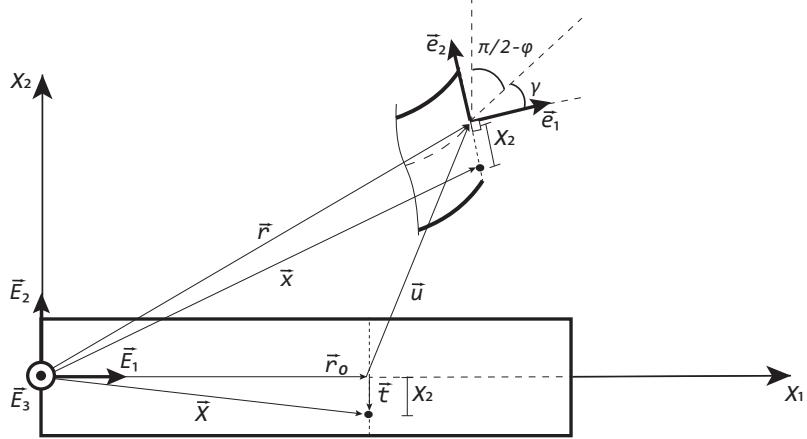
## 2.2 Geometrically-Exact Model

### 2.2.1 Beam Kinematics

Let  $\{\mathbf{E}\}$  be a fixed orthonormal coordinate frame with unit vectors  $\{\vec{\mathbf{E}}_1, \vec{\mathbf{E}}_2, \vec{\mathbf{E}}_3\}$  along the axes  $X_1, X_2, X_3$ . For an initially straight beam of length  $\ell$  with its centerline coinciding with  $X_1$ , the undeformed configuration is completely described by:

$$\vec{\mathbf{X}}(X_1, X_2) = X_1 \vec{\mathbf{E}}_1 + X_2 \vec{\mathbf{E}}_2 = \vec{\mathbf{r}}_o(X_1) + \vec{\mathbf{t}}(X_2) \quad (2.1)$$

where  $\vec{\mathbf{r}}_o$  traces the centerline in the reference configuration and  $\vec{\mathbf{t}}$  locates a material point on the cross section. Assuming a rectangular cross-section of height  $h$ , without loss of generality,



**Figure 2.1.** Undeformed and deformed configurations of the beam.

we have that  $X_1 \in [0, \ell]$  and  $X_2 \in [-\frac{h}{2}, \frac{h}{2}]$ . We should note here that  $X_3$  is omitted as it does not explicitly come into the expressions. We nevertheless hold on to the  $\mathbb{R}^3$  vector formalism to maintain consistency with the representation of rotation as a linear operator. After the application of a displacement field  $\vec{u}(X_1)$ , the deformed configuration is given by vector  $\vec{x}$  such that:

$$\vec{r} = \vec{r}_o + \vec{u} \quad (2.2)$$

$$\vec{x} = \vec{r} + X_2 \vec{e}_2 \quad (2.3)$$

where  $\{\mathbf{e}\}$  is a local coordinate system attached at cross-sections that completely describes their orientation. For the unit vectors of the local base  $\vec{e}_1$ ,  $\vec{e}_2$ ,  $\vec{e}_3$ , we let  $\vec{e}_1$  be normal to the cross-section, but not necessarily tangent to the deformed centerline, and  $\vec{e}_2$  (and  $\vec{e}_3$ ) coincide with the cross-section principal axes of inertia, as shown in Fig. 2.1.

If  $\mathbf{R}$  is the rotation operator that rotates  $\{\mathbf{E}\}$  to  $\{\mathbf{e}\}$ , then:

$$\vec{e}_i = \mathbf{R} \vec{E}_i, \quad i = 1, 2, 3 \quad (2.4)$$

For the plane case, the component form of  $\mathbf{R}$  reduces to:

$$\mathbf{R} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Then, using Eq. (2.4), Eq. (2.3) becomes:

$$\vec{x} = \vec{r} + \mathbf{R}\vec{t} \quad (2.6)$$

For Eqs. (2.3),(2.6) to hold in this manner, we tacitly assume that cross-sections remain rigid during deformation, that is, the position of a material point on a cross-section with respect to the centroid does not change.

### 2.2.2 Strain Measures, Strain-Displacement Relations

Within the context of geometrically-exact formulations, it is common to adopt as a strain measure  $\vec{s}$  the difference of the position vector gradients,  $\vec{r}$  and  $\vec{r}_o$ , with respect to the local frame (e.g. see [10, 125, 126]). Denoting derivatives with respect to  $X_1$  by  $(\cdot)'$ , we get:

$$\vec{s} = \mathbf{R}^T \vec{r}' - \vec{r}'_o \quad (2.7)$$

whereas the cross-section curvature is given as:

$$\kappa = \phi' \quad (2.8)$$

The axial and shear strain of the centroid,  $\epsilon$  and  $\gamma$  respectively, can then be retained as follows:

$$\epsilon = \vec{e}_1^T \vec{s} \quad (2.9)$$

$$\gamma = \vec{e}_2^T \vec{s} \quad (2.10)$$

where  $\vec{s} = \begin{bmatrix} \epsilon & \gamma & 0 \end{bmatrix}^T$ , with the third component, representing the shear strain along the out-of-plane axis, being by assumption zero.

Using Eq. (2.2) and performing the differentiations in Eq. (2.7) with respect to  $X_1$ , we get:

$$\vec{s} = \mathbf{R}^T [\vec{u}' + \vec{E}_1] - \vec{E}_1 \quad (2.11)$$

Deriving the exact differential strain-displacement relations as presented in Reissner<sup>1</sup> is straightforward if we solve Eq. (2.11) for  $\vec{u}'$ :

$$\vec{u}' = \mathbf{R}\vec{s} + \vec{e}_1 - \vec{E}_1 \quad (2.12)$$

If we write  $\vec{u} = \begin{bmatrix} u & v & 0 \end{bmatrix}^T$ , then the explicit expressions for  $u', v'$  along with Eq. (2.8) are:

$$u' = \vec{E}_1^T \vec{u}' = (1 + \epsilon) \cos \phi - \gamma \sin \phi - 1 \quad (2.13a)$$

$$v' = \vec{E}_2^T \vec{u}' = (1 + \epsilon) \sin \phi + \gamma \cos \phi \quad (2.13b)$$

$$\phi' = \kappa \quad (2.13c)$$

By introducing vectors  $\vec{d}$  and  $\vec{q}$ , such that  $\vec{d} = \begin{bmatrix} u & v & \phi \end{bmatrix}^T$  and  $\vec{q} = \begin{bmatrix} \epsilon & \gamma & \kappa \end{bmatrix}^T$ , we can maintain the structure of Eq. (2.12) and express all fields involved in matrix notation, as follows:

$$\vec{d}' = \mathbf{R}\vec{q} + \vec{e}_1 - \vec{E}_1 \quad (2.14)$$

Strain-displacement equations as represented in Eq. (2.14) will be subsequently utilized in the formulation of the beam model.

---

<sup>1</sup>See equations (15) in [9]

## 2.3 Nonlinear Programming Formulation

The hybrid beam element discretization is presented by utilizing the underlying variational structure of the static problem. The minimizing principle for the TPE is recast here in an NLP formulation and the strain-displacement equations in (2.14) are incorporated, representing the equality constraints of the problem.

### 2.3.1 NLP Framework

The NLP problem adopted herein is of the following form:

$$\begin{aligned} & \text{minimize} && f(\vec{x}) \\ & \text{subject to} && \vec{h}(\vec{x}) = \vec{0} \\ & && \vec{x} \in \mathcal{S} \end{aligned} \tag{2.15}$$

The objective function  $f(\vec{x})$  represents the total potential energy of the structure, whereas vector  $\vec{h}(\vec{x})$  includes the active constraints of the program. The corresponding Lagrangian is:

$$\mathcal{L}(\vec{x}, \vec{\lambda}) = f(\vec{x}) + \vec{\lambda}^T \vec{h}(\vec{x}) \tag{2.16}$$

where  $\vec{\lambda}$  are the Lagrange multipliers. The first-order optimality conditions are given from the following expression:

$$\nabla \mathcal{L} = \vec{0} \tag{2.17}$$

This is the standard form of a nonlinear constrained minimization program with equality constraints, as encountered in the literature (e.g. [127]).

### 2.3.2 Total Potential Energy

We consider the following decomposition of the total potential energy of a beam:

$$\Pi = U - W \quad (2.18)$$

where  $U$  and  $W$  are the stored strain energy and potential energy associated with external loading respectively. For the case of concentrated external loads, Eq. (2.18) can be expressed as:

$$\Pi = \int_0^\ell \mathcal{W}_S(X_1, \vec{q}) dX_1 - \sum_{i=1}^2 \vec{P}_i^T \vec{u}_i \quad (2.19)$$

where  $\mathcal{W}_S$  is the strain energy density of a cross section at  $X_1$  and  $\vec{q}$  denotes the neutral axis strain vector,  $\vec{P}_i, \vec{u}_i$  the element nodal force and displacement vectors, respectively. It can be proven[128] that for strain hardening materials within the premises of small strain elastoplasticity, the exact solution to the problem renders  $\Pi$  an absolute minimum. Equivalently, the exact solution solves the program in Eq. (2.15) for  $f = \Pi$  with  $\vec{h}$  acting as constraints. Assuming a stable work-hardening material, the corresponding Euler-Lagrange equations yield the constitutive equations of elastoplasticity (e.g. see discussion in Simo and Hughes [129]). As it will be shown in subsequent sections, since the constitutive update is carried out locally at the section fiber level.

Section stress resultants are then given by the gradient of  $\mathcal{W}_S$ :

$$\vec{F}_{sec} = \nabla_{\mathbf{q}} \mathcal{W}_S \quad (2.20)$$

with  $\vec{F}_{sec} = \begin{bmatrix} N & V & M \end{bmatrix}^T$ . In the case of linear elasticity, the energy density is given by

$\mathcal{W}_S = \frac{1}{2}[EA\epsilon^2 + GA_s\gamma^2 + EI\kappa^2]$  and the section resultants are:

$$\vec{\mathbf{F}}_{sec} = \begin{bmatrix} EA\epsilon \\ GA_s\gamma \\ EI\kappa \end{bmatrix}$$

where  $A_s$  is the effective shear area of the cross-section. Numerical approximation of Eq. (2.19) by an appropriate quadrature yields:

$$\Pi(\vec{\mathbf{q}}_1, \dots, \vec{\mathbf{q}}_n, \vec{\mathbf{u}}_1, \dots, \vec{\mathbf{u}}_N) = \sum_{i=1}^{n_q} w_i \mathcal{W}_{S,i} - \sum_{i=1}^2 \vec{\mathbf{P}}_i^T \vec{\mathbf{u}}_i \quad (2.21)$$

with  $w_i$  and  $n_q$  being the weights and the number of integration points respectively, and  $\mathcal{W}_{S,i} = \mathcal{W}_S(\vec{\mathbf{q}}_i)$ . Eq. (2.21) serves as the objective function of the NLP of Eq. (2.15).

### 2.3.3 Strain-Displacement Constraints

The constrained conditions are derived by applying the same quadrature rule on the integral form of the strain-displacement equations (2.14):

$$\vec{\mathbf{d}}(\ell) - \vec{\mathbf{d}}(0) = \int_0^\ell \vec{\mathbf{d}}' dX_1 = \int_0^\ell \mathbf{R} \vec{\mathbf{q}} + \vec{\mathbf{e}}_1 - \vec{\mathbf{E}}_1 dX_1 \quad (2.22)$$

Using Eq. (2.4), numerical approximation of the right-hand side integral leads to the element specific constraints:

$$\vec{\mathbf{h}}^A = \vec{\mathbf{d}}(\ell) - \vec{\mathbf{d}}(0) - \sum_{i=1}^{n_q} w_i \mathbf{R}_i (\vec{\mathbf{q}}_i + \vec{\mathbf{E}}_1) + \ell \vec{\mathbf{E}}_1 = \vec{\mathbf{0}} \quad (2.23)$$

The component form of Eq. (2.23) can be given as:

$$\vec{h}^A = \begin{bmatrix} u(\ell) - u(0) - \sum_{i=1}^{n_q} w_i \left[ (\epsilon_i + 1) \cos \phi_i - \gamma_i \sin \phi_i \right] + \ell \\ v(\ell) - v(0) - \sum_{i=1}^{n_q} w_i \left[ (\epsilon_i + 1) \sin \phi_i + \gamma_i \cos \phi_i \right] \\ \phi(\ell) - \phi(0) - \sum_{i=1}^{n_q} w_i \kappa_i \end{bmatrix} \quad (2.24)$$

Although the strain fields appear explicitly only in the weighted evaluation points of the quadrature, the rotational field  $\phi$  is involved in both the integration points and the element edge nodes. As such, we introduce a Lagrange interpolation scheme for the curvature field in the same fashion as in [130]:

$$\kappa(\xi) = \sum_{i=1}^{n_q} L_i(\xi) \kappa_i \quad (2.25)$$

where  $L_i$  are the Lagrange cardinal functions:

$$L_i(\xi) = \frac{\prod_{j=1, j \neq i}^{n_q} (\xi - \xi_j)}{\prod_{j=1, j \neq i}^{n_q} (\xi_i - \xi_j)}, \quad \xi = \frac{X_1}{\ell}$$

Substitution of Eq. (2.25) to Eq. (2.13c) yields the expression for rotations  $\phi_i$ :

$$\phi_i - \phi(0) = \sum_{j=1}^{n_q} \left( \int_0^{\xi_i} L_j(x) dx \right) \kappa_j = \sum_{i=1}^{n_q} T_{ij} \kappa_j \quad (2.26)$$

with derivation of  $T_{ij}$  illustrated in Fig. 2.2. In matrix form, the above equation can be directly restated as a linear equality constraint set as:

$$\vec{h}^B = \vec{\phi} - \phi(0) \vec{1} - \mathbf{T} \vec{\kappa} = \vec{0} \quad (2.27)$$

where

$$\vec{\phi} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{n_q} \end{bmatrix}^T, \vec{1} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^{n_q}$$

$$\mathbf{T} = L \begin{bmatrix} \xi_1 & \frac{\xi_1^2}{2} & \dots & \frac{\xi_1^{n_q}}{n_q} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{n_q} & \frac{\xi_{n_q}^2}{2} & \dots & \frac{\xi_{n_q}^{n_q}}{n_q} \end{bmatrix} \Theta^{-1}, \Theta = \begin{bmatrix} 1 & \xi_1 & \xi_1^2 & \dots & \xi_1^{n_q-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & \xi_{n_q} & \xi_{n_q}^2 & \dots & \xi_{n_q}^{n_q-1} \end{bmatrix}$$

We can then collect all active constraints, Eqs. (2.23),(2.27), in one vector  $\vec{h}$ :

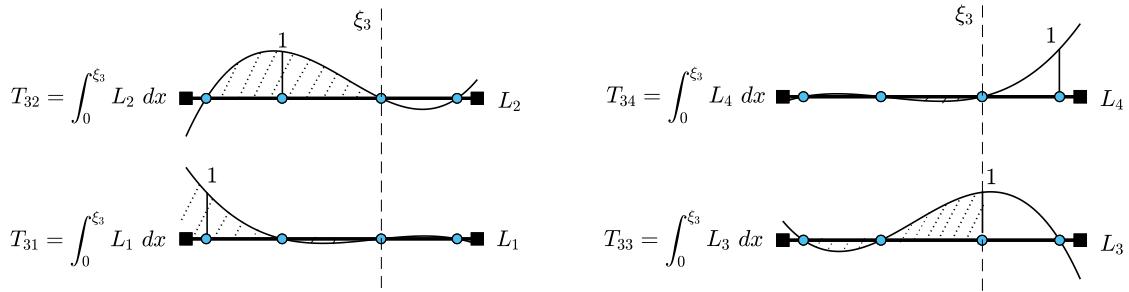
$$\vec{h} = \begin{bmatrix} \vec{h}^A \\ \vec{h}^B \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \end{bmatrix} \quad (2.28)$$

containing all constraints pertaining to one element. While vector  $\vec{h}^A$  of strain-displacement constraints will always contain three components for each element, the number of components in vector  $\vec{h}^B$  will depend on the chosen quadrature rule.

### 2.3.4 Lagrangian Function

In order to derive the Lagrangian function in the form of Eq. (2.16), we introduce a vector  $\vec{\lambda}$  of additional Lagrange multiplier variables and augment Eq. (2.21) with Eq. (2.28):

$$\mathcal{L} = f + \vec{\lambda}^T \vec{h} \quad (2.29)$$



**Figure 2.2.** Integration of curvature shape functions.

A more convenient form for the Lagrangian function can also be achieved by expanding  $f$  into its constituents, namely, the element stored energy and the potential energy:

$$\mathcal{L} = \sum_{e=1}^{n_{el}} U^e - W + \vec{\lambda}^T \vec{h} \quad (2.30)$$

where:

$$U^e = \sum_{i=1}^{n_q} w_i \mathcal{W}_{S,i}^e \quad , \quad W = \sum_{i=1}^{N_{nod}} \vec{P}_i^T \vec{u}_i$$

where  $N_{nod}$  is the total number of nodes in the structure. Thereby, additional elements can be simply incorporated by adding their stored energy in the corresponding sum, and considering element constraints by augmenting vectors  $\vec{\lambda}, \vec{h}$ , after transforming them to the global system. Potential function  $W$  is “global” in the sense that external loads and the corresponding work-conjugate displacement degrees of freedom are directly added to the expression and are not influenced by adding nodal contributions from adjacent elements. As a result, no additional connectivity constraints at the element interfaces are needed, and Eq. (2.30) is thus a global function for the whole structure.

## 2.4 Implementation

The set of vectors associated with internal field variables,  $\{\vec{y}_i\}_{i=1}^n$ , is defined as  $\vec{y}_i = \begin{bmatrix} \vec{q}_i & \phi_i \end{bmatrix}^T = \begin{bmatrix} \epsilon_i & \gamma_i & \kappa_i & \phi_i \end{bmatrix}^T$ . Moreover, and according to Fig. 2.3, we designate the local element edge degrees of freedom in vector form as  $\hat{d}_l = \vec{d}(0)$  and  $\hat{d}_r = \vec{d}(\ell)$ . The element displacement vector is thus  $\hat{d} = \begin{bmatrix} \hat{d}_l^T & \hat{d}_r^T \end{bmatrix}^T$ . The corresponding global displacement vector  $\vec{d}_g$  is related to the local vector  $\hat{d}$  via the typical linear transformation  $\Lambda$ :

$$\hat{\vec{d}} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \Lambda \end{bmatrix} \vec{d}_g \quad , \quad \Lambda = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.31)$$

The element constraints of Eqs. (2.23),(2.27) transformed in the global system become:

$$\vec{h}_g^A = \mathbf{V}_1 \vec{d}_g - \Lambda^T \left[ \sum_{i=1}^{n_q} w_i \mathbf{R}_i (\vec{q}_i + \vec{E}_1) - \ell \vec{E}_1 \right] \quad (2.32a)$$

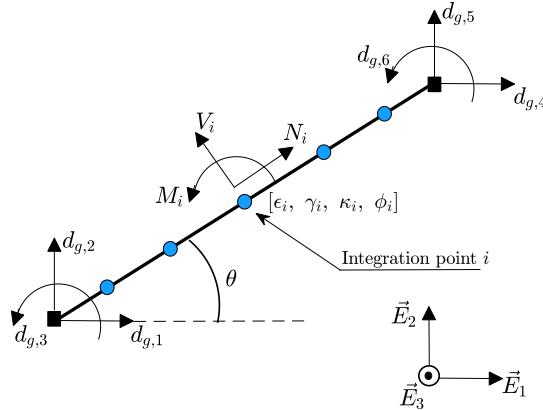
$$\vec{h}_g^B = \vec{\phi} - \mathbf{V}_2 \vec{d}_g - \mathbf{T} \vec{\kappa} \quad (2.32b)$$

where:

$$\mathbf{V}_1 = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \end{bmatrix} \quad , \quad \mathbf{V}_2 = \begin{bmatrix} \vec{0} & \vec{0} & \vec{1} & \vec{0} & \vec{0} & \vec{0} \end{bmatrix}$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad \vec{0} \in \mathbb{R}^{n_q}$$

In the equation above the superscript denoting an element is omitted for clarity. The element



**Figure 2.3.** Typical element representing one member.

contribution to the Lagrangian of the assemblage, denoted  $\mathcal{E}^e$ , is then:

$$\mathcal{E}^e(\vec{\mathbf{y}}_1, \dots, \vec{\mathbf{y}}_{n_q}, \vec{\mathbf{d}}_g, \vec{\boldsymbol{\lambda}}^e) = U^e + (\vec{\boldsymbol{\lambda}}^e)^T \vec{\mathbf{h}}_g^e \quad (2.33)$$

Notice that the external potential function  $W$  is not included in Eq. (2.33) as it is added directly via the work done by external forces in the global degrees of freedom and not from element contributions. The element Lagrange multipliers  $\vec{\boldsymbol{\lambda}}^e$  are force conjugate measures at the corresponding degrees of freedom and  $U^e$  is the sum total of cross-section strain energies. The Lagrangian of the whole system is accordingly given by:

$$\mathcal{L}(\vec{\mathbf{y}}, \vec{\mathbf{d}}_g, \vec{\boldsymbol{\lambda}}) = U - W + \vec{\boldsymbol{\lambda}}^T \vec{\mathbf{h}}_g \quad (2.34)$$

which is a restatement of Eq. (2.30) in the global coordinate system and  $\vec{\mathbf{y}} = \begin{bmatrix} \vec{\mathbf{y}}_1^T & \vec{\mathbf{y}}_2^T & \cdots & \vec{\mathbf{y}}_m^T \end{bmatrix}^T$ , with  $m$  being the total number of quadrature points in the structure.

#### 2.4.1 Equilibrium

In what follows, it is assumed we are dealing with quantities pertaining to the total assemblage, after all element contributions have been resolved. The first-order necessary optimality condition for the Lagrangian function (see Eq. (2.17)), after the imposition of boundary conditions, yields the following relations:

$$\nabla_{\mathbf{y}_i} \mathcal{L} = \nabla_{\mathbf{y}_i} U + [\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g] \vec{\boldsymbol{\lambda}} = \vec{0} \quad (2.35a)$$

$$\nabla_{\mathbf{d}_g} \mathcal{L} = -\nabla_{\mathbf{d}_g} W + [\nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g] \vec{\boldsymbol{\lambda}} = \vec{0} \quad (2.35b)$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L} = \vec{\mathbf{h}}_g = \vec{0} \quad (2.35c)$$

Eq. (2.35a) expresses the equilibrium between external and internal forces acting on the  $i$ -th cross-section for  $i = 1, 2, \dots, m$ , while Eq. (2.35b) ensures consistency between externally

applied loads and Lagrange multipliers. Finally, Eq. (2.35c) requires that all constraints are active when the minimum is attained. Explicit expressions for the first and second derivatives of all quantities involved are given in the Appendix. We should note that derivatives of the strain energy  $U$  are computed numerically during the *cross-section state determination* phase in Sec. (2.4.3).

### 2.4.2 Hessian Matrix

The Hessian matrix of the Lagrangian function contains second order information of the system and is utilized during the solution phase. Eqs. (2.35a)-(2.35c) constitute a set of nonlinear algebraic equations and an iterative scheme is needed to solve the system. In block matrix form, the Hessian is provided as:

$$\mathbf{H} = \begin{bmatrix} \nabla_{\mathbf{y}\mathbf{y}}^2 \mathcal{L} & \nabla_{\mathbf{y}\mathbf{d}_g}^2 \mathcal{L} & \nabla_{\mathbf{y}\lambda}^2 \mathcal{L} \\ \nabla_{\mathbf{y}\mathbf{d}_g}^2 \mathcal{L}^T & \nabla_{\mathbf{d}_g\mathbf{d}_g}^2 \mathcal{L} & \nabla_{\mathbf{d}_g\lambda}^2 \mathcal{L} \\ \nabla_{\mathbf{y}\lambda}^2 \mathcal{L}^T & \nabla_{\mathbf{d}_g\lambda}^2 \mathcal{L}^T & \nabla_{\lambda\lambda}^2 \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{y}\mathbf{y}}^2 \mathcal{L} & \mathbf{0} & \nabla_{\mathbf{y}} \vec{\mathbf{h}}_g^T \\ \mathbf{0} & \mathbf{0} & \nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^T \\ \nabla_{\mathbf{y}} \vec{\mathbf{h}}_g & \nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g & \mathbf{0} \end{bmatrix} \quad (2.36)$$

The resulting Hessian is fairly sparse. Moreover, the block matrix  $\nabla_{\mathbf{y}\mathbf{y}} U$  which contains section stiffness information is block-diagonal (see Appendix C) and its symmetry guarantees the full symmetry of  $\mathbf{H}$ .

### 2.4.3 Solution Scheme

As mentioned in Appendix C, the nonlinear system of equations in Eq. (2.17) is solved in an incremental-iterative fashion. Linearization around the current iteration  $k$  yields:

$$\nabla \mathcal{L}^k + \mathbf{H}^k \delta \vec{\mathbf{z}}^k = \vec{\mathbf{0}} \quad (2.37)$$

where  $\vec{z}$  is the vector of *all* unknowns:

$$\vec{z} = \begin{bmatrix} \vec{y} \\ \vec{d}_g \\ \vec{\lambda} \end{bmatrix}, \quad \text{with} \quad \vec{y} = \begin{bmatrix} \vec{y}_1 \\ \vdots \\ \vec{y}_m \end{bmatrix}$$

Reference to the current iteration step is omitted subsequently for clarity. Solving Eq. (2.37) directly may be cumbersome in some cases, especially for problems that require denser distribution of integration points (e.g. plasticity), since the Hessian is not banded and may also be badly conditioned. In addition, implementation of continuation schemes in order to attain solutions past critical points is more involved since the state variable vector contains displacements, strains and Lagrange multiplier unknowns. A different approach is hence sought, where the initial linear system is reduced to a smaller one involving only the displacement vector  $\vec{d}_g$ . Restating the system in terms of its distinct vector components  $\vec{y}$ ,  $\vec{d}_g$  and  $\vec{\lambda}$  gives:

$$\begin{bmatrix} \nabla_y \mathcal{L} \\ \nabla_{d_g} \mathcal{L} \\ \nabla_\lambda \mathcal{L} \end{bmatrix} + \begin{bmatrix} \nabla_{yy}^2 \mathcal{L} & \mathbf{0} & \nabla_y \vec{h}_g^T \\ \mathbf{0} & \mathbf{0} & \nabla_{d_g} \vec{h}_g^T \\ \nabla_y \vec{h}_g & \nabla_{d_g} \vec{h}_g & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \vec{y} \\ \delta \vec{d}_g \\ \delta \vec{\lambda} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \\ \vec{0} \end{bmatrix} \quad (2.38)$$

Utilizing Eqs. (2.35a)-(2.35c) and setting  $\delta \vec{\lambda} = \vec{\lambda}^{k+1} - \vec{\lambda}^k$ , Eq. (2.38) can be rewritten as:

$$\begin{bmatrix} \nabla_y U \\ -\vec{P} \\ \vec{h}_g \end{bmatrix} + \begin{bmatrix} \nabla_{yy}^2 \mathcal{L} & \mathbf{0} & \nabla_y \vec{h}_g^T \\ \mathbf{0} & \mathbf{0} & \nabla_{d_g} \vec{h}_g^T \\ \nabla_y \vec{h}_g & \nabla_{d_g} \vec{h}_g & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta \vec{y} \\ \delta \vec{d}_g \\ \vec{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \vec{0} \\ \vec{0} \\ \vec{0} \end{bmatrix} \quad (2.39)$$

As mentioned previously, the gradient of the strain energy with respect to the strain vector represents the cross-sectional stress resultants. Solving the first equation in the system of Eq.

(2.39) for  $\delta\vec{y}$  we get:

$$\delta\vec{y} = -\nabla_{yy}^2 \mathcal{L}^{-1} \left[ \vec{F}_{sec} + \nabla_y \vec{h}_g^T \vec{\lambda}_{k+1} \right] \quad (2.40)$$

Substituting Eq. (2.40) in the third equation of Eq. (2.39) and solving for  $\vec{\lambda}^{k+1}$  yields:

$$\vec{\lambda}^{k+1} = \mathbf{B}^{-1} [\vec{h}_g - \vec{b}] + \mathbf{B}^{-1} [\nabla_{d_g} \vec{h}_g] \delta \vec{d}_g \quad (2.41)$$

By substituting Eq. (2.41) into the second system equation Eq. (2.35b) we arrive at the familiar static equilibrium form of the system of equations:

$$\mathbf{K} \delta \vec{d}_g = \vec{P} - \vec{F}_{int} \quad (2.42)$$

which can now be solved for the iterative displacement vector  $\delta \vec{d}_g$ . Vectors  $\vec{b}$ ,  $\vec{F}_{int}$  and matrices  $\mathbf{B}$ ,  $\mathbf{K}$ , are given by the following explicit formulas in the global system:

$$\vec{b} = [\nabla_y \vec{h}_g] [\nabla_{yy}^2 \mathcal{L}]^{-1} \vec{F}_{sec} \quad (2.43a)$$

$$\vec{F}_{int} = [\nabla_{d_g} \vec{h}_g]^T [\mathbf{B}]^{-1} [\vec{h}_g - \vec{b}] \quad (2.43b)$$

$$\mathbf{B} = [\nabla_y \vec{h}_g] [\nabla_{yy}^2 \mathcal{L}]^{-1} [\nabla_y \vec{h}_g]^T \quad (2.43c)$$

$$\mathbf{K} = [\nabla_{d_g} \vec{h}_g]^T [\mathbf{B}]^{-1} [\nabla_{d_g} \vec{h}_g] \quad (2.43d)$$

where  $\vec{b} \in \mathbb{R}^p$ ,  $\vec{F}_{int} \in \mathbb{R}^{3N_{nod}}$ ,  $N_{nod}$  is again the number of structural nodes,  $\mathbf{B} \in \mathbb{R}^{p \times p}$  with  $p = m + 3n_{nel}$ , and  $\mathbf{K} \in \mathbb{R}^{3N_{nod} \times 3N_{nod}}$ .

Equivalently, an assembly process can also be implemented by casting Eqs. (2.43a), (2.43c) in local form, where they can be further simplified by being expanded in terms of element

cross-section contributions:

$$\vec{\mathbf{b}}^e = \sum_{i=1}^{n_q} [\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g] [\nabla_{\mathbf{y}_i}^2 \mathcal{L}]^{-1} \vec{\mathbf{F}}_{sec}^{(i)} \quad (2.44a)$$

$$\mathbf{B}^e = \sum_{i=1}^{n_q} [\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g] [\nabla_{\mathbf{y}_i}^2 \mathcal{L}]^{-1} [\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g]^T \quad (2.44b)$$

where  $n$  is the number of element quadrature points,  $\vec{\mathbf{b}}^e \in \mathbb{R}^{n_q+3}$  and  $\mathbf{B}^e \in \mathbb{R}^{(n_q+3) \times (n_q+3)}$ .

The element stiffness matrix  $\mathbf{K}^e \in \mathbb{R}^{6 \times 6}$  and internal force vector  $\vec{\mathbf{F}}_{int}^e \in \mathbb{R}^6$  are given by the same expressions as in Eqs. (2.43b), (2.43d), but with the gradients now cast in the local element form. For the assembly, the standard FEM routines can be directly employed and the resulting global stiffness operator retains all properties typically associated with it in the context of classical finite element analysis, i.e. it is a symmetric positive definite matrix, it is well-conditioned and, importantly, it is sparse and banded. Hence, in this case, the global internal force vector and the global stiffness matrix are given by the standard assembly process, designated here via operator  $\Lambda$ :

$$\vec{\mathbf{F}}_{int} = \prod_{e=1}^{n_{el}} \vec{\mathbf{F}}_{int}^e \quad (2.45)$$

$$\mathbf{K} = \prod_{e=1}^{n_{el}} \mathbf{K}^e \quad (2.46)$$

with

$$\vec{\mathbf{F}}_{int}^e = [\nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^e]^T [\mathbf{B}^e]^{-1} [\vec{\mathbf{h}}_g^e - \vec{\mathbf{b}}^e] \quad (2.47)$$

$$\mathbf{K}^e = [\nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^e]^T [\mathbf{B}^e]^{-1} [\nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^e] \quad (2.48)$$

where  $\vec{\mathbf{h}}_g^e \in \mathbb{R}^{n_q+3}$  denotes the *element* constraint vector given by Eq. (2.28) and  $\nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^e \in \mathbb{R}^{(n_q+3) \times 6}$  its gradient with respect to the global displacement degrees of freedom (DOFs) asso-

ciated with it, is given in Appendix B. It is clear that with this formulation the inversion of large global matrices for the computation of  $\mathbf{K}$  and  $\vec{\mathbf{F}}_{int}$  is avoided and, instead, only inversion of the local elements flexibility matrices  $\mathbf{B}^e$  is required. Given that for highly nonlinear problems we typically have  $n \in \mathbb{N}([5, 10])$  for satisfactory accuracy, this results in an element flexibility matrix of dimension  $\dim(\mathbf{B}^e) \leq 13$ , thus accelerating the analysis considerably.

Having written the system of equations in the form of Eq. (2.42), implementation of arc-length type schemes is now straightforward. In the present work, we adopt the algorithm proposed by Crisfield [87] whereby the additional equation supplemented to the system is:

$$\Delta \vec{\mathbf{d}}_g^T \Delta \vec{\mathbf{d}}_g = \Delta s^2 \quad (2.49)$$

with  $\Delta s$  being the user-specified arc-length parameter. The incremental displacement vector  $\Delta \vec{\mathbf{d}}_g$  is updated in each iteration as follows:

$$\Delta \vec{\mathbf{d}}_g^k = \Delta \vec{\mathbf{d}}_g^{k-1} + \delta \vec{\mathbf{d}}_g^k \quad (2.50)$$

where  $k$  denotes the current iteration and  $\delta \vec{\mathbf{d}}_g^k$  is the vector of iterative displacements.

### *Solution updating*

After the determination, at an arbitrary iteration  $k$  within step  $j$ , of the iterative displacement vector  $\delta \vec{\mathbf{d}}_g^k$  from Eq. (2.42), the strain and Lagrange multiplier vectors in Eqs. (2.40),(2.41) have then to be updated. The detailed steps of the updating procedure are as follows:

*Step  $j$ , iteration  $k$ :*  $\left\{ \vec{\mathbf{y}}_j^0, \quad \vec{\mathbf{d}}_{g,j}^0, \quad \vec{\lambda}_j^0, \quad \vec{\mathbf{P}}_j, \quad \Delta \vec{\mathbf{y}}_j^{k-1}, \quad \Delta \vec{\mathbf{d}}_{g,j}^{k-1} \right\}$

**(1):** Get section stiffnesses  $\nabla_{yy}^2 \mathcal{L}^k$  from Eq. (C) in Appendix C and section forces  $\vec{\mathbf{F}}_{sec}^k$  from section integration, Eq. (2.55).

**(2):** Evaluate  $\vec{\mathbf{b}}^k$  and  $\mathbf{B}^k$  from Eqs. (2.43a), (2.43c) respectively (or, alternatively,  $\vec{\mathbf{b}}^{e,k}$ ,  $\mathbf{B}^{e,k}$  from Eqs. (2.44a), (2.44b)).

(3): Evaluate  $\mathbf{K}^k$  from Eq. (2.43d), (or, alternatively, from Eqs. (2.46), (2.48)).

(4): Solve Eq. (2.42) for  $\delta\vec{\mathbf{d}}_g^k$ .

(5): Update incremental displacement vector:  $\Delta\vec{\mathbf{d}}_{g,j}^k \leftarrow \Delta\vec{\mathbf{d}}_{g,j}^{k-1} + \delta\vec{\mathbf{d}}_g^k$

(6): Update displacement vector:  $\vec{\mathbf{d}}_{g,j}^k \leftarrow \vec{\mathbf{d}}_{g,j}^0 + \Delta\vec{\mathbf{d}}_g^k$

(7): Update Langrange multiplier  $\vec{\lambda}_j^k$  from Eq. (2.41).

(8): Evaluate iterative strain vector  $\delta\vec{\mathbf{y}}^k$  from Eq. (2.40).

(9): Update incremental strain vector:  $\Delta\vec{\mathbf{y}}_j^k \leftarrow \Delta\vec{\mathbf{y}}_j^{k-1} + \delta\vec{\mathbf{y}}^k$

(10): Update total strain vector:  $\vec{\mathbf{y}}_j^k \leftarrow \vec{\mathbf{y}}_j^0 + \Delta\vec{\mathbf{y}}_j^k$

(11): Evaluate  $\vec{\mathbf{F}}_{int}^k$  from Eq. (2.43b) (or, alternatively, from Eqs. (2.45), (2.47)).

(12): Check  $\|\vec{\mathbf{P}}_j - \vec{\mathbf{F}}_{int}^k\| \leq \text{tol} \cdot \|\vec{\mathbf{P}}_j - \vec{\mathbf{F}}_{int}^0\|$ .

(13): If **FALSE**, set  $k \leftarrow k + 1$  and go to (1). If **TRUE**, set  $j \leftarrow j + 1$ ,  $\vec{\mathbf{y}}_j^0 \leftarrow \vec{\mathbf{y}}_{j-1}^k$ ,  $\vec{\mathbf{d}}_{g,j}^0 \leftarrow \vec{\mathbf{d}}_{g,j-1}^k$ ,  $\vec{\lambda}_j^0 \leftarrow \vec{\lambda}_{j-1}^k$ .

#### 2.4.4 Cross-Section State Determination

During the updating scheme described in the previous section, the section stress resultants  $\vec{\mathbf{F}}_{sec}$ , along with the matrix  $\nabla_{yy}^2 \mathcal{L}$ , are required. The latter, as seen subsequently, corresponds to the generalized section stiffnesses. In the general case where inelastic behavior is considered, the stress-strain constitutive law has to be integrated at the cross-section level and the material properties have to be updated accordingly. In this work, each cross-section is discretized in  $n_l$  number of layers and the stress update is performed independently for each layer. This is equivalent to the composite midpoint rule applied along the height of the cross-section. The shear coefficient  $k_s$  is introduced as a correction factor for the simplifying aforementioned assumption, in accordance with Cowper [131].

Below we present the kinematic and constitutive relations for a cross-section object. The (multiaxial) constitutive law at the fiber level is treated in the next chapter. The update procedure is regarded as *strain-driven*, in the sense that the known initial state is updated given an increment in the centerline strains at a particular quadrature point. First, we define the stress and strain vectors associated with a particular fiber:

$$\vec{\sigma}_f = \begin{bmatrix} \sigma_{11} & \sigma_{12} \end{bmatrix}^T, \quad \vec{\epsilon}_f = \begin{bmatrix} \epsilon_{11} & \gamma_{12} \end{bmatrix}^T \quad (2.51)$$

In line with the plane section hypothesis, the fiber strain vector can be determined from the centerline strain vector as follows:

$$\begin{aligned} \epsilon_{11} &= \varepsilon - X_2 \kappa \\ \gamma_{12} &= \varphi(X_2) \gamma \end{aligned} \iff \vec{\epsilon}_f = \mathbf{N}_s \vec{q}, \quad \mathbf{N}_s = \begin{bmatrix} 1 & 0 & -X_2 \\ 0 & \varphi(X_2) & 0 \end{bmatrix} \quad (2.52)$$

where  $\varphi(X_2)$  is an as of now unspecified function that defines a shear strain distribution along the height of the cross-section.

Thus, given  $\{\vec{q}, \vec{q}^{pl}\}$ ,  $\Delta\vec{q}$ , along with a set of internal state variables (e.g. accumulated plastic strain), we can evaluate the incremental strains at the midpoint of a layer at distance  $X_2$  from the neutral axis as follows:

$$\vec{\epsilon}_f = \mathbf{N}_s \vec{q} \quad (2.53a)$$

$$\Delta\vec{\epsilon}_f = \mathbf{N}_s \Delta\vec{q} \quad (2.53b)$$

where it is reminded that  $\vec{q}$  is the vector containing the neutral axis generalized strains and  $\vec{\epsilon}_f$  is the strain vector associated with the fiber midpoint. With Eqs. (2.53a), (2.53b) and the initial state known, we can perform the stress update which will yield the updated stress vector  $\vec{\sigma}_f$  and the (consistent) elastoplastic modulus for the fiber. The conjugate stress resultants associated with the assumed kinematic assumptions are derived from the element virtual work equation:

$$\vec{P}_l^T \delta \vec{d}_l + \vec{P}_r^T \delta \vec{d}_r = \int_V \vec{\sigma}_f^T \delta \vec{\epsilon}_f dV = \int_0^L \left[ \int_A \mathbf{N}_s^T \vec{\sigma}_f dA \right]^T \vec{q} dX_1 \quad (2.54)$$

where we define the conjugate section stress resultants associated with  $\vec{q}$  as follows:

$$\vec{F}_{sec}^{(i)} = \int_A \mathbf{N}_s^T \vec{\sigma}_f dA \quad (2.55)$$

Application of the composite midpoint rule on Eq. (2.55) yields:

$$\vec{F}_{sec}^{(i)} \approx \begin{bmatrix} \sum_{j=1}^{n_l} \sigma_{11,j} \Delta A_j \\ \sum_{j=1}^{n_l} \sigma_{12,j} \Delta A_j \\ \sum_{j=1}^{n_l} X_{2,j} \sigma_{11,j} \Delta A_j \end{bmatrix} \quad (2.56)$$

where  $\Delta A_j$  the area of layer  $j$ .

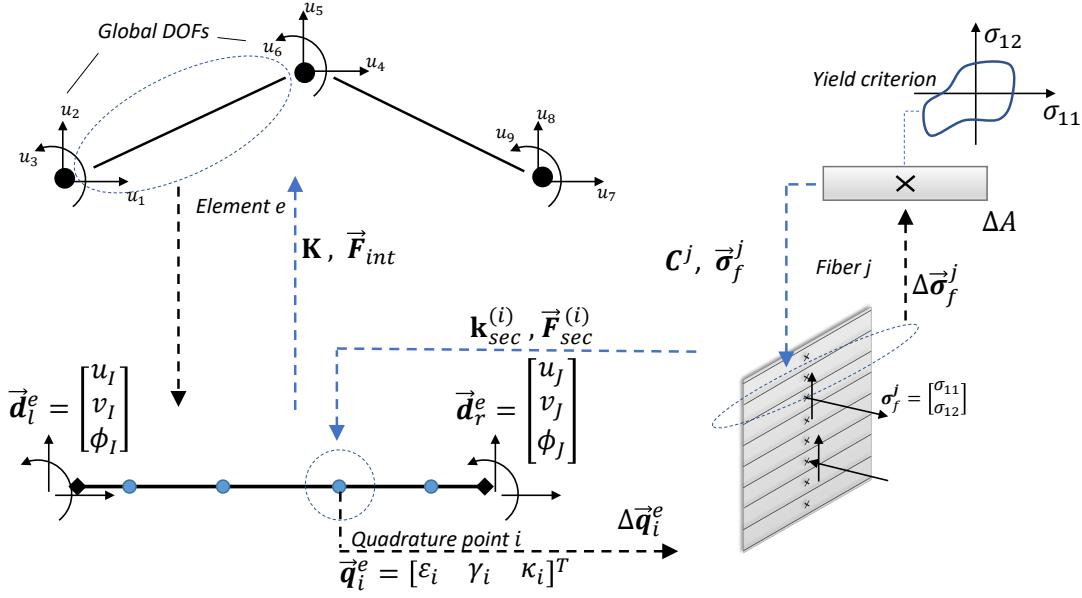
Let  $\mathbf{C} = \partial \vec{\sigma}_f / \partial \vec{\epsilon}_f$  designate the tangent modulus at a fiber. During elastic steps,  $\mathbf{C} \equiv \mathbf{C}^{el} = \text{diag}[E, G]$ . In the next chapter we will see that during plastic steps the diagonal components of  $\mathbf{C}$  are, generally, also non-zero. The tangent section stiffness is derived as follows:

$$\mathbf{k}_{sec}^{(i)} = \frac{\partial \vec{F}_{sec}^{(i)}}{\partial \vec{q}} = \int_A \mathbf{N}_s^T \frac{\partial \vec{\sigma}_f}{\partial \vec{q}} dA = \int_A \mathbf{N}_s^T \frac{\partial \vec{\sigma}_f}{\partial \vec{\epsilon}_f} \frac{\partial \vec{\epsilon}_f}{\partial \vec{q}} dA = \int_A \mathbf{N}_s^T \mathbf{C} \mathbf{N}_s dA \quad (2.57)$$

Application of the midpoint rule on Eq. (2.57) yields the following component form for  $\mathbf{k}_s^{(i)}$ :

$$\mathbf{k}_{sec}^{(i)} = \begin{bmatrix} C_{11}^i & \varphi^i C_{12}^i & -X_2^i C_{11}^i \\ \varphi^i C_{21}^i & (\varphi^i)^2 C_{22}^i & -x_2^i \varphi^i C_{21}^i \\ -X_2^i C_{11}^i & -x_2^i \varphi^i C_{12}^i & (X_2^i)^2 C_{11}^i \end{bmatrix} \quad (2.58)$$

where  $C_{ij}$  are the components of the elastoplastic consistent tangent modulus of the fiber. The



**Figure 2.4.** Solution process stages. Quantities  $\mathbf{K}^e$  and  $\vec{\mathbf{F}}_{int}$  stand for stiffness matrix and internal nodal force vector for element  $e$  respectively.

section stiffness is then given by the following sum:

$$\mathbf{k}_{sec} = \sum_{i=1}^{n_f} \mathbf{k}_{sec}^{(i)} \Delta A^i \quad (2.59)$$

Given an increment in nodal displacements at the global/structural level, the assembly process has to loop over all elements in order to determine the element stiffness and internal nodal forces. This procedure, illustrated in Fig. 2.4, involves analysis at three different levels: the global/element, the section state determination, and the fiber stress update. Having briefly outlined the basic theory for the hybrid element as it pertains to the first two levels mentioned above, next we introduce the elastoplastic stress update formulation, which pertains to the third level of the solution process and results in a fast return mapping algorithm.

The generalized section stiffness,  $\nabla_{\mathbf{y}_i \mathbf{y}_i}^2 \mathcal{L}$ , is given by the second derivatives of the Lagrangian with respect to strain vector  $\vec{\mathbf{q}}_i = [\vec{\mathbf{q}}_i^T \ \phi_i]^T$ :

$$\nabla_{\mathbf{y}_i \mathbf{y}_i}^2 \mathcal{L} = \nabla_{\mathbf{y}_i \mathbf{y}_i}^2 U + \nabla_{\mathbf{y}_i} ([\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g] \vec{\lambda}) \quad (2.60)$$

where:

$$\nabla_{\mathbf{y}_i \mathbf{y}_i}^2 U = w_i \begin{bmatrix} \mathbf{k}_{sec}^{(i)} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

For the explicit expression of the second term in the right-hand side of Eq. (2.60), see Appendix C. It generally includes parameters associated with section rotations and Lagrange multipliers, and is greatly simplified if small displacement assumptions are used.

## 2.5 Numerical Examples

In this section we present the capabilities and efficiency of the proposed element in a number of well-known benchmark examples. In cases where analytical solutions are available, they are included in the comparisons. Our results are compared against the ones achieved i) with the structural FEM software OpenSees[132], using its flexibility-based beam element [123, 133] with five quadrature points and fiber section integration, and ii) with the FEM software Abaqus[134], using plane-stress modeling for the frame members. For the elastic analyses, reduced integration linear quadrilateral elements (CPS4R) were used in Abaqus to avoid shear-locking, while for the elastoplastic simulations, reduced integration quadratic quadrilaterals (CPS8R) elements were used instead. For all cases the Gauss-Legendre quadrature rule is used for approximating the integrals in Eqs. (2.19),(2.22). For the elastoplastic analyses, we assume a linear isotropic hardening law and the elastoplastic modulus is given as a percentage  $r$  of the elastic modulus  $E$ . Moreover, we assume a rigid rectangular cross-section shape for all cases, with shear coefficient  $k_s = 0.870$ [131]. For all examples, the SI units corresponding to material and geometric properties are Pa and m, while for the load units Newton(N) is used. The convergence termination criterion for the iterative procedure was  $10^{-7}$  for all cases. Shear deformation is allowed to occur in all analyses unless stated otherwise.

### 2.5.1 Cantilever with Tip Load

In this first example we demonstrate the element performance with fully nonlinear assumptions against available analytical solutions, as well as its locking-free behavior when small displacement assumptions hold.

#### *Nonlinear analyses*

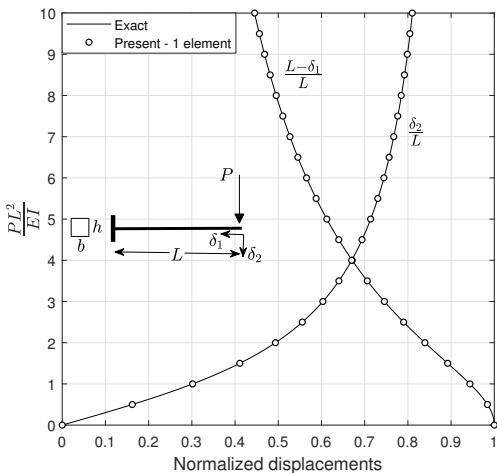
Both elastic and inelastic geometrically nonlinear cases are considered. One element and six integration points are used, with the beam assumed to be inextensible, and the results are presented in Fig. 2.5. The elastic case assumed  $EI = 10$ ,  $L = 1$  and shear-rigid response for comparison purposes with available exact equilibrium paths[135], while the inelastic case allows for shear effects to take place with  $GA_s = 500$ . In order to maintain consistency with the attributes  $EI = 10$  and  $L = 1$ , we used  $E = 200 \times 10^9$  for Young's modulus and prescribed the height of the cross-section to be  $h = 0.02$ , which is now required in order to carry out the cross-sectional stress-update. For the material nonlinearity example, the yield stress is  $\sigma_y = 180 \times 10^7$  with a tangent-to-elastic modulus ratio  $r = 3\%$  and results are compared against Abaqus and OpenSees. As can be seen in Fig. 2.5, in both cases remarkable accuracy is achieved with just one element. Slight differences between the Abaqus quadrilateral model and the beam models are observed when extensive plastification takes place. In Table 2.1 we also provide distinct comparisons with exact values for the elastic case. It is clear that when the normalized load is equal to 1, displacements remain small, and thus, only  $n = 4$  quadrature points suffice to achieve small absolute error. At the maximum load level, however,  $n = 6$  quadrature points were required for highly accurate results. For the shear-rigid case exact solutions are computed from numerical evaluation of the elliptic integrals that describe the problem. The reader can also consult Table 1 in [136] for a wide range of values reported. Additional results pertaining to the elastic shear-flexible case are also reported in Table 2.1. For this case, we used again  $GA_s = 500$ ,  $EI = 10$ ,  $L = 1$  and compared against available analytical solutions from[137], where the normalized load level therein is held constant and equal to 1. The values correspond-

**Table 2.1** Comparison of tip normalized displacements for different load levels for elastic response.

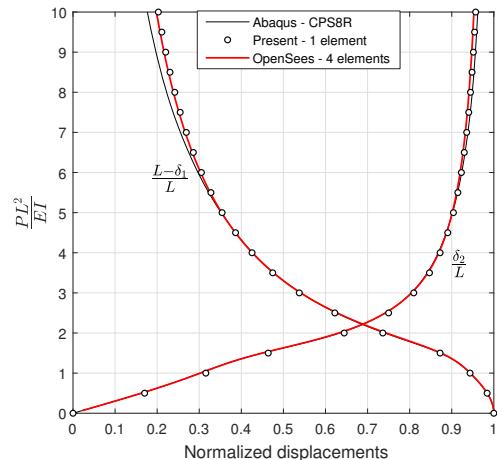
$\frac{PL^2}{EI}$	$n$	Shear Rigid			Shear Flexible			
		Exact	Present	Error	Exact	Present	Error	
$\frac{\delta_2}{L}$	1	4	0.3017207	0.3017206	1.12E-07	0.3178138	0.3178137	1.64E-07
$\frac{L - \delta_1}{L}$	10	6	0.8106090	0.8106113	2.32E-06	-	0.8536137	-
$\frac{L - \delta_1}{L}$	1	4	0.9435668	0.9435666	1.82E-07	0.9386844	0.9386842	1.75E-07
$\frac{L - \delta_1}{L}$	10	6	0.4450044	0.4450017	2.62E-06	-	0.4208910	-

ing to the maximum normalized load equal to 10 are also reported. Although exact solutions are not available for this load level to compare against, we can see that the relative change in the transverse and axial displacement with respect to the shear-rigid case is consistent and about 5.30% and 5.40%, respectively.

In Fig. 2.6 we see how the moment and curvature distributions change based on the beam deformations for both cases of Fig. 2.5, captured at distinct load levels. The moment values at the left fixed end are also reported on the graphs. The observed nonlinear distributions for the elastic case are due to the deformed shape of the beam being drastically different from its initial configuration. As seen, an important feature of the formulation is its ability to capture curvature localization when plastic hinges are also formed. In displacement-based conventional beam elements, a fine mesh is in constrast required in the areas where plastification is expected.

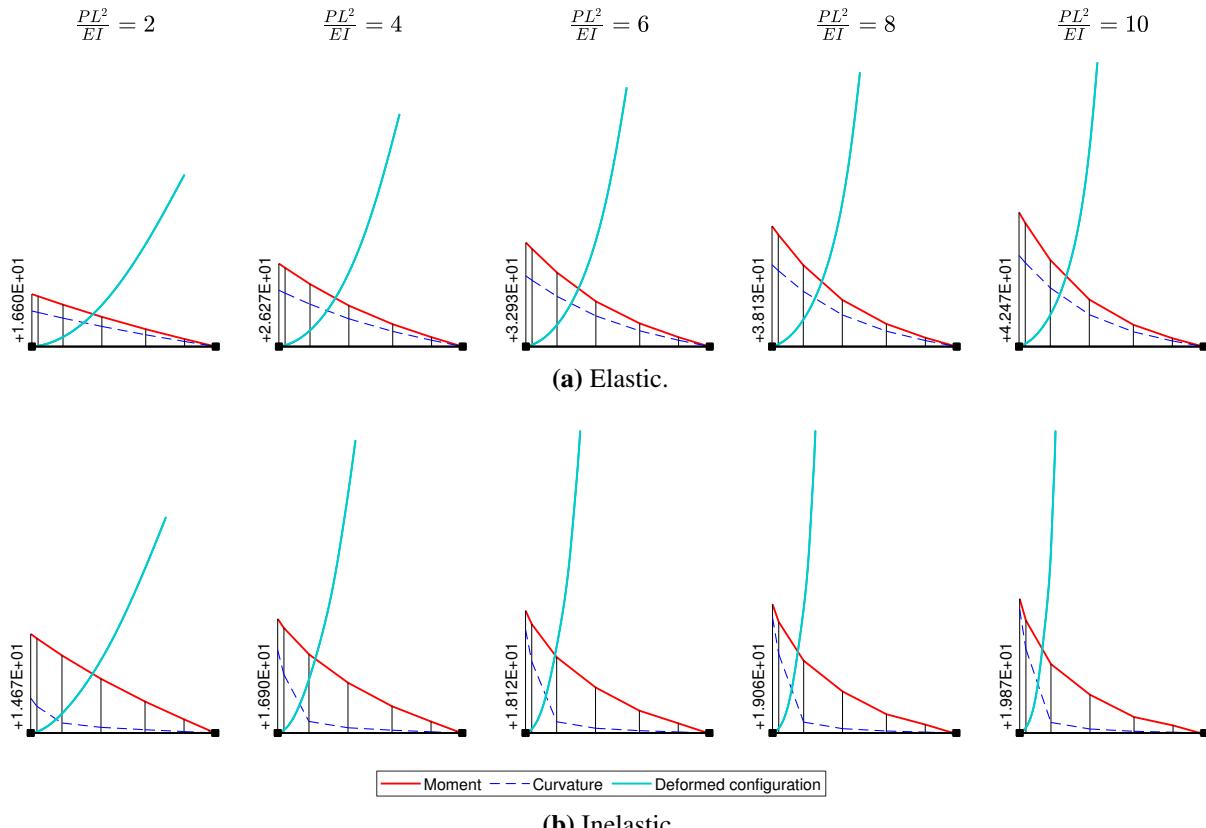


(a) Elastic analysis

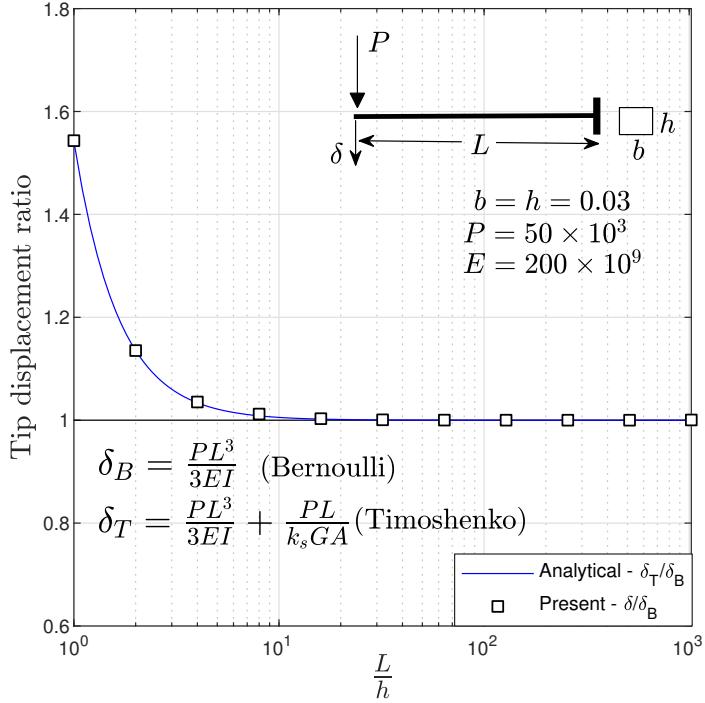


(b) Inelastic analysis

**Figure 2.5.** Geometrically nonlinear analyses of cantilever.



**Figure 2.6.** Moment and curvature distributions for the nonlinear analyses of the cantilever - one element with  $n = 6$  quadrature points.



**Figure 2.7.** Influence of  $L/h$  on the tip deflection and comparison with analytical solution.

### *Shear-locking*

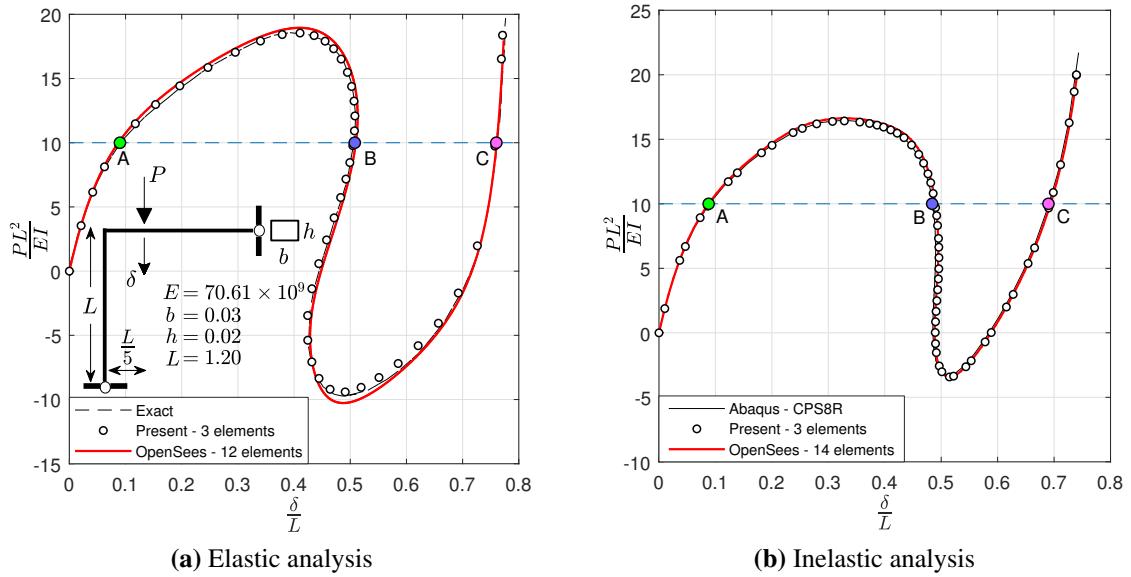
Shear locking may occur in shear deformable elements undergoing bending. Due to inconsistent interpolation of fields that are naturally related, spurious shear stresses develop and the element response appears to be stiffer, particularly as the beam slenderness is increased. Appropriate numerical treatments include using field consistent higher order elements or applying reduced numerical integration [138, 139, 140]. Here, we perform small displacement elastic analyses for various length-to-thickness ratios. The tip displacement ratios of  $\delta_T/\delta_B$  and  $\delta/\delta_B$  are then plotted against the length to thickness ratio  $L/h$ , where  $\delta_B$  and  $\delta_T$  denote the Bernoulli and Timoshenko displacements respectively. Only one element is used with two integration points. The section dimensions are kept fixed, while the length  $L$  of the beam varies. Results, along with used data are shown in Fig. 2.7, where it can be seen that the proposed element exhibits locking-free behavior.

### 2.5.2 Lee's Frame

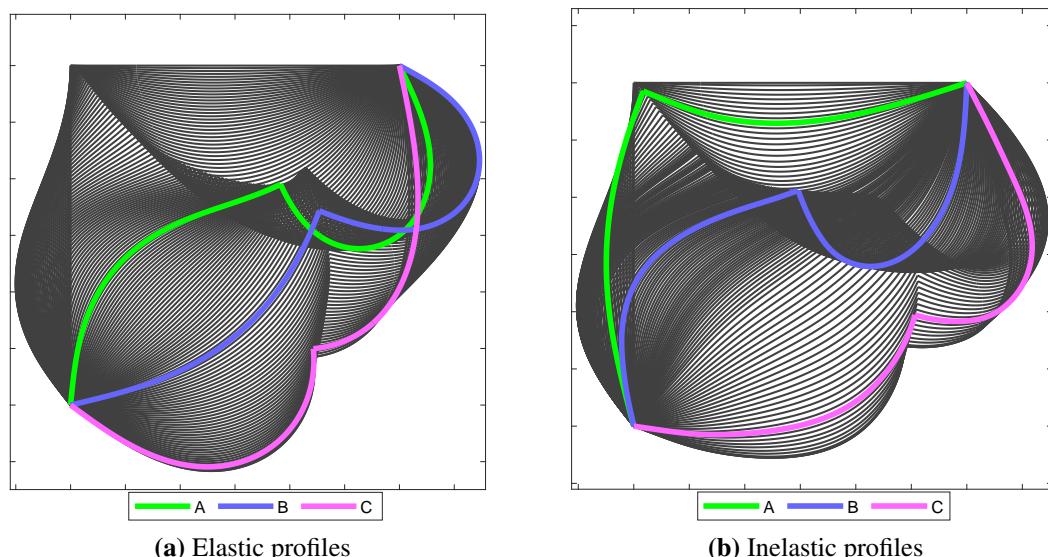
Lee's frame is a well-known benchmark for geometrically nonlinear capabilities of beam-column element formulations. It was first analyzed by Seng-Lip Lee et.al. in [141] where the exact solution to the problem was also provided. We first consider here the classic case, where the frame is pinned on both supports. An investigation on the effect of shear flexibility to the elastic response is also carried out by varying the length-to-thickness ratio, while maintaining large-displacement assumptions. It is demonstrated that for ratios smaller than five, the equilibrium path ceases to be similar to the shear-rigid case. Lastly, we investigate the response of the same frame when both supports are fixed. This again leads to a drastically different response, with abrupt changes in the equilibrium path. In all cases, the centerline is assumed inextensible.

#### *Frame with pinned supports*

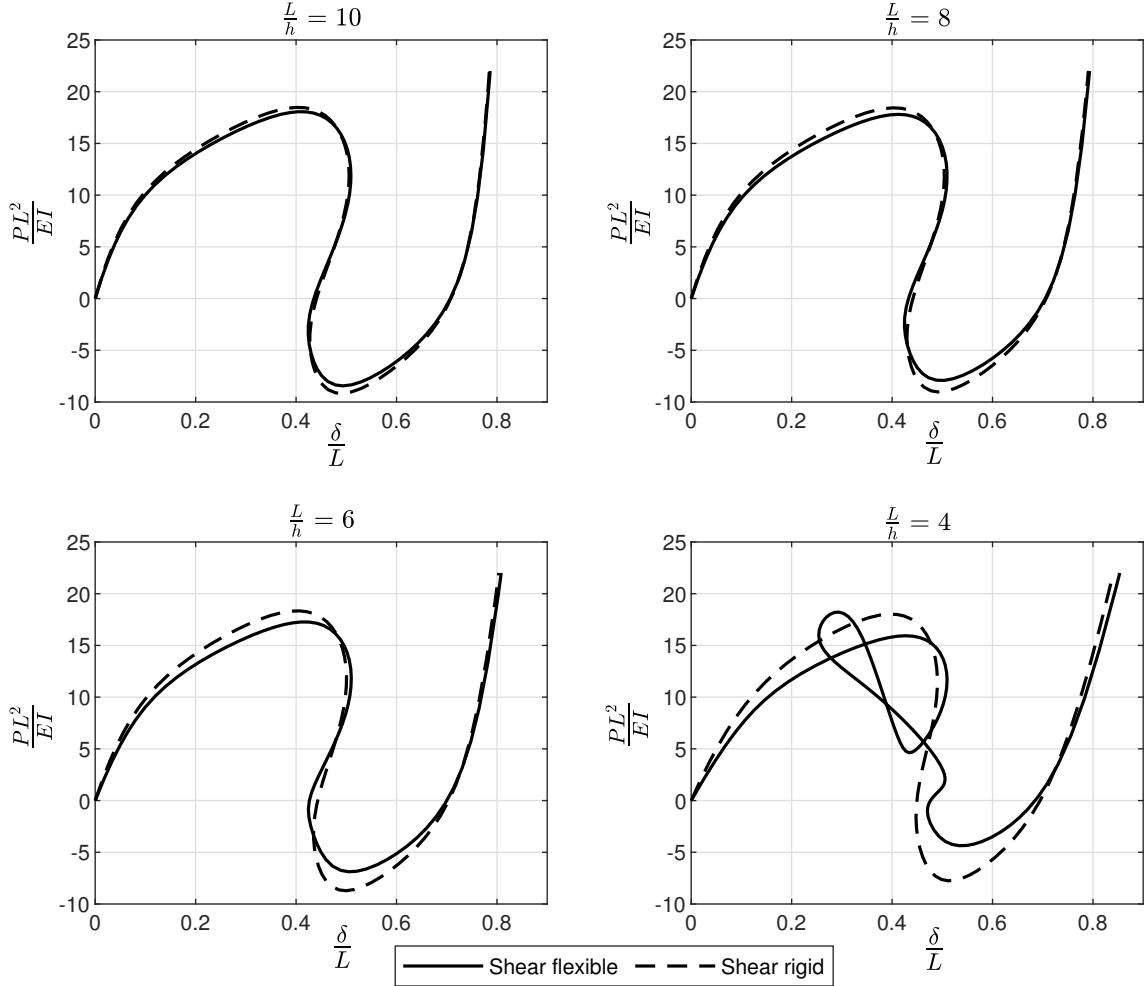
An elastic, geometrically nonlinear analysis is first performed. The minimum number of elements that can be used for this specific example is three and a 4-point quadrature is applied for each element, whereas twelve flexibility elements are required in OpenSees to achieve comparable accuracy. For the elastoplastic analysis, again three elements are used in our simulation with eight integration points, whereas fourteen flexibility elements are now required in OpenSees for comparable accuracy. The yield stress is  $\sigma_y = 1.3 \times 10^9$  and the tangent-to-elastic modulus is  $r = 3.00\%$ . In the elastic analysis, the frame members are also considered shear-rigid, in order to remain consistent to the initial assumptions in the original investigation[141]. Results are shown in Fig. 2.8, with the colored markers corresponding to the deformed profiles of the same color shown in Fig. 2.9.



**Figure 2.8.** Geometrically nonlinear analyses of Lee's frame with pinned supports.



**Figure 2.9.** Deformation profiles for Lee's frame with pinned supports.

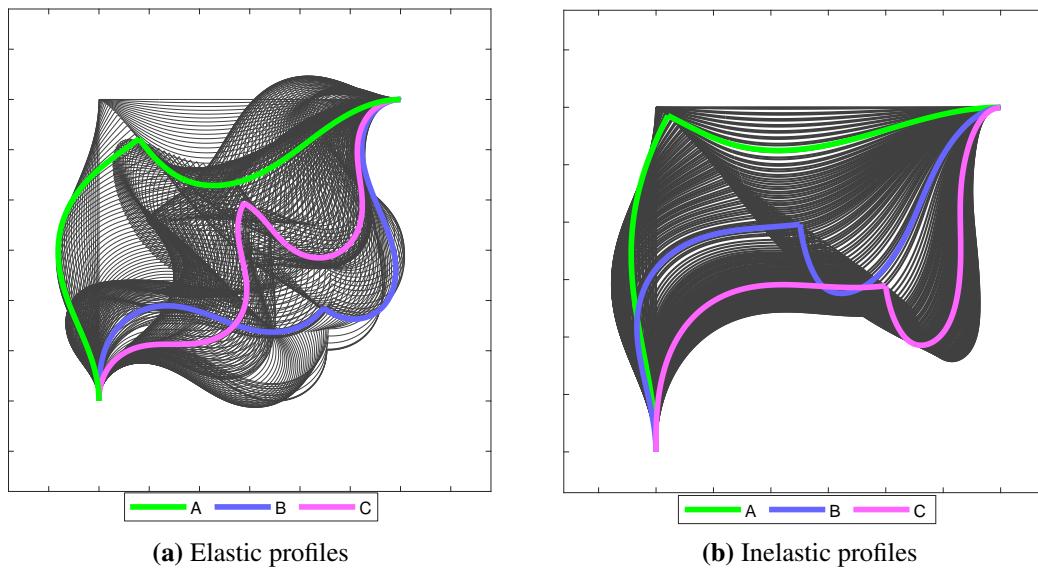


**Figure 2.10.** Effect of shear flexibility on Lee’s frame response.

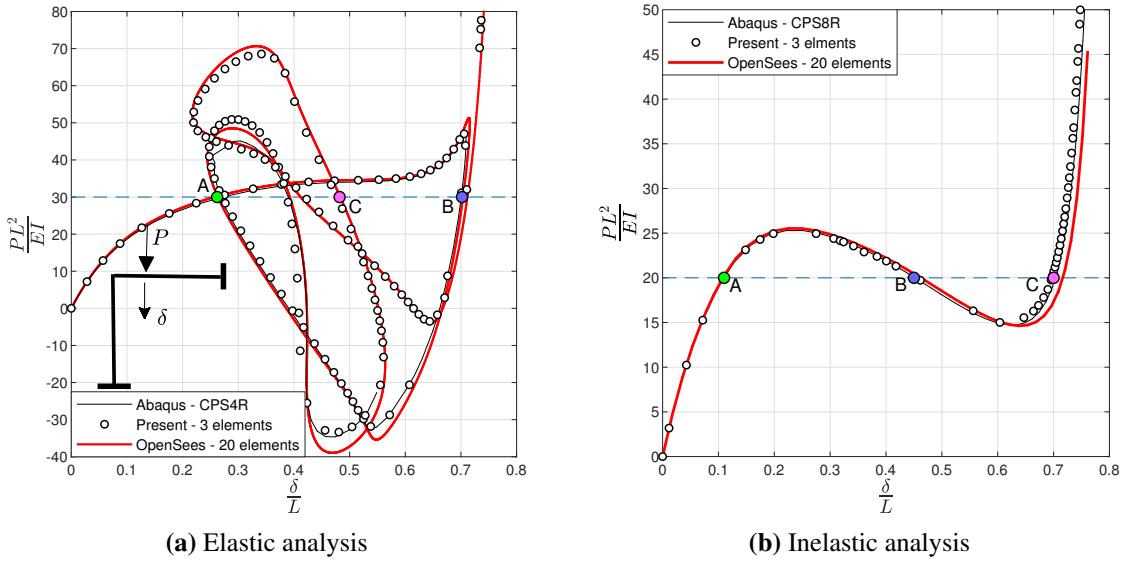
In Fig. 2.10, results related to shear flexibility effects on the frame’s response are shown. As previously mentioned, in [141] the members are assumed to be shear rigid, effectively adhering to the Bernoulli assumptions. In this example, we relax this assumption and examine how the slenderness decrease affects the structural response. As seen in the figure, four different length-to-thickness ratios are examined and the equilibrium path gradually deviates from the shear-rigid case. For ratio  $L/h = 4$ , the equilibrium path is drastically different and the structure, after entering the snap-back region, follows a rather complex equilibrium branch. Note that around  $\delta/L = 0.5$ , the structure exits its current branch and continues on a similar path as in the previous cases.

### *Frame with fixed supports*

This variation of the benchmark problem results in a response that exhibits a dramatic snap-back, with abrupt changes in the stiffness sign along the equilibrium path for the nonlinear elastic problem. Seven integration points are used here per element and, again, one element per member, to achieve the same accuracy as with the Abaqus plane stress model, which terminated prematurely. In contrast, twenty elements are necessary in OpenSees for comparable results. The geometric and material properties are the same as in Sec. (2.5.2) (see Fig. 2.8). For the inelastic analysis, five integration points are sufficient, since the response is now fairly smooth overall, in comparison to the elastic case. All members are shear flexible in these examples, however the contribution of shear deformation to the overall responses is negligible for both elastic and inelastic cases. The response and deformation profiles for both analyses are shown in Figs. (2.12),(2.11) respectively.



**Figure 2.11.** Deformation profiles for Lee's frame with fixed supports.

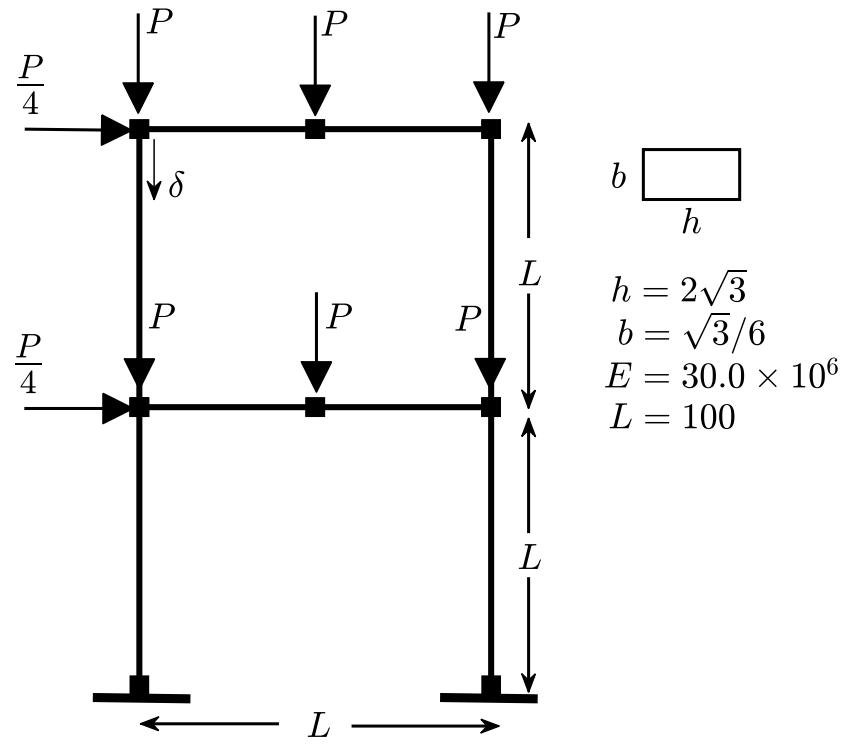


**Figure 2.12.** Geometrically nonlinear analyses of Lee’s frame with fixed supports.

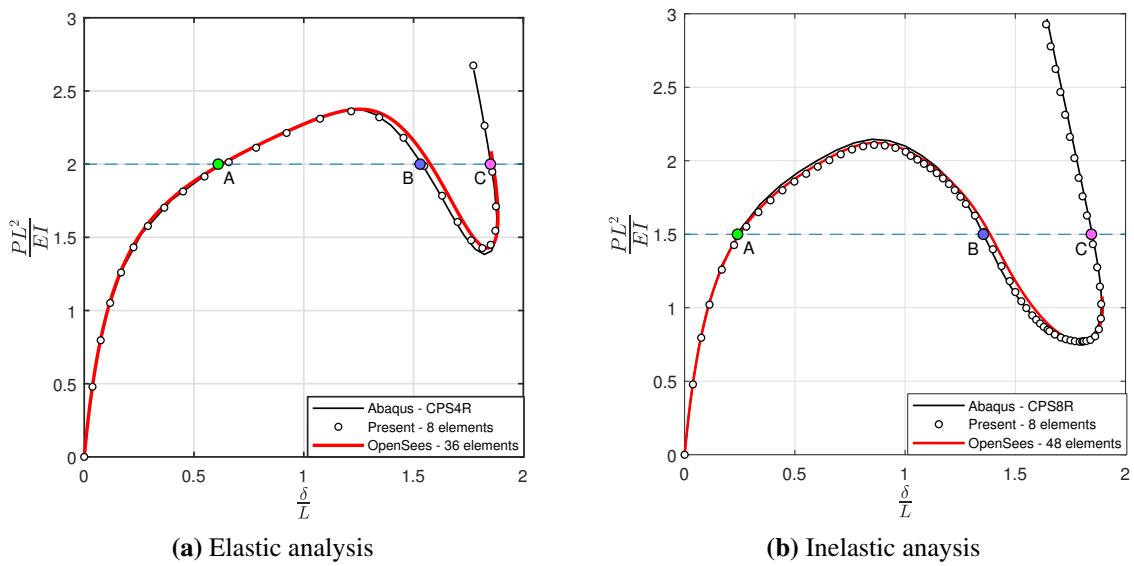
### 2.5.3 Two-Storey Frame

In this example a two-story frame with one bay is analyzed, first studied regarding its plastic limit load in [142]. It was also later used as a benchmark example by Santos [143], for the performance of a nonlinear beam finite element based on complementary energy principles. The example is slightly modified here, however the rectangular cross-section dimensions of the structural members and their length  $L$  have been chosen so their the slenderness remains in accordance with [143]. The yield stress is  $\sigma_y = 0.9 \times 10^6$ , and the tangent-to-elastic modulus ratio is  $r = 3\%$ . The full frame characteristics and discretization are shown in Fig. 2.13.

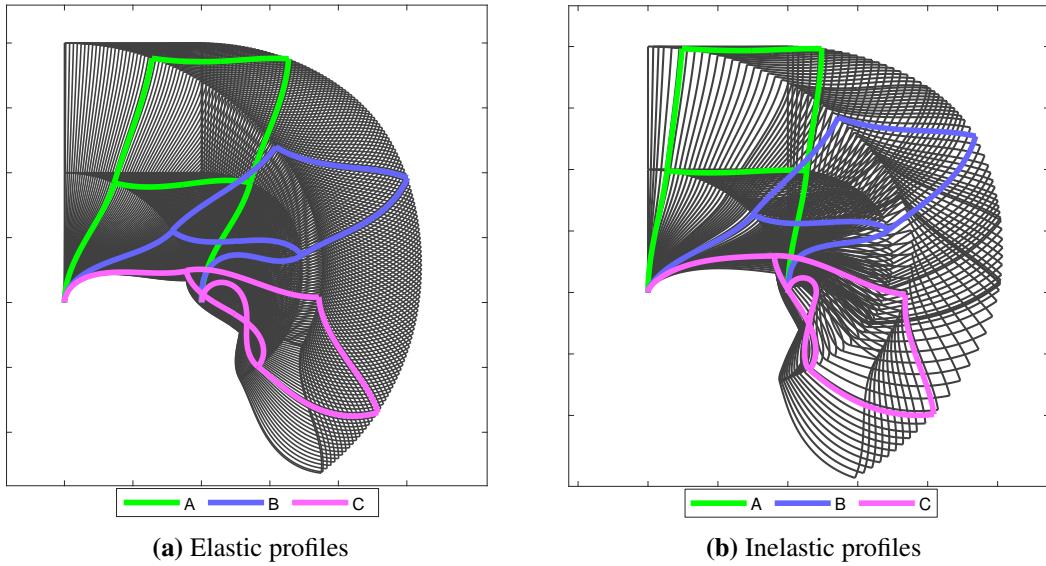
The minimum number of hybrid elements for this example is eight and the number of integration points per element for the elastic and plastic analyses are seven and ten, respectively. For the OpenSees analyses, 36 and 48 elements were used in total for the elastic and inelastic analyses. In both cases, however, the OpenSees analyses stopped prematurely. Results are demonstrated in Fig. 2.14, showing excellent agreement with Abaqus results, while the relevant deformation profiles are provided in Fig. 2.15.



**Figure 2.13.** Two-storey portal frame discretization.



**Figure 2.14.** Geometrically nonlinear analyses of the two-storey frame.



**Figure 2.15.** Deformation profiles for two-storey frame.

## 2.6 Summary

A novel geometrically exact hybrid beam element formulation is presented in this work, conceptually and numerically developed using nonlinear programming principles, instead of the traditionally used FEM techniques. Structural analysis is accordingly seen as a nonlinear program, where the total potential energy functional is treated as the objective function of the problem, constrained in the space of feasible kinematic configurations of the structure. Field integrals in the objective function and kinematic constraints are functions approximated by an appropriate quadrature rule, and kinematics of large displacements and rotations are modeled using the exact strain-displacement relations of Reissner's beam theory. These kinematic equations serve as constraints of the nonlinear program and are appended to the objective function by means of Lagrange multipliers. With this formulation, the involved shear and axial strain-related fields do not need to be interpolated, but are rather collocated at certain quadrature points inside the element, as defined by the numerical integration method. The only interpolated field is curvature, whereas displacement unknowns eventually emerge only at element edges. A Newton-based numerical approach to the Lagrangian of the problem is presented, based on the second-order

approximation of the necessary optimality conditions, and it is also appropriately connected to standard FEM assemblage techniques. Hence, using the block elimination technique, FEM assembly routines can be directly reused, while the global stiffness is assembled from the inversion of local flexibility operators. This is computationally efficient because these matrices are processed independently and their dimension is upper bounded, with the number of quadrature points per element ranging from 5 to 10. The presented approach is also capable to capturing flexural strain localization due to plastification even when geometric nonlinearities are present. The efficiency of the suggested formulation is demonstrated through several benchmark examples and is shown to be capable to accurately capture highly nonlinear structural behaviors with one element per member in all cases, accounting for shear flexibility and locking-free capabilities, while also favorably comparing against well-established beam-column and quadrilateral finite element methodologies.

## CHAPTER 3

### MATERIAL CONSTITUTIVE MODEL

#### 3.1 Introduction

In this chapter we introduce an algorithm for shear-axial-flexure interaction suited for fibre beam elements of any kind (displacement-based or otherwise). The coupling is taken into account by considering a multiaxial constitutive law at the level of cross-section fibers. Thus, stress resultant interaction as well as the consistent tangent stiffness of a cross-section are derived during the state determination phase. Integration of the elastoplastic rate equations on each fiber is carried out using the fully backward (implicit) Euler method, which as is known, results in the so-called closest point projection algorithm. In addition, the constrained stress state of a fiber is exploited in order to devise a return-mapping scheme that avoids incorporating stress components that are not energetically active. The appropriate mapping on the constrained space is inspirited by the work of Simo[144] and appropriately adapted for fibre discretized elements. This leads to a stress update algorithm that is significantly faster while also requiring less memory requirements for storage of non-active tensor components, compared to formulations that utilize a general three dimensional algorithm[48, 49, 50, 51, 52]. In the latter case, stress update is a nested procedure within an outer Newton iteration which is necessitated in order to enforce the constraint for the transverse stress components,  $\sigma_{22} = \sigma_{33} = 0$ . This is because during the trial phase of a plastic step, the transverse components become non-zero. Furthermore, a static condensation of the three-dimensional consistent tangent is required to enforce both the local Newton scheme and to derive the consistent tangent modulus pertaining to the fiber stress state. The algorithm introduced in this chapter bypasses the need for additional outer iterations, thereby resulting in a much faster stress update procedure.

For the exposition we adopted the  $J_2$  constitutive model which is based on the von Mises

yield criterion, which is suited for modelling the elastoplastic response of metals. Linear kinematic and general isotropic hardening are incorporated in the formulation. Moreover, infinitesimal strains and a rate-independent associative plasticity framework are the core assumptions that underlie the proposed algorithm.

This chapter is subdivided in three main sections: first, we briefly outline the general (rate) form of the three-dimensional elastoplastic equations. In the second section we present the rate constitutive model as it pertains to the fiber stress state and derive expressions for the continuous elastoplastic moduli. In the third section we outline the implementation details, which are based on application of the fully implicit Euler scheme. In addition we derive the expression for the consistent tangent modulus at a fiber, which is required to ensure quadratic rates of convergence for the global Newton method. Finally, we conclude this chapter with a section dedicated to the assessment of the proposed algorithm. Accuracy is demonstrated through iso-error maps for both perfect and hardening plasticity. Furthermore, we compare the performance in terms of total plastic step iterations required for i) the present formulation, ii) the general, three-dimensional and iii) the plane stress return mapping algorithm developed in [144]. Numerical examples pertaining to ultimate collapse load, cyclic loading, elastoplastic buckling and a pushover analysis of a steel frame are also included to highlight the capabilities of the Hybrid NLP element equipped with the multiaxial stress-update algorithm proposed herein.

### 3.2 Three-dimensional Constitutive Model

Below we outline the general form of rate-independent elastoplasticity with combined isotropic and linear kinematic hardening. Infinitesimal strains and associative plasticity are assumed, while the yield criterion is purposefully unspecified at this state in order to maintain generality.

Let  $\sigma$ ,  $\epsilon$  be the second order symmetric stress and strain tensors. We introduce the internal hardening variables  $\alpha$  and  $q$  which represent the second order back-stress tensor and the equivalent uniaxial yield stress respectively. The former is associated with kinematic hardening while the latter models isotropic hardening. Finally, consider the decomposition of the strain

tensor into elastic and plastic parts as follows:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{el} + \boldsymbol{\epsilon}^{pl} \quad (3.1)$$

Then the rate form of the constitutive equations for the assumptions mentioned above is the following:

$$\dot{\boldsymbol{\epsilon}} = \dot{\boldsymbol{\epsilon}}^{el} + \dot{\boldsymbol{\epsilon}}^{pl} \quad (\text{Strain decomposition}) \quad (3.2a)$$

$$\dot{\boldsymbol{\sigma}} = \mathbb{C}^{el} [\dot{\boldsymbol{\epsilon}} - \dot{\boldsymbol{\epsilon}}^{pl}] \quad (\text{Elastic constitutive law}) \quad (3.2b)$$

$$\dot{\boldsymbol{\epsilon}}^{pl} = \dot{\lambda} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \quad (\text{Flow rule}) \quad (3.2c)$$

$$\dot{\boldsymbol{\alpha}} = -\dot{\lambda} H_{kin} \frac{\partial \Phi}{\partial \boldsymbol{\alpha}} \quad (\text{Kinematic hardening law}) \quad (3.2d)$$

$$\dot{q} = \frac{\partial q}{\partial e^{pl}} \dot{e}^{pl} \quad (\text{Isotropic hardening law}) \quad (3.2e)$$

$$\dot{e}^{pl} = \dot{\lambda} \quad (\text{Equivalent plastic strain}) \quad (3.2f)$$

$$\Phi(\boldsymbol{\sigma}, \boldsymbol{\alpha}, e^{pl}) = \sigma_{eq}(\boldsymbol{\sigma}, \boldsymbol{\alpha}) - q(e^{pl}) \leq 0 \quad (\text{Yield criterion}) \quad (3.2g)$$

In the above system of differential-algebraic equations,  $\lambda$  is the plastic parameter,  $H_{kin}$  is the kinematic hardening modulus,  $\mathbb{C}^{el}$  is the elastic moduli, a fourth order tensor, of the material and  $\sigma_{eq}$  is the equivalent stress, which is determined by the yield criterion used. Furthermore, in the case of linear isotropic hardening with modulus  $H_{iso}$ , the corresponding harening law (Eq. 3.2e) becomes  $\dot{q} = H_{iso}\dot{e}^{pl}$ , where  $e^{pl}$  is the equivalent plastic strain. Finally, with the inequality in Eq. (3.2g) we define a feasible stress space such that stress points strictly satisfying the inequality,  $\Phi < 0$  represent an elastic state while these points that cause plastic flow render  $\Phi$  zero.

The so-called Karush-Kuhn-Tucker loading/unloading conditions derived from the underly-

ing constrained variational problem[145] can be stated as follows:

$$\Phi \leq 0, \quad \dot{\lambda} \geq 0, \quad \dot{\lambda}\Phi = 0 \quad (3.3)$$

From these conditions we can determine the current stress state:

- if  $\dot{\lambda} > 0$ , then  $\dot{\lambda}\Phi = 0$  implies  $\Phi = 0$  which means the state is plastic
- if  $\Phi < 0$ , then  $\dot{\lambda}\Phi = 0$  implies  $\dot{\lambda} = 0$  and from (3.2c) we get that the state is elastic.

Since  $\Phi = 0$ ,  $\dot{\lambda} > 0$  when plastic loading persists, the third equation in (3.3) leads to the so-called plastic consistency condition:

$$\dot{\lambda}\dot{\Phi} = 0 \quad (3.4)$$

### 3.3 The $J_2$ Formulation for Fiber Stress State

#### 3.3.1 Mapping on the Constrained Stress Space

We begin by defining the admissible stress space for a planar beam fiber,  $\mathcal{S}$ :

$$\mathcal{S} = \{\vec{\sigma} \in \mathbb{R}^6 \mid \sigma_{22} = \sigma_{33} = \sigma_{23} = \sigma_{13} = 0\}$$

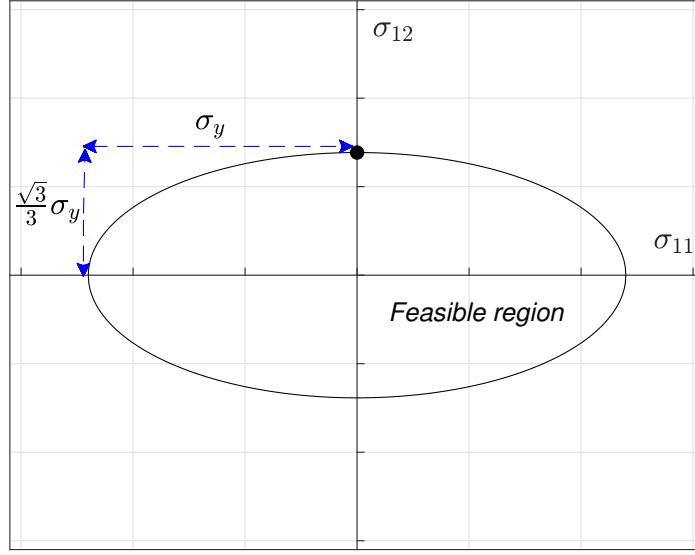
The von Mises yield criterion is stated as follows:

$$\Phi(J_2, e^{pl}) = \sqrt{3J_2} - q(e^{pl}) \leq 0 \quad (3.5)$$

where  $J_2$  is the second invariant of the deviatoric stress tensor:

$$J_2 = \frac{1}{2}\boldsymbol{\sigma}^d : \boldsymbol{\sigma}^d \quad (3.6)$$

The deviatoric part of any tensor is given by  $( )^d = ( ) - \frac{1}{3}\mathbf{I} \cdot \text{trace}[( )]$ , with  $\mathbf{I}$  being the identity



**Figure 3.1.** Feasible space and yield locus for the planar fiber constrained  $J_2$  model. At the ellipse boundary we have  $\sqrt{3}J_2 = q$ .

tensor and symbol ":" designates the contraction operator:  $\boldsymbol{\sigma}^d : \boldsymbol{\sigma}^d = \sum_{i,j} \sigma_{ij}^d \sigma_{ij}^d$ .

In the von Mises  $J_2$  framework, the hydrostatic pressure cannot cause plastic flow. In other words, plastic deformation is due to the deviatoric components of the stress tensor and volume changes are caused only by elastic deformations. This implies that  $\text{trace}[\boldsymbol{\epsilon}^{pl}] = 0$  which, coupled with Eq. (3.2d), also leads to  $\text{trace}[\boldsymbol{\alpha}^{pl}] = 0$ . This means that the back stress tensor in Eq. (3.2d) deviatoric.

In active stress space  $\mathcal{S}$  the yield function is  $\Phi = \sqrt{\sigma_{11}^2 + 3\sigma_{12}^2} - q$  describes an ellipse, shown in Fig. 3.1, with semi-major and semi-minor axes  $\sigma_y$ ,  $\sqrt{3}/3\sigma_y$ , respectively.

When kinematic hardening is included, we introduce the effective stress tensor,  $\boldsymbol{\zeta}$ , in order to facilitate notational simplicity. Its deviatoric part, which plays central role in the three-dimensional formulation of the  $J_2$  model is given by:

$$\boldsymbol{\zeta}^d = \boldsymbol{\sigma}^d - \boldsymbol{\alpha}^d \quad (3.7)$$

We now seek a mapping from the deviatoric stress space onto the fiber stress space  $S$  such that the inner product in the expression for  $J_2$  is preserved. For this purpose, and in line with

the vector notation for stress and strain tensors adopted earlier, we define the following vectors which involve only these tensor components that explicitly enter subsequent derivations:

$$\vec{\alpha}_f = \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \end{bmatrix}, \quad \vec{\alpha}_f^d = \begin{bmatrix} \alpha_{11}^d \\ \alpha_{12}^d \end{bmatrix}, \quad \vec{\epsilon}_f^{pl} = \begin{bmatrix} \epsilon_{11}^{pl} \\ \gamma_{12}^{pl} \end{bmatrix},$$

$$\vec{\sigma}_f^d = \begin{bmatrix} \sigma_{11}^d \\ \sigma_{12}^d \end{bmatrix}, \quad \vec{\zeta}_f = \begin{bmatrix} \zeta_{11} \\ \zeta_{12} \end{bmatrix}, \quad \vec{\zeta}_f^d = \begin{bmatrix} \zeta_{11}^d \\ \zeta_{12}^d \end{bmatrix}$$

The subscript  $f$  indicates that the vector involves only these tensor components pertinent to the fiber stress state. The non-zero components of  $\vec{\zeta}^d$  are  $\zeta_{11}^d, \zeta_{22}^d, \zeta_{33}^d, \zeta_{12}^d, \zeta_{21}^d$  and thus we can establish the following mappings between the active components of  $\vec{\zeta}^d$  and  $\vec{\zeta}_f$ :

$$\vec{\zeta}^d = \mathbf{Q} \vec{\zeta}_f, \quad \vec{\zeta}_f^d = \mathbf{S} \vec{\zeta}_f \quad (3.8)$$

where the mapping matrices above are given by:

$$\mathbf{Q} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T, \quad \mathbf{S} = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & 1 \end{bmatrix} \quad (3.9)$$

where the value of  $Q_{24} = 2$  because of the stress tensor symmetry.

The vector representation of (3.6) in terms of the effective stress is:

$$J_2 = \frac{1}{2} (\vec{\zeta}^d)^T \mathbf{J} \vec{\zeta}^d \quad (3.10)$$

where matrix  $\mathbf{J} = \text{diag}[1, 1, 1, 2, 2, 2]$  is introduced to account for the symmetry of  $\zeta^d$ . If the axisymmetric version of vectors is used, where only 4 tensor components are stored, then  $\mathbf{J} = \text{diag}[1, 1, 1, 2, ]$ .

We can now express  $J_2$  in terms of  $\vec{\zeta}_f$ :

$$J_2 = \frac{1}{2}(\vec{\zeta}^d)^T \mathbf{J} \vec{\zeta}^d = \frac{1}{2}\vec{\zeta}_f^T \mathbf{Q}^T \mathbf{J} \mathbf{Q} \vec{\zeta}_f = \frac{1}{2}\vec{\zeta}_f^T \mathbf{V} \vec{\zeta}_f = \frac{1}{2}\|\vec{\zeta}_f\|_{\mathbf{V}}^2 \quad (3.11)$$

where  $\mathbf{V} = \text{diag}[\frac{2}{3}, 2]$  and the notation used for the inner product induced by matrix  $\mathbf{A}$  between vectors  $\vec{w}$ ,  $\vec{v}$  is  $(\vec{w}, \vec{v})_{\mathbf{A}} = \vec{w}^T \mathbf{A} \vec{v}$ , which leads to the notation adopted in Eq. (3.11).

With these derivations at hand, system (3.2) can be recast in the constrained fiber stress state as follows:

$$\dot{\vec{\epsilon}}_f = \dot{\vec{\epsilon}}_f^{el} + \dot{\vec{\epsilon}}_f^{pl} \quad (\text{Strain decomposition}) \quad (3.12a)$$

$$\dot{\vec{\sigma}}_f = \mathbf{C}^{el} [\dot{\vec{\epsilon}}_f - \dot{\vec{\epsilon}}_f^{pl}] \quad (\text{Elastic constitutive law}) \quad (3.12b)$$

$$\dot{\vec{\epsilon}}_f^{pl} = \lambda \vec{n} \quad (\text{Flow rule}) \quad (3.12c)$$

$$\dot{\vec{\alpha}}_f = \lambda H_{kin} \mathbf{V}^{-1} \vec{n} \quad (\text{Kinematic hardening law}) \quad (3.12d)$$

$$\dot{q}_f = \lambda \frac{\partial q_f}{\partial e_f^{pl}} \quad (\text{Isotropic hardening law}) \quad (3.12e)$$

$$\Phi(\vec{\zeta}_f, e^{pl}) = \sqrt{\frac{3}{2}}\|\vec{\zeta}_f\|_{\mathbf{V}} - q_f(e_f^{pl}) \leq 0 \quad (\text{Yield criterion}) \quad (3.12f)$$

Vector  $\vec{n}$  is expressed as follows:

$$\vec{n} = \frac{\partial \Phi}{\partial \vec{\sigma}_f} = -\frac{\partial \Phi}{\partial \vec{\alpha}_f} = \sqrt{\frac{3}{2}} \frac{\mathbf{V} \vec{\zeta}_f}{\|\vec{\zeta}_f\|_{\mathbf{V}}} \quad (3.13)$$

Equations (3.2e), (3.2f) remain the same in the current system. In addition, note that for the  $J_2$  model, the left-hand side of eq. (3.2d) is a deviatoric tensor whereas in (3.12d) it has been mapped to fiber stress space  $\mathcal{S}$ .

### 3.3.2 Continuous Tangent Modulus

The continuous elastoplastic modulus  $\mathbf{C}^{ep}$  is found by enforcing the plastic consistency condition (3.4) during a plastic step:

$$\vec{\mathbf{n}}^T \dot{\boldsymbol{\zeta}}_f - \dot{\lambda} \frac{\partial q_f}{\partial e_f^{pl}} = 0 \quad (3.14)$$

The rate form for the effective stress is given by combining Eqs. (3.12b), (3.12d) with  $\dot{\boldsymbol{\zeta}}_f = \dot{\boldsymbol{\sigma}}_f - \dot{\boldsymbol{\alpha}}_f$ :

$$\dot{\boldsymbol{\zeta}}_f = \mathbf{C}^{el} \dot{\boldsymbol{\epsilon}}_f - \dot{\lambda} \mathbf{Z} \vec{\mathbf{n}} \quad (3.15)$$

where  $\mathbf{Z} = \mathbf{C}^{el} + H_{kin} \mathbf{V}^{-1}$ . Substituting (3.15) into (3.14) and solving for  $\dot{\lambda}$  we get:

$$\dot{\lambda} = \frac{\vec{\mathbf{n}}^T \mathbf{C}^{el} \dot{\boldsymbol{\epsilon}}_f}{\vec{\mathbf{n}}^T \mathbf{Z} \vec{\mathbf{n}} + \frac{\partial q_f}{\partial e_f^{pl}}} \quad (3.16)$$

Finally, by making use of (3.16), (3.12c) and (3.12b), we arrive at the expression for the continuous elastoplastic modulus:

$$\mathbf{C}^{pl} = \mathbf{C}^{el} - \frac{\vec{\mathbf{m}} \vec{\mathbf{m}}^T}{\vec{\mathbf{n}}^T \mathbf{Z} \vec{\mathbf{n}} + \frac{\partial q_f}{\partial e_f^{pl}}} \quad (3.17)$$

where  $\vec{\mathbf{m}} = \mathbf{C}^{el} \vec{\mathbf{n}}$ .

## 3.4 Implicit Integration of Rate Equations

We employ the fully implicit backward Euler method to discretize the rate constitutive equations in system (3.12) in the pseudotime interval  $[t_n, t_{n+1}]$ . We regard the incremental/iterative process as strain driven, in that at the start of increment  $n$ , the dependent state variables for each fiber,  $S_{f,n} = \{\vec{\boldsymbol{\epsilon}}_f, \vec{\boldsymbol{\epsilon}}_f^{pl}, \vec{\boldsymbol{\alpha}}_f, e_f^{pl}\}_n$ ,

are known and stored. We also know the increment in the centerline strain vector  $\vec{\mathbf{d}}, \Delta \vec{\mathbf{d}}$ .

The incremental strain vector for the current fiber is then determined using eq. (2.53b):

$$\Delta \vec{\boldsymbol{\epsilon}}_f = \mathbf{N}_s \Delta \vec{\mathbf{q}}$$

Given the incremental strain vector  $\Delta\vec{\epsilon}_f$ , we need to update the state variables for  $t_{n+1}$ :  $S_{f,n} \rightarrow S_{f,n+1}$ . In what follows we omit the subscript "f" which indicated that the involved measure pertains to the fiber state:

$$S_{f,n} \equiv S_n = \{\vec{\epsilon}_n, \vec{\epsilon}_n^{pl}, \vec{\alpha}_n, e_n^{pl}\}, \quad S_{f,n+1} \equiv S_{n+1} = \{\vec{\epsilon}_{n+1}, \vec{\epsilon}_{n+1}^{pl}, \vec{\alpha}_{n+1}, e_{n+1}^{pl}\},$$

The update of the fiber total strain vector  $\vec{\epsilon}$  is trivial:  $\vec{\epsilon}_{n+1} = \vec{\epsilon}_n + \Delta\vec{\epsilon}$ . For the remaining state variables the rate system (3.12) is discretized as follows:

$$\vec{\sigma}_{n+1} = \mathbf{C}^{el} \left[ \vec{\epsilon}_{n+1} - \vec{\epsilon}_{n+1}^{pl} \right] \quad (3.18a)$$

$$\vec{\epsilon}_{n+1}^{pl} = \vec{\epsilon}_n^{pl} + \dot{\lambda} \vec{n}_{n+1} \quad (3.18b)$$

$$\vec{\alpha}_{n+1} = \vec{\alpha}_n + \lambda H_{kin} \mathbf{V}^{-1} \vec{n}_{n+1} \quad (3.18c)$$

$$q_{n+1} = q_n + \lambda \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}} \quad (3.18d)$$

$$\Phi(\vec{\zeta}_{n+1}, e_{n+1}^{pl}) = \sqrt{\frac{3}{2}} \|\vec{\zeta}_{n+1}\|_{\mathbf{V}} - q_{n+1}(e_{n+1}^{pl}) \leq 0 \quad (3.18e)$$

and

$$\vec{\zeta}_{n+1} = \vec{\sigma}_{n+1} - \vec{\alpha}_{n+1} \quad (3.19)$$

### 3.4.1 Elastic Predictor - Plastic Corrector algorithm

It is convenient to recast system (3.18) purely in terms of stress by utilizing (3.19),(3.18a),(3.18c):

#### Stress rate system

$$\dot{\vec{\zeta}} = \mathbf{C}^{el} \dot{\vec{\epsilon}} - \dot{\lambda} \mathbf{Z} \vec{n} \quad (3.20a)$$

$$\dot{q} = \dot{\lambda} \frac{\partial q}{\partial e^{pl}} \quad (3.20b)$$

#### Discretized system

$$\vec{\zeta}_{n+1} = \vec{\zeta}_n + \mathbf{C}^{el} \Delta \vec{\epsilon}_{n+1} - \lambda \mathbf{Z} \vec{n}_{n+1} \quad (3.21a)$$

$$q_{n+1} = q_n + \lambda \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}} \quad (3.21b)$$

We need not differentiate between initial and final value for the plastic parameter  $\lambda$  since it is initialized to zero at the beginning of every step:  $\lambda_n = 0$ . The solution of (3.21) is based

on a two-step procedure. The first phase is called trial elastic step, where we assume that the increment  $\Delta\vec{\epsilon}$  is entirely elastic and all state variables are updated by setting  $\lambda = 0$ :  $S_n \rightarrow S_{n+1}^{TR}$ . If the trial stresses satisfy the yield condition  $\Phi_{n+1}^{TR} < 0$ , then the step was indeed elastic and we can set  $S_{n+1}^{TR} \rightarrow S_{n+1}$ . If the yield criterion is violated,  $\Phi_{n+1}^{TR} > 0$ , then the step is plastic and we need to apply (plastic) corrections to the state variables so that, upon convergence, we have achieved  $\Phi_{n+1} = 0$ . This procedure is referred to as Elastic Predictor - Plastic Corrector algorithm and can be traced back to the work of Wilkins[146]. It has enjoyed widespread application in solving problems in elastoplasticity[54, 147, 148, 149, 150, 151] and the basic relations for each step are summarized below.

<b>Elastic Predictor</b>	<b>Plastic Corrector</b>
$\vec{\zeta}_{n+1}^{TR} = \mathbf{C}^{el} [\vec{\epsilon}_{n+1} - \vec{\epsilon}_n^{pl}] - \vec{\alpha}_n$	$\vec{\zeta}_{n+1} = \vec{\zeta}_{n+1}^{TR} - \lambda \mathbf{Z} \vec{n}_{n+1}$ (3.23a)
$q_{n+1}^{TR} = q_n$	$\xrightarrow{\Phi_{n+1}^{TR} > 0}$ $q_{n+1} = q_{n+1}^{TR} + \lambda \frac{\partial q_{n+1}}{\partial \epsilon_{n+1}^{pl}}$ (3.23b)
$\Phi_{n+1}^{TR} = f(\vec{\zeta}_{n+1}^{TR}, q_{n+1}^{TR})$	$\Phi_{n+1} = 0$ (3.23c)

For the planar fiber formulation, the Plastic Corrector step is a system of four nonlinear algebraic equations with four unknowns,  $\{\zeta_{11,n+1}, \zeta_{12,n+1}, q_{n+1}, \lambda\}$ , and generally a local Newton method is required to determine find a solution. It should be stressed here that, in multiaxial constitutive algorithms that do not make use of the mapping as developed here, this local process is nested within an upper level Newton procedure that enforces the zero transverse stress condition.

Substitution of Eqs. (3.23a),(3.23b) into (3.23c) furnishes a nonlinear scalar equation that needs to be solved for  $\lambda$ . Once  $\lambda$  is computed, then the fiber state can be updated  $S_{n+1}^{TR} \rightarrow S_{n+1}$ . Refer also to Box 1 for a detailed summary of the plastic updates. Detailed expressions for the local Newton method on  $\Phi_{n+1}(\lambda) = 0$  are provided in Appendix D.

### 3.4.2 Consistent Tangent Modulus

The tangent modulus consistent with the numerical discretization used on system (3.12) is necessary to restore the quadratic rates of convergence for the global Newton method[152]. That is:

$$\mathbf{C}_c^{pl} = \frac{d\vec{\sigma}_{n+1}}{d\vec{\epsilon}_{n+1}} \quad (3.24)$$

We take the differentials of (3.18a), (3.18c) and (3.18e), using the fact that  $d\vec{\zeta}_{n+1} = d\vec{\sigma}_{n+1} - d\vec{\alpha}_{n+1}$  from (3.19) and arrive at the following system:

$$\begin{bmatrix} \mathbf{C}^{el} + \lambda \Psi_{n+1} & -\lambda \Psi_{n+1} \\ -\lambda \Psi_{n+1} & H_{kin}^{-1} \mathbf{V} + \lambda \Psi_{n+1} \end{bmatrix} \begin{bmatrix} d\vec{\sigma}_{n+1} \\ d\vec{\alpha}_{n+1} \end{bmatrix} = \begin{bmatrix} d\vec{\epsilon}_{n+1} - d\lambda \vec{n}_{n+1} \\ d\lambda \vec{n}_{n+1} \end{bmatrix} \quad (3.25a)$$

$$\vec{n}_{n+1}^T [d\vec{\sigma}_{n+1} - d\vec{\alpha}_{n+1}] = d\lambda \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}} \quad (3.25b)$$

where  $\Psi$  is given as follows:

$$\Psi = \frac{\partial \vec{n}}{\partial \vec{\zeta}} = \frac{\partial^2 f}{\partial \vec{\zeta}^2} = \frac{1}{\|\vec{\zeta}\|_V} [\mathbf{V} - \vec{n} \vec{n}^T] \quad (3.26)$$

Upon solving (3.25a) for  $d\vec{\sigma}_{n+1}$ ,  $d\vec{\alpha}_{n+1}$  we substitute them into (3.25b) and solve for  $d\lambda$ :

$$d\lambda = \frac{\vec{n}_{n+1}^T [\Xi_{11} + \Xi_{12}^T] d\vec{\epsilon}_{n+1}}{\vec{n}_{n+1}^T \mathbf{T} \vec{n}_{n+1} + \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}}} \quad (3.27)$$

Finally, substituting  $d\lambda$  into (3.18a) furnishes the formula for the consistent tangent modulus:

$$\mathbf{C}_c^{pl} = \Xi_{11} - \frac{\vec{N}_{n+1} \vec{N}_{n+1}^T}{1 + \delta_{n+1}} \quad (3.28)$$

Details regarding all newly defined vectors and matrices involved in the derivation of  $\mathbf{C}_c^{pl}$  is given in Appendix E. In Box 1 we have summarised the return mapping algorithnm for a

cross-section fiber of a planar problem.

### 3.4.3 Consistent Section Stiffness

Once the stress update on a particular fiber has been carried out, its contribution to the stress resultant vector and the section stiffness are accounted for via a midpoint integration rule along the section height (see Eqs. (2.56), (2.58)).

**Box 1 : Summary of Elastic Predictor - Plastic Corrector algorithm.**

- **Start of step - Elastic Predictor**

$$\begin{array}{ll}
 \blacksquare \quad \vec{\epsilon}_{n+1}^{pl,TR} = \vec{\epsilon}_n^{pl} & \blacksquare \quad e_{n+1}^{pl,TR} = e_n^{pl} \\
 \blacksquare \quad \vec{\sigma}_{n+1}^{TR} = \mathbf{C}^{el} [\vec{\epsilon}_{n+1} - \vec{\epsilon}_n^{pl}] & \blacksquare \quad q_{n+1}^{TR} = q_n \\
 \blacksquare \quad \vec{\alpha}_{n+1}^{TR} = \vec{\alpha}_n & \blacksquare \quad f_{n+1}^{TR} = \Phi(\vec{\zeta}_{n+1}^{TR}, q_{n+1}^{TR})
 \end{array}$$

- IF  $\Phi_{n+1}^{TR} < 0 \rightarrow \text{Elastic Step}$

$$\begin{array}{ll}
 \blacksquare \quad \vec{\epsilon}_{n+1}^{pl} = \vec{\epsilon}_{n+1}^{pl,TR} & \blacksquare \quad e_{n+1}^{pl} = e_{n+1}^{pl,TR} \\
 \blacksquare \quad \vec{\sigma}_{n+1} = \vec{\sigma}_{n+1}^{TR} & \blacksquare \quad q_{n+1} = q_{n+1}^{TR} \\
 \blacksquare \quad \vec{\alpha}_{n+1} = \vec{\alpha}_{n+1}^{TR} & \blacksquare \quad \mathbf{C} = \mathbf{C}^{el}
 \end{array}$$

- ELSE  $f_{n+1}^{TR} > 0 \rightarrow \text{Plastic Corrector}$

1. Iteratively solve (3.4.1) for  $\lambda$ :  $\lambda^{j+1} = \lambda^j - \frac{\Phi_{n+1}^j}{(\frac{d\Phi_{n+1}}{d\lambda})^j}$ , with  $\lambda^0 = 0$ ,  $\Phi_{n+1}^0 = \Phi_{n+1}^{TR}$ . Loop termination:  $|\Phi_{n+1}^{j+1}| \leq e_{tol}$ , where  $e_{tol}$  small.

2. Update state variables:

$$\begin{array}{ll}
 \blacksquare \quad \vec{\zeta}_{n+1} = \Omega(\lambda) \vec{\zeta}_{n+1}^{TR} & \blacksquare \quad \vec{\sigma}_{n+1} = \vec{\zeta}_{n+1} + \vec{\alpha}_{n+1} \\
 \blacksquare \quad \vec{\epsilon}_{n+1}^{pl} = \vec{\epsilon}_n^{pl} + \lambda \vec{n}_{n+1} & \blacksquare \quad e_{n+1}^{pl} = e_n^{pl} + \lambda \\
 \blacksquare \quad \vec{\alpha}_{n+1} = \vec{\alpha}_n + \lambda H_{kin} \mathbf{V}^{-1} \vec{n}_{n+1} & \blacksquare \quad q_{n+1} = q_n + \lambda \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}}
 \end{array}$$

3. Get consistent tangent modulus

$$\mathbf{C} = \mathbf{C}_c^{pl} = \Xi_{11} - \frac{\vec{N}_{n+1} \vec{N}_{n+1}^T}{1 + \delta_{n+1}}$$

See Appendix D for  $\Omega(\lambda)$ .

Notice that while in the present formulation  $C_{21} = C_{12}$ , this will not be the case if a generalized integration rule is used for the rate equations. In such a scenario, if enforcement of the plastic consistency does not correspond to the interior time  $t_a \in [t_n, t_{n+1}]$  chosen, that is  $\Phi_a = 0$ , then the consistent tangent will not be symmetric. For more details on this refer to [153].

## 3.5 Numerical Examples

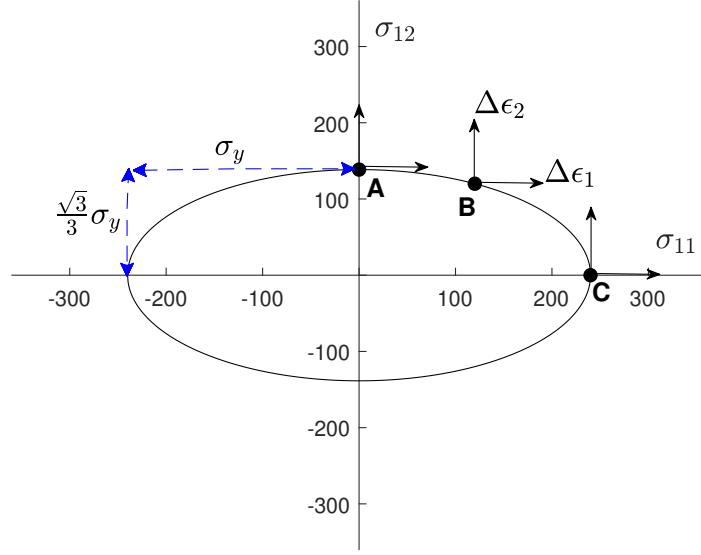
### 3.5.1 Accuracy of the Proposed Algorithm

Iso-error maps are commonly used to test the accuracy of return-mapping algorithms in elasto-plasticity[55]. The contours are generated by imposing a family of stress or strain prescribed loadings from a selected point on the yield surface which cause further plastification (no unloading). For each load incrementation, the error is computed as the relative difference between the stress predicted by the proposed algorithm and “exact” stress. The exact stress is typically found by imposing the same increment in a number of sequential subincrements,  $n_s$ . A convergence analysis is required to ensure that subincrementation indeed converges asymptotically to a stress point. In the present study it was found that for  $n_s \geq 500$  the stress components had achieved convergence with respect to the first 7 decimal points. Therefore, we use  $n_s = 500$ .

The error plotted is given by:

$$err = \frac{\|\vec{\sigma}^* - \vec{\sigma}\|}{\|\vec{\sigma}^*\|} \cdot 100 \quad (3.29)$$

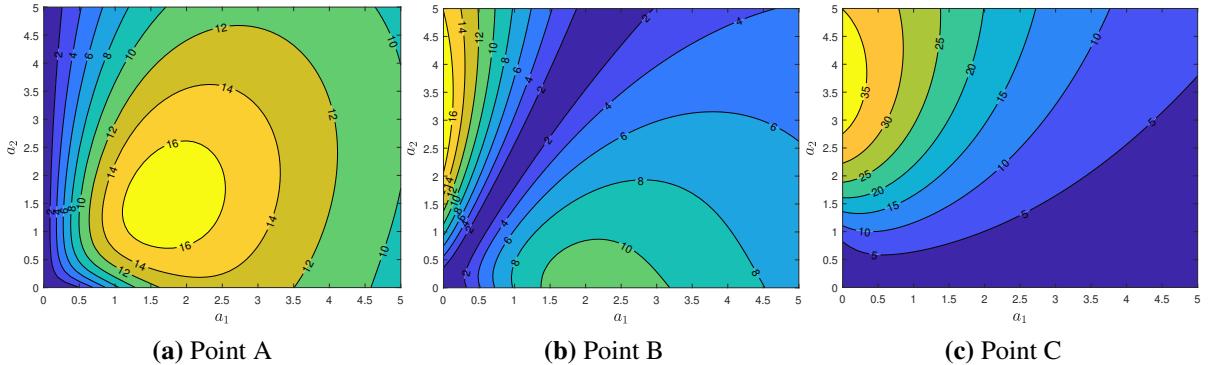
where  $\vec{\sigma}^*$  is the exact stress vector and  $\vec{\sigma}$  the one predicted by the proposed scheme by solving the return mapping problem in one increment. The three points on the yield surface that are selected and shown in Fig. 3.2 correspond to an initial state of pure shear (A), mixed state where  $\sigma_{11,0} = \sigma_{12,0}$  (B) and a pure uniaxial stress state (C). For each of these initial states the family of strain increments is imposed along the directions shown, with range up to  $\Delta\|\vec{\epsilon}\| = 5\epsilon_y$ , where  $\epsilon_y$  is the yield strain in tension/compression. The material data used are:  $E = 2.1 \times 10^5$ ,  $\sigma_y = 2.4 \times 10^2$  and Poisson’s ratio  $\nu = 0.3$ . For the hardening cases, linear isotropic and kinematic



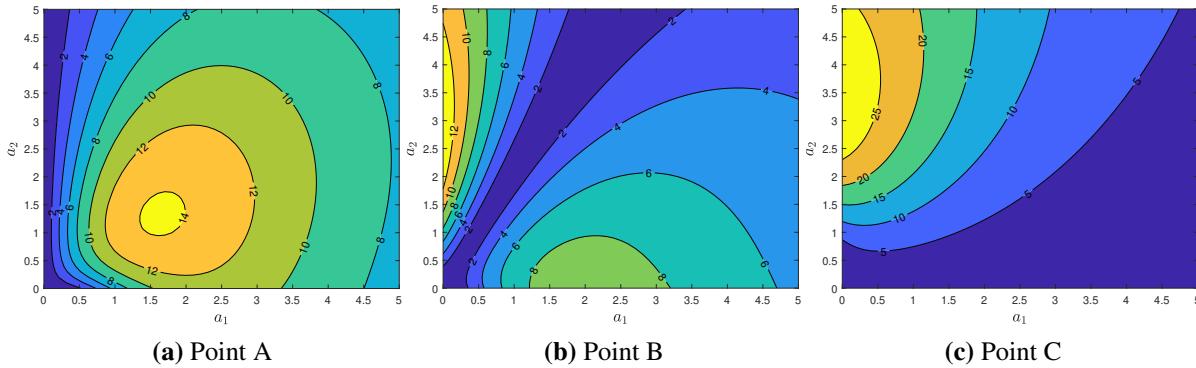
**Figure 3.2.** Points of interest on yield surface and direction of load increments.

moduli were prescribed as follows:  $H_{iso} = H_{kin} = 10^4$ .

The parameters associated with the horizontal and vertical axes,  $a_1$ ,  $a_2$ , are defined in the following way:  $a_1 = \Delta\vec{\epsilon}_1/\epsilon_y$ ,  $a_2 = \Delta\vec{\epsilon}_2/\epsilon_y$ . As can be seen from figures 3.3, 3.4, the relative error remains small even for quite large increments in the strain vector. Note also that along the semi-major and semi-minor axes, which correspond to the cases of pure uniaxial and pure shear states respectively, the return mapping gives the exact stress even without subincrementation. The same is true for point B when the strain increments are along a direction around  $68^\circ$  with respect to  $a_1$  axis.



**Figure 3.3.** Iso-error maps for a perfectly plastic material.

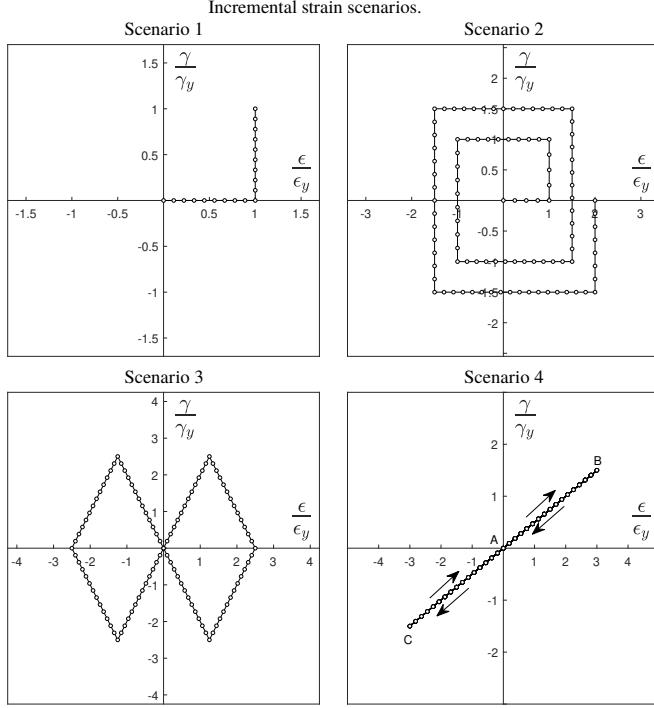


**Figure 3.4.** Iso-error maps for combined linear isotropic and kinematic hardening.

### 3.5.2 Performance Against Non-Monotonic Strain Histories

In this benchmark we investigate the performance of the proposed algorithm against the three dimensional/axisymmetric procedure typically implemented in beam elements when shear-flexure interaction is taken into account using a multiaxial law at the fiber level. We also compare it with the plane stress projection algorithm by Simo[152], which we followed in our derivations. In the former approach, the return mapping using implicit integration results in the so-called radial return, which doesn't require iterations in the case of  $J_2$  with perfect plasticity or linear hardening[146]. In contrast, the latter requires iterations since the topology of the stress space is not as simple as the von Mises cylinder anymore. We show below the drastic performance boost when using the present algorithm. As stated earlier, this is mainly due to the fact that we avoid an additional Newton procedure to enforce the zero transverse stress condition. The results for each scenario are shown in the tables below. A perfectly plastic material was assumed and both the average execution time  $t(\text{sec})$  and the total number of iterations,  $N_i$  are considered as performance metrics. The *additional* iterations during purely elastic steps are also included in parentheses for the 3D and Simo's plane stress algorithm. Moreover, for each scenario, four different incrementations of the strain history are employed, shown in Fig. 3.5, and the effect of this scaling on the performance is highlighted. Parameter  $a$  determines the increment "length":  $\alpha = \|\Delta\vec{\epsilon}\|/\epsilon_y$ . Material data are the same as in the previous example.

As can be seen from the Tables 3.1-3.4, the proposed scheme offers significant speed-up



**Figure 3.5.** The four different strain histories imposed, with the unstresses configuration as initial point.

compared to the other two methods. It should be highlighted that the 3D algorithm outperforms the one proposed by Simo because a perfectly plastic  $J_2$  was used. In general, it too would require internal iterations to determine  $\lambda$ , in addition to the outer level Newton procedure to enforce the stress constraints. For Scenario 4, shown in Table 3.4, both the 3D and plane stress algorithms did [440, 188, 84, 62] elastic iterations corresponding to values of parameter  $\alpha$ : [0.10, 0.25, 0.40, 0.50].

### 3.5.3 Plastic Collapse of a Doubly Clamped Beam

In this example we investigate the collapse load of a doubly clamped beam, shown in Fig. 3.6, for two slenderness ratios,  $L/h$ : 20 and 4, where  $h$  is the cross-section height. We conduct analysis using both uncoupled and coupled, multiaxial, constitutive models at the fiber level and compare result, for each case, with the lower bound collapse load as predicted by limit analysis.

In addition, we showcase the influence of numerical quadrature in predicting the analytical

**Table 3.1** Results for Scenario 1, perfect plasticity.

Algorithm	Exact $(\sigma_{11}, \sigma_{12})$	$\alpha = 0.10$		$\alpha = 0.25$		$\alpha = 0.40$		$\alpha = 0.50$	
		$t^a$	$N_i$	$t$	$N_i$	$t$	$N_i$	$t$	$N_i$
Proposed	(1.617,1.023)	0.57	29	0.44	13	0.40	8	0.38	5
3D		1.40	46(20)	0.83	20(8)	0.65	11(4)	0.56	6(2)
Simo		1.64	165(19)	1.13	88(7)	0.96	64(3)	0.83	55(1)
Error(%) <sup>b</sup>			1.32		3.00		5.16		8.05

<sup>a</sup> Measured in  $\times 10^{-3}$  sec.<sup>b</sup> Relative error in converged stress per increment scenario.**Table 3.2** Results for Scenario 2, perfect plasticity.

Algorithm	Exact $(\sigma_{11}, \sigma_{12})$	$\alpha = 0.10$		$\alpha = 0.25$		$\alpha = 0.40$		$\alpha = 0.50$	
		$t^a$	$N_i$	$t$	$N_i$	$t$	$N_i$	$t$	$N_i$
Proposed	(1.173,1.208)	0.33	524	0.16	234	0.10	144	0.08	108
3D		1.24	849(97)	0.61	394(40)	0.31	196(17)	0.23	143(11)
Simo		2.09	2791(96)	1.17	1363(39)	0.57	790(15)	0.42	596(10)
Error(%)			2.32		5.00		10.10		13.35

<sup>a</sup> Numbers in execution time columns are measured in  $\times 10^{-2}$  s.**Table 3.3** Results for Scenario 3, perfect plasticity.

Algorithm	Exact $(\sigma_{11}, \sigma_{12})$	$\alpha = 0.10$		$\alpha = 0.25$		$\alpha = 0.40$		$\alpha = 0.50$	
		$t^a$	$N_i$	$t$	$N_i$	$t$	$N_i$	$t$	$N_i$
Proposed	(1.628,1.018)	0.35	546	0.16	237	0.11	147	0.09	126
3D		1.15	920(115)	0.61	405(50)	0.31	194(22)	0.29	167(16)
Simo		2.20	2711(114)	1.42	1307(50)	0.55	789(20)	0.48	685(14)
Error(%)			1.15		2.51		5.40		5.80

<sup>a</sup> Numbers in execution time columns are measured in  $\times 10^{-2}$  s.**Table 3.4** Results for Scenario 4, five cycles, perfect plasticity. One cycle: A-B-C-A.

Algorithm	Exact $(\sigma_{11}, \sigma_{12})$	$\alpha = 0.10$		$\alpha = 0.25$		$\alpha = 0.40$		$\alpha = 0.50$	
		$t^a$	$N_i$	$t$	$N_i$	$t$	$N_i$	$t$	$N_i$
Proposed	(2.196,0.559)	0.92	1500	0.44	678	0.22	392	0.19	345
3D		3.50	2500	1.61	1130	0.77	490	0.69	445
Simo		5.20	7497	2.93	4519	1.60	2352	1.38	2122
Error(%)			0.19		0.40		0.84		0.91

<sup>a</sup> Numbers in execution time columns are measured in  $\times 10^{-2}$  s.

solution in the slender case. More specifically, we compare Gauss-Legendre and Gauss-Lobatto quadratures. The quadrature rule used is crucial for both flexibility-based beam elements and the hybrid one presented and used herein, since, in both formulations, accuracy relies more on the precision of integration rather than mesh density. Lastly, each cross-section is discretized into 15 layers.

The shear strain distribution function,  $k_q\varphi(X_2)$ , is assumed to be parabolic and defined as follows:

$$\varphi(X_2) = 1 - 4 \frac{X_2^2}{h^2} \quad (3.30)$$

where  $k_q$  is a parameter introduced in order to establish shear strain energy equivalence between assumed and exact distributions. For that matter, it is convenient to consider the uniform shear strain-average force hypothesis coupled with the corresponding shear coefficient as a representation of the exact one, assuming elastic behavior:

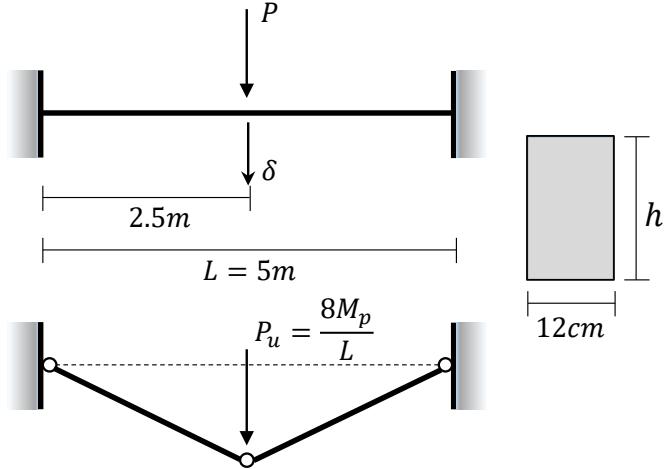
$$\mathcal{U}_{\text{shear}}^{\text{exact}} = k_s \mathcal{U}_{\text{shear}}^{\text{unif}} = \mathcal{U}_{\text{shear}}^{\text{quad}} \iff \frac{1}{2} k_s G A \gamma^2 = \frac{1}{2} G k_q^2 \left( \int_A \varphi(X_2)^2 dA \right) \gamma^2 \quad (3.31)$$

where  $k_s$  is the shear coefficient for uniform shear strain assumption. For rectangular sections with  $\nu = 0.3$ ,  $k_s = 0.886$  and solving eq. (3.31) for  $k_q$  we get:

$$k_q = \sqrt{\frac{k_s A}{\int_A \varphi(X_2)^2 dA}} \quad (3.32)$$

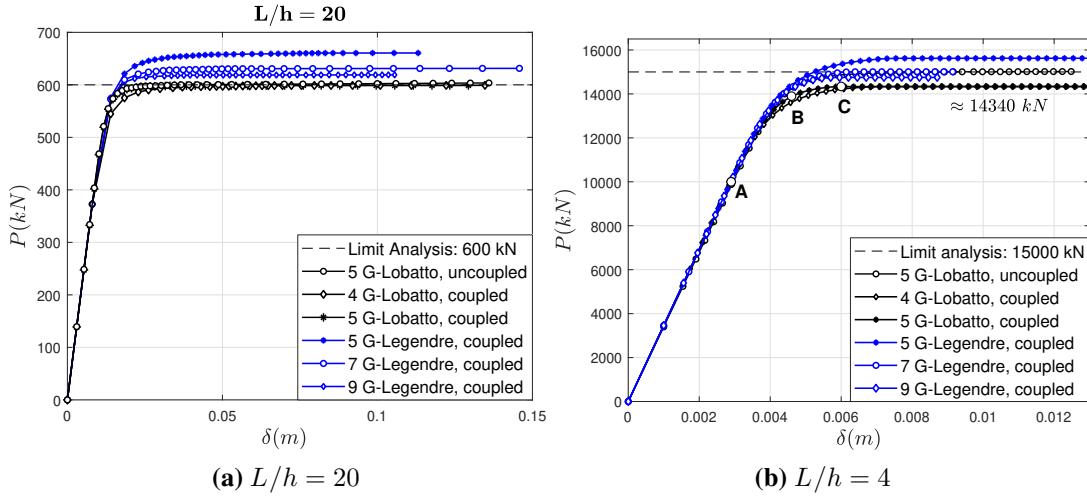
A first observation from Fig. 3.7 is that Gauss-Legendre quadrature tends to overestimate the capacity load on both edges of the slenderness spectrum. Gauss-Lobatto quadrature, in contrast, is more accurate and converges to the “exact” capacity from “below”. In addition, it provides accuracy with just five quadrature points whereas even with nine points, Gauss-Legendre still is not as accurate and converges from “above”. The results are in agreement with the ones reported by Papachristidis et al.[48] in a similar investigation.

When it comes to the influence of slenderness in the plastic capacity, the collapse load for the



**Figure 3.6.** Geometry, initial and final configuration of clamped beam, along with cross-section dimensions.

$L/h = 4$  case predicted by the multiaxial constitutive law is lower than the one based on limit analysis. Interestingly, the uncoupled constitutive law results in a purely flexural collapse mode, thus yielding a ultimate load coinciding with the theoretical from limit analysis. When the beam is slender, both coupled and uncoupled constitutive laws result, expectably, in a flexure-dominant mode. The results are summarized in Table 3.5 and Fig. 3.7. In Fig. 3.8 the stress distributions for  $\sigma_{11}$ ,  $\sigma_{12}$  and the von Mises stress,  $\sigma_{VM} = \sqrt{3J_2}$  are presented at three different load levels indicated by points A, B and C (see Fig. 3.7b). The plastic zone propagation is also exhibited. While the von Mises and P.Z profiles resemble plastification profiles typical of flexure-dominant members, we observe that the lower capacity captured by the multiaxial model in the  $L/h = 4$  case, which Fig. 3.8 pertains to, is due to shear plastification of the shrinking elastic core, a phenomenon which is neglected in a purely uniaxial formulation.



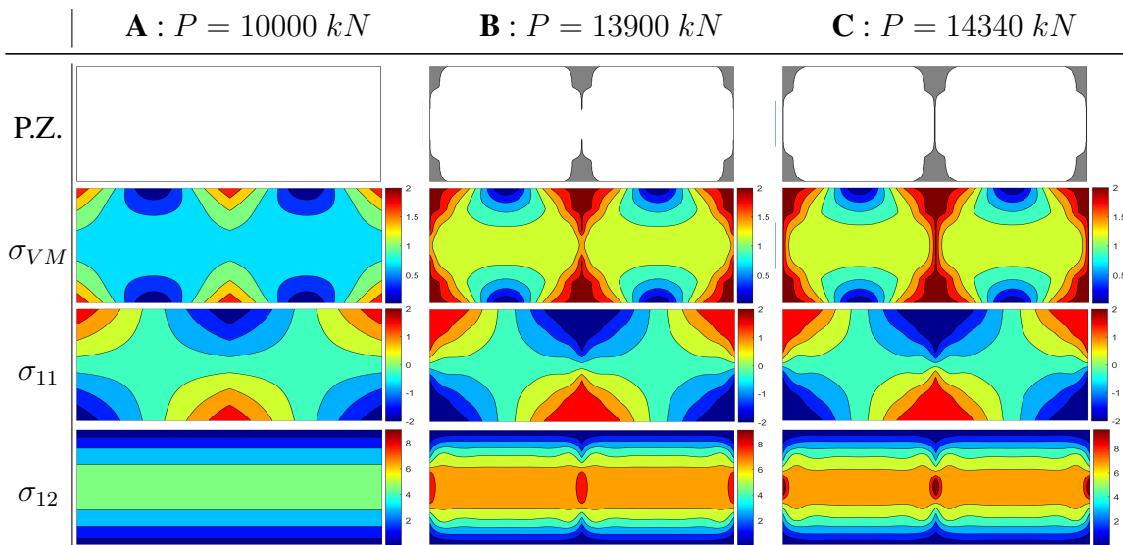
**Figure 3.7.** Load-deflection plots for different slenderness ratios. Influence of quadrature type and number of points.

**Table 3.5** Collapse moments and loads for clamped beam.

Collapse load, $P_u(kN)^a$				
$L/h$	$M_p(kN \cdot m)$	Limit Analysis	Uncoupled	Coupled
20	$3.75 \cdot 10^2$	$6.00 \cdot 10^2$	$\approx 6.00 \cdot 10^2$	$\approx 6.00 \cdot 10^2$
4	$9.37 \cdot 10^3$	$1.50 \cdot 10^4$	$\approx 1.50 \cdot 10^4$	$\approx 1.43 \cdot 10^4$

Material data:  $E = 200 \text{ GPa}$ ,  $\sigma_y = 200 \text{ MPa}$ .

<sup>a</sup> For coupled and uncoupled loads, 5 Gauss-Lobatto points are assumed.



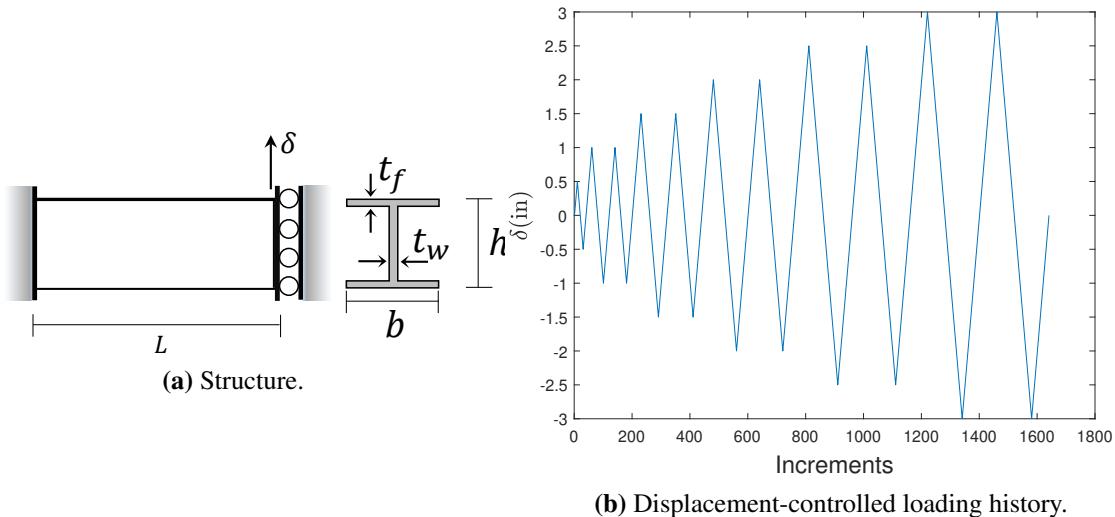
**Figure 3.8.** Case  $L/h = 4$ . Spread of Plastic Zone (P.Z.) and stress distribution for von Mises, axial and shear stress at three different load levels.

### 3.5.4 Cyclic Loading of Shear Link

In this example we are examining the response of a metallic shear link (Fig.3.9a) with a symmetric wide flange cross-section under displacement-controlled cyclic loading. The specimen we are concerned with was part of an experimental investigation by Hjelmstad & Popov[154] and later was numerically analyzed by Saritas & Filippou[49] using a mixed beam element formulation with shear-flexure interaction at the fiber level. In our analysys the specimen was modelled using one hybrid element with five Gauss-Lobatto points and 21 layers at each section with three layers on each flange.

The specimen length is  $L = 28$  in. and the cross-section type is W 18 × 40 with basic dimensions specified as follows: total height  $h = 17.88$  in., flange width  $b = 5.985$  in., web and flange thickness,  $t_w = 0.314$  in. and  $t_f = 0.521$  in. respectively. The material properties are summarized in Table 3.6, while the displacement history, in accordance with Hjelmstad & Popov's experiment, is shown in Fig.3.9b.

As can be seen in Fig. 3.10, satisfactory agreement with the experimental results is achieved when setting  $H_{kin} = 78.4$  kips/in<sup>2</sup> and  $H_{iso} = 8.40$  kips/in<sup>2</sup> and assuming a quadratic shear strain distribution along the cross-section height, in accordance with the previous example. More specifically, it was found that better agreement with the experimental results was achieved by slightly increasing the shear factor  $k_q$  to 1.62 from the default value of 1.33, as determined by eq. (3.32). The main cause for the discrepancy is that with the procedure outlined in Example 3 for the determination of the equivalent “quadratic” shear coefficient, an assumption for the shear stress distribution is not considered and this can be a source of non-negligible inaccuracies when open thin-walled sections are used. Nevertheless, since neither the distribution of shear fields on the cross-section shapes nor the particularization of flow and hardening laws for cyclic loading are the main focus of this study, some inaccuracies during cyclic plastification are to be expected and tolerated. In addition, a slight divergence during the unloading phases can be observed between the numerical and experimental tests. As explained in [49], this is due to improper restraint of the specimen during the experiment.

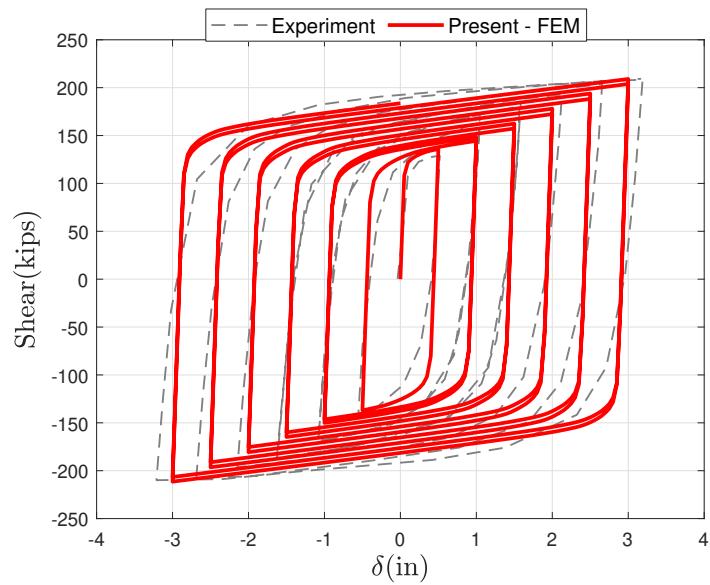


**Figure 3.9.** Geometry of the specimen and displacement-controlled loading history.

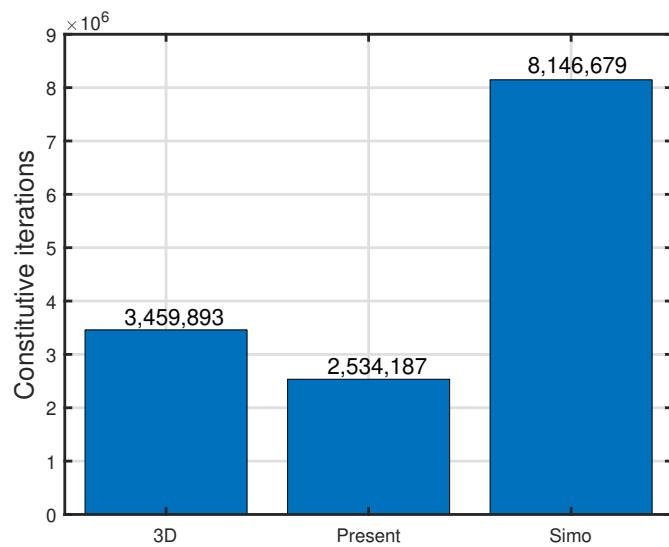
Performance analysis of the three different return-mapping schemes is also conducted for this example, where now we will examine the total number of constitutive iterations during the whole history of the imposed displacement. This helps highlight the significant acceleration the proposed algorithm offers during a more complex analysis that involves nonlinear response of an element comprised of multiple sections, each of which is discretized in several layers. The results when we use the three-dimensional/axisymmetric, Simo's plane stress and the proposed algorithm are shown in Fig. 3.11. As mentioned earlier, the linear hardening constitutive model even tends to favor the three-dimensional algorithm since the plastic correction reduces to the so-called radial return. However, in a more general setting where nonlinear hardening or a different material model is used, the performance differentials between the proposed scheme and its three-dimensional counterpart would be even more pronounced.

**Table 3.6** Material properties for Example 3.5.4.

Section	$\sigma_y$ (kips/in <sup>2</sup> )	$\sigma_u$ (kips/in <sup>2</sup> )	$\epsilon_y$ (%)	$\epsilon_u$ (%)	$E$ (kips/in <sup>2</sup> )	$H_{kin}$ (kips/in <sup>2</sup> )
W 18 × 40	39.5	60.1	1.8	22	28,300	102
Web	35.0	58.5	1.4	24	28,000	102



**Figure 3.10.** Experimental and numerical results for the specimen.



**Figure 3.11.** Performance of the three different return-mapping algorithms applied for the section state determination of the shear link under the imposed displacement history.

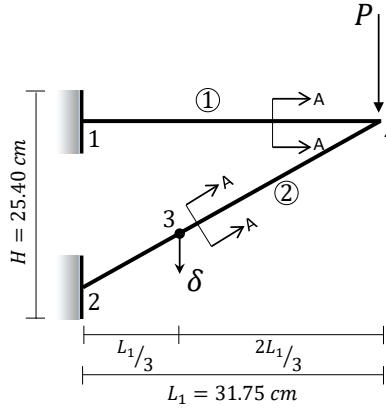
**Table 3.7** Member geometric and material data.

Member	$E(\frac{\text{kN}}{\text{cm}^2})$	$H_{kin}(\frac{\text{kN}}{\text{cm}^2})$	$\sigma_y(\frac{\text{kN}}{\text{cm}^2})$	$h_s(\text{cm})$	$h_t(\text{cm})$	$b(\text{cm})$
(1)	$6.86 \times 10^3$	$2.84 \times 10^2$	26.20	1.91	8.00	2.54
(2)	$6.86 \times 10^3$	$2.84 \times 10^2$	33.10	1.91	8.00	2.54

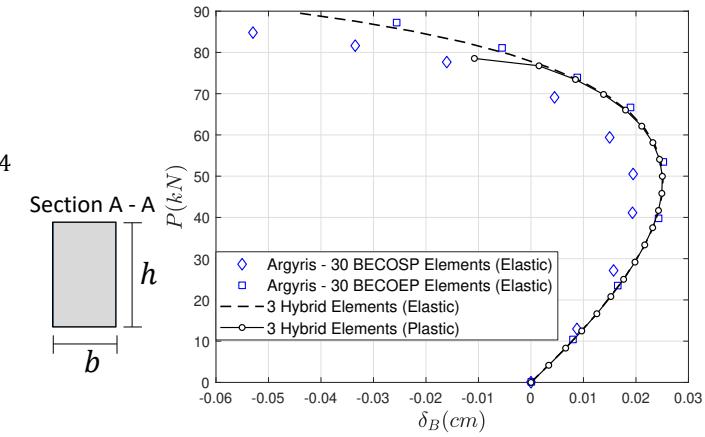
### 3.5.5 Elastoplastic Post-Buckling Behavior of Two-Beam Structure

This problem has been previously analyzed in [155, 156]. The structure, shown in Fig. 3.12a, enters its inelastic regime immediately after the buckling load is reached, whereby point B starts moving inwards. In Fig. 3.12b we compare the results we get using just three hybrid elements and five Gauss-Lobatto points in each element with the ones of Argyris et al.[155]. In addition, we include the inelastic analysis, which demonstrates that plastification starts immediately after the instability. Material and geometric data are shown in Table 3.7. Again, we highlight the impact of a coupled multiaxial law by investigating a variation of this problem. For all cases, inelastic response with kinematic hardening is assumed. Convergence data in residual norm and energy norms for the case of thick members are also shown in Tables 3.8, 3.9 for selected steps. In Fig. 3.14 we also show plots of convergence rates for the global Newton method when using the consistent and continuous tangent respectively for two representative steps. It is evident that use of the continuous tangent modulus slows down convergence significantly.

As can be seen from Fig. 3.13b for the inelastic analysis of the thick member variation, the equilibrium paths corresponding to a coupled and uncoupled constitutive law tend to diverge shortly after the start of plastic deformation.



(a) Structure.



(b) Load vs deflection at point B.

**Figure 3.12.** Geometry and loading of two-beam structure. Comparison of load vs deflection curves for different cases.

**Table 3.8** Convergence in residual norm during steps 5, 15, 25 35.  $\epsilon_{tol} = 10^{-10}$ .

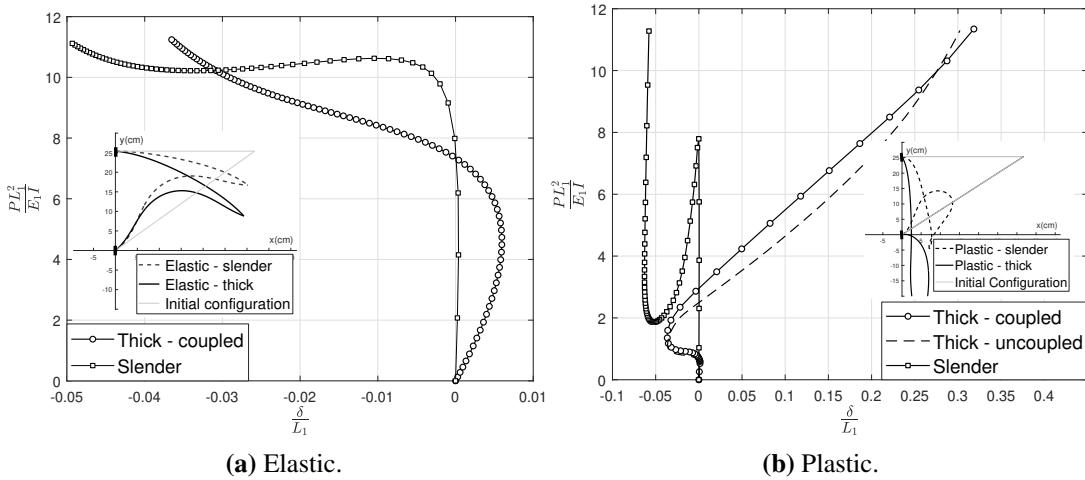
iters.	5	15	25	35
1	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$
2	$1.251499 \cdot 10^{-2}$	$1.988783 \cdot 10^{-2}$	$1.737845 \cdot 10^{-3}$	$6.187773 \cdot 10^{-2}$
3	$2.132711 \cdot 10^{-5}$	$1.066255 \cdot 10^{-5}$	$1.824164 \cdot 10^{-6}$	$4.214041 \cdot 10^{-3}$
4	$9.483080 \cdot 10^{-10}$	$4.882520 \cdot 10^{-10}$	$2.229120 \cdot 10^{-11}$	$6.111545 \cdot 10^{-5}$
5	$5.150000 \cdot 10^{-14}$	$7.790000 \cdot 10^{-13}$	$4.800000 \cdot 10^{-14}$	$1.218231 \cdot 10^{-8}$
6				$7.110000 \cdot 10^{-14}$

The full post-buckling elastoplastic equilibrium curve for the slender model is also provided for comparison. In Fig. 3.13a the post-buckling response of both slender and thick models are illustrated. In the former, both uniaxial and multiaxial constitutive models yield identical results, assuming an infinite yield stress.

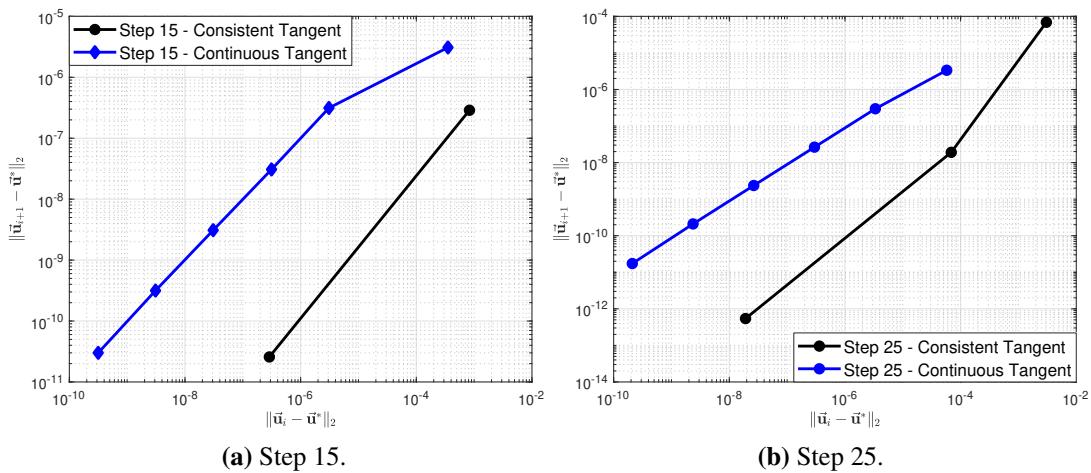
As we reduce the slenderness, slight differences arise compared to a purely uniaxial behavior at the fiber. This is mainly due to the assumed strain distribution, which is user defined and not accounted in the purely flexural model.

**Table 3.9** Convergence in energy norm during steps 5, 15, 25, 35.  $\epsilon_{tol} = 10^{-10}$ .

iters.	5	15	25	35
1	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$	$1.000000 \cdot 10^0$
2	$3.050574 \cdot 10^{-4}$	$1.065944 \cdot 10^{-4}$	$2.618542 \cdot 10^{-2}$	$1.188951 \cdot 10^{-2}$
3	$2.235712 \cdot 10^{-5}$	$5.373434 \cdot 10^{-5}$	$4.974768 \cdot 10^{-5}$	$1.495507 \cdot 10^{-4}$
4	$1.532842 \cdot 10^{-8}$	$2.585263 \cdot 10^{-9}$	$1.551550 \cdot 10^{-7}$	$1.706384 \cdot 10^{-5}$
5	$6.283000 \cdot 10^{-12}$	$1.962000 \cdot 10^{-12}$	$1.687900 \cdot 10^{-11}$	$3.948037 \cdot 10^{-7}$
7				$5.157400 \cdot 10^{-11}$



**Figure 3.13.** Influence of slenderness and constitutive model on the response for elastic and elastoplastic behavior.



**Figure 3.14.** Convergence rates for two representative steps when using the consistent and continuous tangents at fibers.

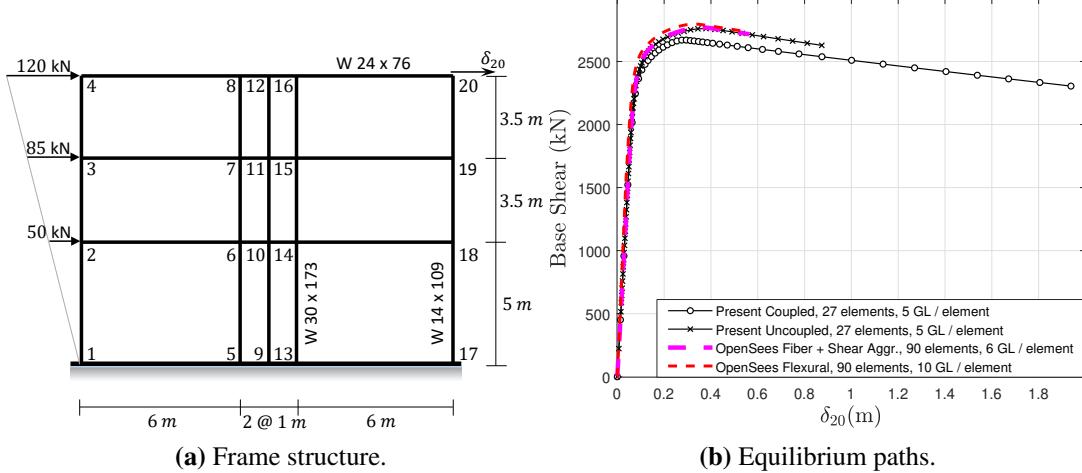
### 3.5.6 Three Story Frame With Shear Links

In this example we are concerned with the three-story four-bay steel frame shown in Fig. ???. It was first analyzed in [157] and in the present investigation the diagonal braces are ignored. Gravity loads are derived from the material and geometric properties of the elements and, for simplicity, are assumed to be acting on frame joints only. The values for each joint gravity load are shown in Table 3.10.

**Table 3.10** Vertical gravity loads at nodes.

Nodes	2,18	3,19	4,20	6,14	7,15	8,16	10	11	12
$\downarrow P_i(\text{kN})$	240	220	145	340	300	190	255	215	120

A static pushover analysis is then conducted with respect to the superimposed horizontal loads on nodes 2,3, and 4. All members have the same yield stress and Young's modulus:  $\sigma_y = 290 \text{ MPa}$ ,  $E = 210 \text{ GPa}$ , and the Poisson's ratio is set to  $\nu = 0.3$ . Both material and geometric nonlinearity options are activated for the analysis and perfect plasticity is assumed. In Fig. 3.15b we compare the results we get using the hybrid NLP element with the nonlinear flexibility element in OpenSees[132]. In OpenSees we used two different modelling approaches with regards to cross-section constitutive behavior: 1) a purely flexural fiber section and 2) a section aggregator object, whereby a uniaxial shear force-strain law is overloaded on an existing fiber section. In the latter approach, the flexural and shear effects remain uncoupled. For the analyses using the proposed element, we again adopted 2 different approaches. In the first approach we assumed an uncoupled section constitutive law, whereby axial-flexural effects are determined by integrating the relevant equations over the section fibers while the shear effects are assumed concentrated in the centerline. The underlying assumption in this approach is that the shear strains are uniform. For the second approach we assumed full coupling at the section fibers utilizing the J2 constitutive law and computationally solving the stress update problem using the return mapping algorithm proposed in this chapter. As can be seen from 3.15b, the results we get with aggregator sections in OpenSees are, expectably, identical with



**Figure 3.15.** Geometry of the steel frame and 3.15b response curves for three different models.

the uncoupled constitutive model when using the NLP element. Both predict a higher collapse load than the fully coupled NLP, which is also an expected result. The purely flexural model in OpenSees gives an even slightly higher collapse load than the uncoupled models. It is clear that, in the presence of shearing components, such as the effective shear wall in this example, a purely flexural model will result in non-conservative estimates even in simple push-over analyses.

To achieve convergent results, the analyses with OpenSees required 90 elements with six Gauss-Lobatto points per element for the section aggregator model and ten points for the purely flexural model. Each member besides the shear links was discretized into 4 nonlinear beam-column elements. In contrast, for the analyses using the element proposed in Chapter 2 only one hybrid element per member was required, resulting in 27 elements in total, and five Gauss-Lobatto quadrature points per element. Both models account for second order effects under the presence of gravity loads.

### 3.6 Summary

In this chapter a return mapping algorithm for planar, shear flexible beam finite elements was presented that fully accounts for shear-axial-flexure interaction. The proposed formulation is particularly suited for flexibility-based or hybrid-type elements, which tend to rely more on

higher order quadrature rules. Focusing our attention on the von Mises yield criterion with both linear kinematic and isotropic hardening, we have shown that the proposed model is much faster in terms of computational cost than conventional stress update procedures typically implemented in beam models. In addition, we derived the material tangent modulus consistent with the fully implicit Euler scheme used for the integration of rate constitutive equations, as well as the general form of the consistent section tangent stiffness. It was seen that the latter depends on the return mapping scheme and the assumption on shear strain distribution on the cross-section. Use of consistent tangents is crucial in the numerical treatment of such problems because they restore the quadratic rates of converge for the global Newton method. Lastly, the accuracy and efficacy of the proposed framework was demonstrated in a number of numerical problems, presented in the last section.

## CHAPTER 4

### NATURALLY PARAMETERIZED NLP FRAMEWORK

#### 4.1 Introduction

This chapter presents a parametric nonlinear programming (PNLP) framework that extends the problem solving capabilities of the Hybrid NLP beam element beyond analyses concerned with mechanical loading. Such cases can be contact problems or structural sensitivity analysis. A strong feature of this framework lies in the utilization of homotopy continuation principles in order to establish conditions of global convergence to at least one solution. With the term “solution” we broadly refer to any critical point for the objective function, which in the present context is the TPE, with respect to the system variables.

Homotopy continuation, which forms the backbone of the PNLP framework, has been the subject of extensive studies in the field of applied mathematics[64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75] as a technique for “globalizing” root finding algorithms. That is, even if the initial guess is not in the neighborhood of a root, homotopy theory guarantees convergence to at least one root, given certain regularity conditions hold. These conditions in essence establish the existence of a smooth path connecting the initial guess and that root. When applied to finding a root of a system of equations  $F(x) = 0$ , this can be done by introducing an “easy” problem whose a solution is easy to find, say  $x - x_0 =$ , and then construct a homotopy function  $H(x, t) = tF(x) + (1 - t)(x - x_0)$ , which represents a system of  $n$  equations with  $n + 1$  unknowns. The system  $H(x, t) = 0$  is sequentially solved for a discretization of  $t \in [0, 1]$ , starting at  $t_0 = 0$ , where the solution is known (“initial guess”). When  $t = 1$  we have arrived at a root of the initial problem  $F(x) = 0$ . The process of varying  $t$  from 0 to 1 and solving the homotopy function is called deformation (not to be confused with mechanical deformation) and construction of such a homotopy function is called artificial homotopy. It is clear that the same principles can be

applied if system  $F(x) = 0$  is undetermined, with  $n$  equations,  $n + 1$  unknowns and a solution is available or can be found easily. In a nonlinear optimization context, application of the same principles have led to the formulation of global optimization algorithms[158, 159, 160, 161, 162, 163, 164, 165]. In addition, this framework highlights the firm theoretical underpinnings of various incremental methods that have been developed independently over the past decades for the numerical treatment of engineering problems[87, 81, 82, 83, 84, 85, 86, 95, 111, 101, 104, 117, 118, 119, 120, 121]. Application of this framework for approximating roots of systems of equations is commonly referred to as numerical continuation, whereas in the the context of finding optimizers it is termed parametric optimization or PNLP.

In structural mechanics, differential equilibrium equations can be reformulated as an optimization problem due to the existence of a variational structure, as discussed in Chapter 2. Thus, the PNLP framework provides a suitable and versatile engine for generating global solutions to engineering problems using the Hybrid NLP-based element, since it allows for the incorporation of constraint specifications directly at the minimization statement. It is shown that, in the presence of inequality constraints, the global path that connects the initial guess to a critical point is continuous but piecewise differentiable[158, 160, 159]. Finally, utilizing the underlying homotopy arguments, we establish the conditions under which the path we follow is unique and convergent to at least one critical point. Non-uniqueness of the homotopy path implies the existence of bifurcation points. Because the derivations do not make use of artificial functions in order to introduce initial guesses, we termed the formulation as Naturally Parameterized NLP (NPNLP). While this has the downside of requiring a known solution of the system in order to initiate the homotopy deformation process, in structural mechanics such states can often be found easily (e.g. initial configuration).

This chapter is divided into 5 sections. In the first section we give an overview of homotopy continuation as applied in (globally) solving systems of nonlinear equations and in parametric optimization. In the second section formulate the NPNLP framework for the Hybrid element and in section three we outline the conditions that both the TPE and the element constraints

need to satisfy in order to guarantee global convergence following a unique path. A brief note on bifurcating paths is also included but the topic is not explored further herein. Finally, we demonstrate the capabilities of the framework through a set of problems from the mechanics and applied mathematics literature.

## 4.2 Homotopy Continuation

### 4.2.1 Main idea

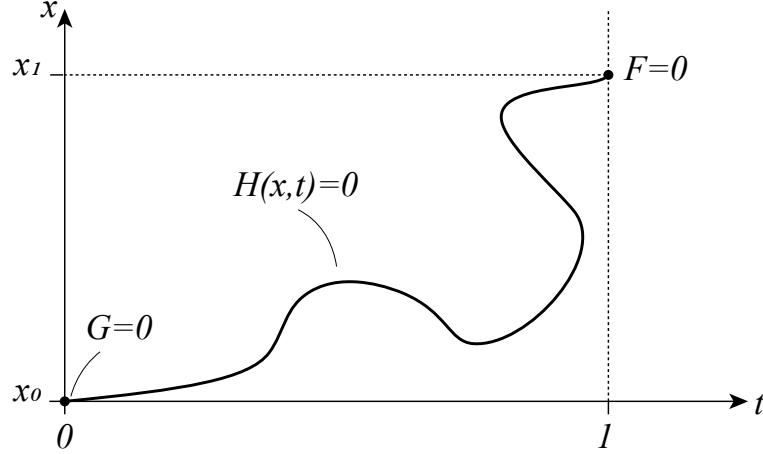
The application of numerical approximation methods in virtually all engineering and scientific problems leads, generally, to an algebraic system of equations,  $\vec{F}$ , to be solved for the system unknowns,  $\vec{x}$ . This system, of which we seek one or more roots, is expressed in the following form:

$$\vec{F}(\vec{x}) = \vec{0} \quad (4.1)$$

where for simplicity we assume that  $\vec{x} \in \mathbb{R}^n$  and  $\vec{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a sufficiently smooth vector function. In the derivations that follow, boldface symbols denote vectors while bracketed bold-face symbols denote matrices. Given an initial estimate,  $\vec{x}_0$ , one can use the Newton method to iteratively find a root:

$$\vec{x}_{i+1} = \vec{x}_i - \mathbf{F}_{\mathbf{x}, i}^{-1} \vec{F}_i, \quad i = 0, 1, \dots \quad (4.2)$$

This approach will yield a solution if i)  $\vec{x}_0$  is already in the vicinity of a root and ii) if the Jacobian  $\mathbf{F}_{\mathbf{x}}$  is not singular for any of the iterates  $\{\vec{x}\}_i$ . The main idea of the homotopy continuation is to overcome the difficulty that singularity points pose by imbedding problem (4.1) in a higher dimensional space where both  $\vec{x}$  and parameter  $t$  are considered independent variables. This is done by pairing function  $\vec{F}$  with a simpler problem,  $\vec{G} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  which has a trivial solution  $\vec{G}(\vec{x}_0) = \vec{0}$ , using a homotopy function,  $\vec{H}(\vec{x}, t)$ . For  $\vec{H} : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$  we require that i)  $\vec{H} = \vec{0}$  for all  $(\vec{x}, t)$  and ii)  $\vec{H}(\vec{x}_0, 0) = \vec{G}(\vec{x}_0)$  and  $\vec{H}(\vec{x}_1, 1) = \vec{F}(\vec{x}_1) = \vec{0}$ . Under certain regularity conditions, the Jacobian of  $\vec{H}$ ,  $D\vec{H}$ , is of full rank even at points where  $\mathbf{F}_{\mathbf{x}}$  is singular and, by the implicit function theorem, there exists a unique differentiable path  $\Gamma : \mathbb{R} \rightarrow \mathbb{R}^n \times \mathbb{R}$ ,



**Figure 4.1.** A homotopy path connecting the trivial solution of  $G = 0$  to a root of  $F = 0$  for a single scalar equation dependent on  $x$ .

with  $\Gamma(s) = (\vec{x}(s), t(s))$ ,  $s \in \mathbb{R}$  and  $\vec{H}(\Gamma) = \vec{0}$ . This path connects the solution of the trivial problem,  $\vec{x}_0$ , at  $t_0 = 0$  with at least one root of  $\vec{F}$ ,  $\vec{x}_1$ , when  $t = 1$  and can be tracked using numerical continuation[64]. The process of varying parameter  $t$  from  $0 \rightarrow 1$  is referred to as deformation from the simple problem  $\vec{G} = \vec{0}$  to the initial one,  $\vec{F} = \vec{0}$ , and such a path is depicted in figure (2.1).

The path  $\Gamma$  depends on how  $\vec{H}$  is defined, which in turn is determined by the choice for  $\vec{G}$ . We distinguish two general cases of homotopy imbeddings: artificial and natural.

One typically resorts to artificial homotopies when a simple solution to problem (4.1) is generally not available. The homotopy parameter  $t$  and the simpler problem  $\vec{G}$  are then simply devices that initialize the deformation process. In this case, only the final point  $(\vec{x}_1, 1)$  is of interest and not the path  $\Gamma$ . Such an artificial deformation process is depicted in Fig. 4.1. A commonly used form of artificial imbedding is the convex homotopy:

$$\vec{H}(\vec{x}, t) = t\vec{F}(\vec{x}) + (1 - t)\vec{G}(\vec{x}) = \vec{0} \quad (4.3)$$

Among the variety of convex homotopy functions that have been developed, the ones most frequently used in implementations are:

- Probability one homotopy:  $\vec{G}(\vec{x}) = \vec{x} - \vec{x}_0$  [68, 166]

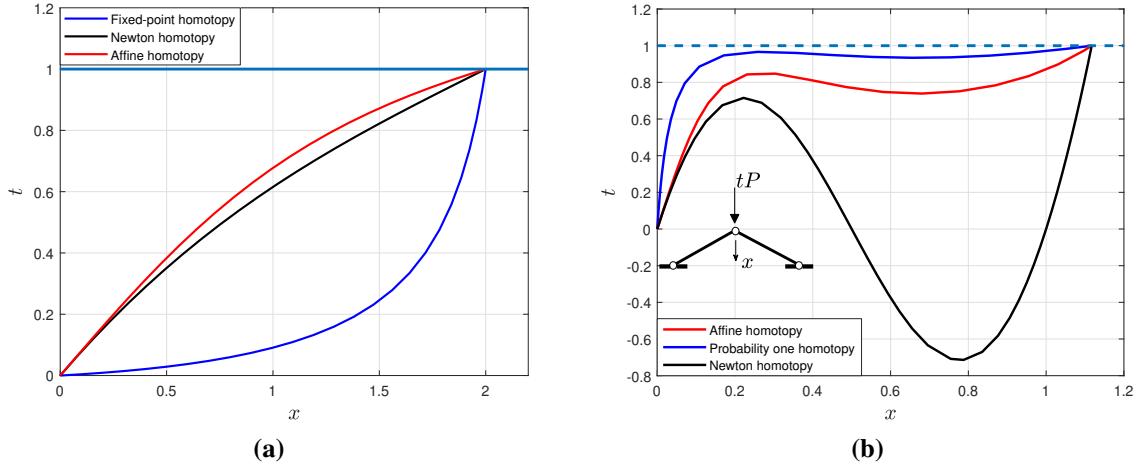
- Newton homotopy:  $\vec{G}(\vec{x}) = \vec{F}(\vec{x}) - \vec{F}(\vec{x}_0)$  [66, 167]

- Affine homotopy:  $\vec{G}(\vec{x}) = \mathbf{F}_{\mathbf{x}0}(\vec{x} - \vec{x}_0)$  [168, 75]

On the other hand, systems whose response depends on variations of a parameter,  $t$ , that has physical meaning can be recast as natural homotopies as follows:

$$\vec{R}(\vec{x}) = 0 \xrightarrow{\text{parameterization}} \vec{R}(\vec{x}, t) = 0 \quad (4.4)$$

Frequently, in such cases an initial state of the system,  $\vec{R}_0$ , is easily found and the deformation process starts from there. The process ends when one or more target states are found, which, by convention, correspond to  $t = 1$ . In contrast with artificial imbeddings, the path  $\Gamma$  generated by Eq. (4.4) is also of interest because all  $(\vec{x}, t) \in \Gamma$  represent states of the system. As an example, the residual balance equations in nonlinear mechanics,  $\vec{R}(\vec{u}) = \vec{0}$ , can be cast as a natural homotopy,  $\vec{R}(\vec{u}, t) = \vec{0}$ , where  $\vec{u}$  is the displacement vector and  $\lambda$  represents the load intensity of extern loads.



**Figure 4.2.** Homotopy paths for (a)  $f(x) = x^3 - 6x^2 + 21x - 26 = 0$ , with starting point  $x_0 = 0$  and convergence at root  $x_1 = 2$  when  $t = 1$  and (b) truss, with initial state the undeformed configuration( $x_0 = 0$ ) and target load level  $P = 0.03$ .

A simple demonstration is provided in figure 4.2, where the three artificial homotopies mentioned above are used in two problems. In the first problem (Fig. 4.2a), we find a root of a cubic polynomial, while in the second (Fig. 4.2b) we solve the one degree of freedom truss for target state corresponding to load  $P = 0.03$ . Details for this example are taken from [96]. We see from the truss case that while, in principle, artificial imbeddings can be used for systems that can be naturally parameterized, the intermediate solution points on the path do not correspond to actual equilibrium configurations.

#### 4.2.2 Regularity and Existence of Homotopy Paths

From the discussion so far it is evident that the study of mappings  $\vec{H} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  is of primary importance. Consider such a mapping,  $\vec{H}(\vec{v})$ , which is at least of  $C^k$  continuity,  $k \geq 2$ , with  $\vec{v} \in \mathbb{R}^{n+1}$  and  $t$  being its  $(n+1)$ th component. For any given  $\vec{w} \in \mathbb{R}^n$ , the equation  $\vec{H}(\vec{v}) = \vec{w}$  represents a system of  $n$  equations with  $n+1$  unknowns and we designate the subset of  $\mathbb{R}^{n+1}$  satisfying this system as follows:

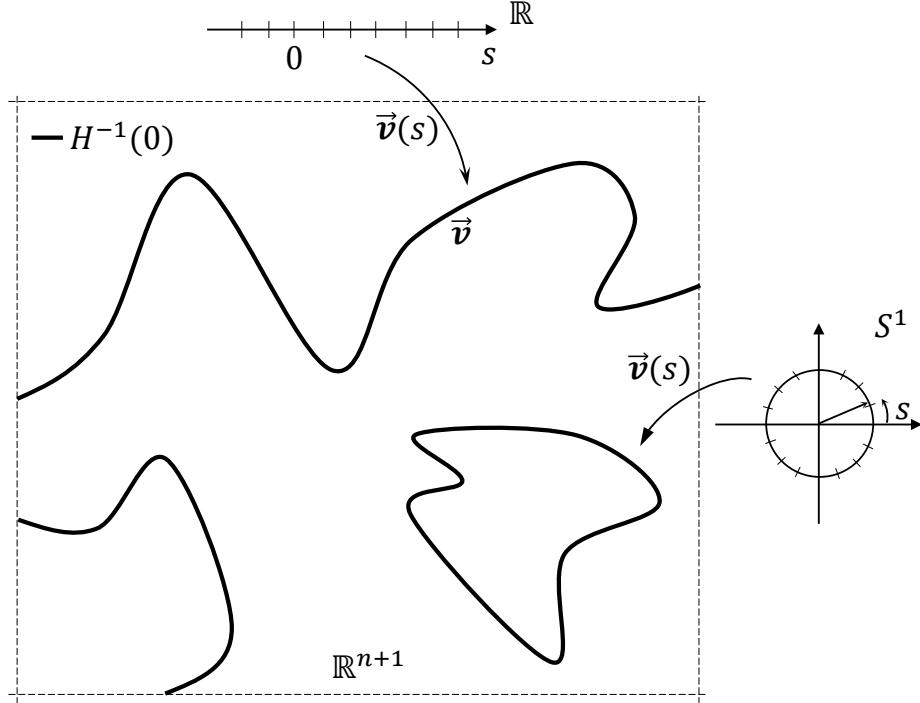
$$\vec{H}^{-1}(\vec{w}) = \{\vec{v} \in \mathbb{R}^{n+1} \mid \vec{H}(\vec{v}) = \vec{w}\} \subseteq \mathbb{R}^{n+1}$$

**Definition 4.2.1** (Regular values of  $\vec{H}$ ). *An element  $\vec{w} \in \mathbb{R}^n$  is a regular value of  $\vec{H} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  if its Jacobian,  $\mathbf{D}\vec{H}$ , has rank  $n$  for all  $\vec{v} \in \vec{H}^{-1}(\vec{w})$ .*

**Definition 4.2.2** (Regular points of  $\vec{H}$ ). *An element  $\vec{v} \in \mathbb{R}^{n+1}$  is a regular point of  $\vec{H} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  if its Jacobian at that point is of rank  $n$ .*

The components of the  $n \times (n+1)$  Jacobian are given by  $\mathbf{D}\vec{H}_{ij} = \partial H_i / \partial v_j$ . It follows that if  $\vec{w}$  is a regular value for  $\vec{H}$ , then all  $\vec{v} \in \vec{H}^{-1}(\vec{w})$  are regular points. The following lemma[64], stated here without proof, plays a constitutive role in homotopy continuation theory:

**Lemma 4.2.1.** *Let  $\vec{H} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  be a  $C^k$  map,  $k \geq 2$ , and having  $\vec{0}$  as a regular value. Then, the set  $\vec{H}^{-1}(\vec{0})$  is a  $C^k$  1-dimensional manifold in  $\mathbb{R}^{n+1}$ , comprised of non-intersecting*



**Figure 4.3.** Components of  $\vec{H}^{-1}(\vec{0})$  in  $\mathbb{R}^{n+1}$ .

components that are either i) homeomorphic with the unit circle  $S^1$  (loops) or ii) homeomorphic with the real line  $\mathbb{R}$  (paths).

We refer to the collection of paths and loops in  $\vec{H}^{-1}(\vec{0})$  as (connected) components. Figure 4.3 depicts a collection of such components for illustrative purposes. If  $\vec{0}$  is a regular value of  $\vec{H}$ , by the implicit function theorem, a path(loop)  $\Gamma$  can be described as a  $C^k$  map  $\Gamma : \mathbb{R}(\text{or } S^1) \rightarrow \mathbb{R}^{n+1}$  such that  $\vec{v} = \vec{v}(s)$ . The tangent to that curve is given by  $\dot{\vec{v}} = d\vec{v}/ds$  and is always a non-zero vector(cf. [64, 168]).

Consider now the case of  $\vec{H}(\vec{v})$  where the  $n + 1$ -th component of  $\vec{v}$  can be regarded as a special parameter of interest, designated as  $t$ . We are interested in the zero points of  $\vec{H}$ :

$$\vec{H}(\vec{x}, t) = 0 \quad (4.5)$$

where  $\vec{x} \in \mathbb{R}^n$ ,  $t \in \mathbb{R}$ . Any given initial point  $(\vec{x}_0, 0)$  that satisfies Eq. (4.5) sets us on a

component of  $\mathbf{H}^{-1}(\vec{\mathbf{0}})$  which we will designate as  $\Gamma_0$  and define as follows:

$$\Gamma_0 = \{(\vec{x}, t) \mid \vec{x} \in \mathbb{R}^n, t \in \mathbb{R}, (\vec{x}_0, 0) \text{ in } \Gamma_0, \vec{\mathbf{H}}(\vec{x}_0, 0) = \vec{\mathbf{0}}\} \quad (4.6)$$

In most applications we want to ensure the initial point  $\vec{x}_0$  sets us on a component  $\Gamma_0$  which is a path, not a loop. Under the smoothness and regularity assumptions, the condition which guarantees that is for the  $n \times n$  submatrix  $\mathbf{D}\mathbf{H} = \partial\vec{\mathbf{H}}/\partial\vec{x}$  to be non-singular at  $\vec{x}_0$ [168]. More conditions are generally required to guarantee that the path will reach  $t = 1$  at least once and depend on the particular form of  $\vec{\mathbf{H}}$ . The interested reader can refer to [64], which provide proofs of existence and convergence for various types of homotopies. In addition, it is customary to formulate homotopies with  $t \in [0, 1]$ . In our formulation we adhere to the convention that a solution is reached when  $t = 1$ . However, as is the case in many practical applications, the restriction of  $t$  to lie in the unit interval is relaxed and the parameter is allowed to take values outside of that range. The expectation is that more solutions will be reached by crossing the level  $t = 1$  multiple times. However, it should be noted that homotopy theory guarantees the (global) convergence to one solution  $(\vec{x}_1, 1)$ . While certain homotopy imbeddings can find multiple solutions when tracking numerically a single component path[66, 102, 169], it is difficult to establish conditions that guarantee convergence to many or all solutions in general[169].

#### 4.2.3 Parametric Nonlinear Programming

Our derivations follow closely the works of Kojima & Hirabayashi[158], Gfrerer et al.[161], Gfrerer, Wacker et al.[159], Guddat et al.[160]. Consider the following one-parameter optimization program:

$$\mathbf{P}(t) : \min\{f(\vec{x}, t) \mid \vec{x} \in \Omega(t), t \in \mathbb{R}\} \quad (4.7)$$

with

$$\Omega(t) = \{\vec{x} \in \mathbb{R}^n \mid h_i(\vec{x}, t) = 0, i \in I_h, g_j(\vec{x}, t) \leq 0, j \in I_g\} \quad (4.8)$$

where we assume that functions  $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ ,  $h_i : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  and  $g_j : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  are considered to be  $C^k$ ,  $k \geq 3$ ,  $I_h = \{1, \dots, l\}$ ,  $I_g = \{1, \dots, p\}$  and  $l \ll n$ ,  $p \ll n$ . Let  $\mathcal{L}(\vec{x}, \vec{\lambda}, \vec{\mu}, t)$  be the corresponding Lagrangian function of problem  $\mathbf{P}(t)$ :

$$\mathcal{L}(\vec{x}, \vec{\lambda}, \vec{\mu}, t) = f(\vec{x}, t) + \sum_{i \in I_h} \lambda_i h_i(\vec{x}, t) + \sum_{j \in I_g} \mu_j g_j(\vec{x}, t) \quad (4.9)$$

where  $\vec{\lambda} \in \mathbb{R}^l$  and  $\vec{\mu} \in \mathbb{R}^p$  are vectors of Lagrange multipliers, with  $\vec{\lambda} = [\lambda_1 \dots \lambda_l]^T$ ,  $\vec{\mu} = [\mu_1 \dots \mu_p]^T$ . Additional specifications need to be imposed on set  $\Omega(t)$  in order to derive a set of equations which will determine the solution set. Two conditions which we base our exposition on are the Linear Independence Constraint Qualification (LICQ) and the Strict Complementarity Condition (SCC). Let the index set of the active inequality constraints be defined as follows:

$$I_g^a(p) = \{j \mid g_j(\vec{x}, t) = 0, j \in I_g\} \quad (4.10)$$

**Definition 4.2.3 (LICQ).** At any  $\vec{x} \in \Omega(t)$ , if the gradients of the active constraints  $\{\nabla h_i, \nabla g_j\}$ ,  $i \in I_h$ ,  $j \in I_g^a$ , is a set of linearly independent vectors, then the linear independence constraint qualification holds at  $\vec{x}$ .

**Definition 4.2.4 (SCC).** At any  $\vec{x} \in \Omega(t)$ , if the Lagrange multipliers of the active set of inequality constraints are strictly positive,  $\mu_j > 0 \forall j \in I_g^a$ , then the strict complementarity condition holds at  $\vec{x}$ .

Under the LICQ, SSC and the smoothness assumptions, if  $\vec{x}$  is a minimum(maximum) point for  $\mathbf{P}(t)$ , then there exist unique  $\vec{\lambda}$ ,  $\vec{\mu}$  such that the point  $(\vec{x}, \vec{\lambda}, \vec{\mu}, t)$  satisfies the so-called Karush-Kuhn-Tucker (KKT) system:

$$\mathbf{KKT}(t) : \quad \nabla_{\vec{x}} \mathcal{L}(\vec{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, t) = \vec{0} \quad (4.11a)$$

$$h_i(\vec{x}, t) = 0, \quad i \in I_h \quad (4.11b)$$

$$\mu_j g_j(\vec{x}, t) = 0 \quad (4.11c)$$

$$\mu_j \geq 0, g_j(\vec{x}, t) \leq 0, \quad j \in I_g \quad (4.11d)$$

The primary objective of PNLP is to examine the local and global behavior of points that satisfy the  $\mathbf{KKT}(t)$  system (4.11). It is convenient to recast the KKT system above into a form that contains only equalities and, thus, homotopy continuation principles can be applied.

Let  $T$  be a subdivision of the augmented Lagrangian space  $\mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^p \times \mathbb{R}$  into cells  $\tau(\bar{I}_g)$ , where  $\tau(\bar{I}_g) = \mathbb{R}^n \times \mathbb{R}^l \times \{\mu \in \mathbb{R}^p \mid \mu_j \geq 0, j \in \bar{I}_g, \mu_j \leq 0, j \in I_g \setminus \bar{I}_g\} \times \mathbb{R}$  for  $\bar{I}_g \subset I_g$  and  $I_g \setminus \bar{I}_g = \{j \in I_g \mid j \notin \bar{I}_g\}$ . That is, the space is partitioned in cells with each cell identified by a particular active index set. Using an active index set strategy, Gfrerer et al[159] showed that problem  $\mathbf{P}(t)$  and, consequently,  $\mathbf{KKT}(t)$  can be replaced by an equivalent reduced system, which contains only (active) equality constraints[160, 159]:

$$\mathbf{P}^\tau(t) : \quad \min\{f(\vec{x}, t) \mid h_i(\vec{x}, t) = 0, i \in I_h, g_j(\vec{x}, t) = 0, j \in \bar{I}_g, \vec{x} \in \mathbb{R}^n, t \in \mathbb{R}\} \quad (4.12)$$

with the corresponding Lagrangian being:

$$\mathcal{L}^\tau(\vec{x}, \vec{\lambda}, \vec{\mu}, t) = f(\vec{x}, t) + \sum_{i \in I_h} \lambda_i h_i(\vec{x}, t) + \sum_{j \in \bar{I}_g} \mu_j g_j(\vec{x}, t) \quad (4.13)$$

Setting  $\vec{z} = (\vec{x}, \vec{\lambda}, \vec{\mu}) \in \mathbb{R}^{n+l+|\bar{I}_g|}$ , where  $|\bar{I}_g| \leq p$  is the cardinality of  $\bar{I}_g$ , the reduced KKT

system for program **P** is:

$$\mathbf{KKT}^\tau(t) : \vec{\mathbf{H}}_{KKT}^\tau(\vec{\mathbf{z}}, t) = \begin{bmatrix} \nabla_{\vec{\mathbf{x}}} \mathcal{L}^\tau(\vec{\mathbf{z}}, t) \\ -h_i(\vec{\mathbf{x}}, t), i \in I_h \\ -g_j(\vec{\mathbf{x}}, t), j \in \bar{I}_g \end{bmatrix} = \begin{bmatrix} \vec{0} \\ 0 \\ 0 \end{bmatrix} \quad (4.14)$$

where  $\vec{\mathbf{H}}_{KKT}^\tau(\vec{\mathbf{z}}, t) : \mathbb{R}^{n+l+|\bar{I}_g|+1} \rightarrow \mathbb{R}^{n+l+|\bar{I}_g|}$ , that is, in each cell  $\vec{\mathbf{H}}_{KKT}^\tau$  is of the form 4.5. This reformulation presupposes that i) LICQ and SCC hold throughout and ii) that  $\vec{0}$  is a regular value of the reduced system  $\vec{\mathbf{H}}_{KKT}^\tau$  on each cell  $\tau(\bar{I}_g)$ , which in turn leads to the following corollaries[158]:

1. The set  $\cup_\tau [\vec{\mathbf{H}}_{KKT}^\tau]^{-1}(\vec{0})$  is comprised of piecewise continuously differentiable paths(loops) on  $\mathbb{R}^{n+l+p+1}$ .
2. Each path(or loop) is  $C^{k-1}$  in each  $\tau(\bar{I}_g) \neq \emptyset$ .
3. If a path(or loop) crosses a cell boundary between cells  $\tau(\bar{I}_g^1)$  and  $\tau(\bar{I}_g^2)$ , then sets  $\bar{I}_g^1$  and  $\bar{I}_g^2$  differ by only one index and the tangent vector to the path at the intersection is not tangent to the boundary.

In addition, utilizing condition 3 above, there exists[159] a unique subdivision  $\mathcal{D} = \bigcup_{q=1}^Q W^q$  of an interval  $[\alpha, \beta]$  on the real line, such that:

- $[\alpha, \beta] = \bigcup_{q=1}^Q W^q$ ,  $W^q = [s_q, s_{q+1}]$ , and  $s_1 = \alpha, s_Q = \beta$ ,  $q = 1, \dots, Q$ .
- $\bar{I}_g(s) = I^q$  where  $I^q = I_g^a(s)$ , for all  $s$  in  $(s_q, s_{q+1})$ .
- $I^q \neq I^{q+1}$ .
- By the implicit function theorem, there exist a  $C^{k-1}$  function  $\vec{\phi}(s) = (\vec{\mathbf{z}}(s), t(s)) : W^q \rightarrow \mathbb{R}^{n+l+|\bar{I}_g|+1}$ .
- $\vec{\phi}(s)$  is the solution set for  $\mathbf{KKT}^\tau(t)$  for all  $s \in W^q$ .

In addition, if we require that  $\|\mathrm{d}\vec{z}/\mathrm{d}s, \mathrm{d}t/\mathrm{d}s\| = 1$ , then  $s$  represents the arc-length of the path(or loop). By repeating the process for  $q = 1, \dots, Q$ , a  $C^{k-1}$  piecewise continuous component  $\Gamma_0$  can be traced numerically using appropriate continuation techniques. At a particular solution point  $(\vec{z}^*, t^*)$ , the values of inactive constraints  $g_j(\vec{x}^*, t^*)$ ,  $j \in I_g \setminus I^q$  and the Lagrange multipliers of the active constraints,  $\mu_j$ ,  $j \in I^q$ , are checked to ensure that complementarity slackness holds (eq. (4.11d)). Since adjacent cells differ in their respective active index sets by only one index, violation of the complementarity slackness condition indicates that the continuation entered cell  $\tau(I^{q+1})$  and there is a unique index,  $j_k$ , such that:

$$g_{j_k}(\vec{x}^*, t^*) \geq 0, \quad j_k \in I_g \setminus I^q \quad \rightarrow \quad I^{q+1} = I^q \cup \{j_k\}, \quad g_{j_k} = 0$$

OR

$$\mu_{j_k} \leq 0, \quad j_k \in I^q \quad \rightarrow \quad I^{q+1} = I^q \setminus \{j_k\}, \quad \mu_{j_k} = 0$$

The collection of paths and loops of stationary points satisfying (4.11) is depicted in Fig. 4.4. For further details on the relevant proofs, refer to Gfrerer et al.[159] and the textbook by Guddat, Vasquez & Jongen[160].

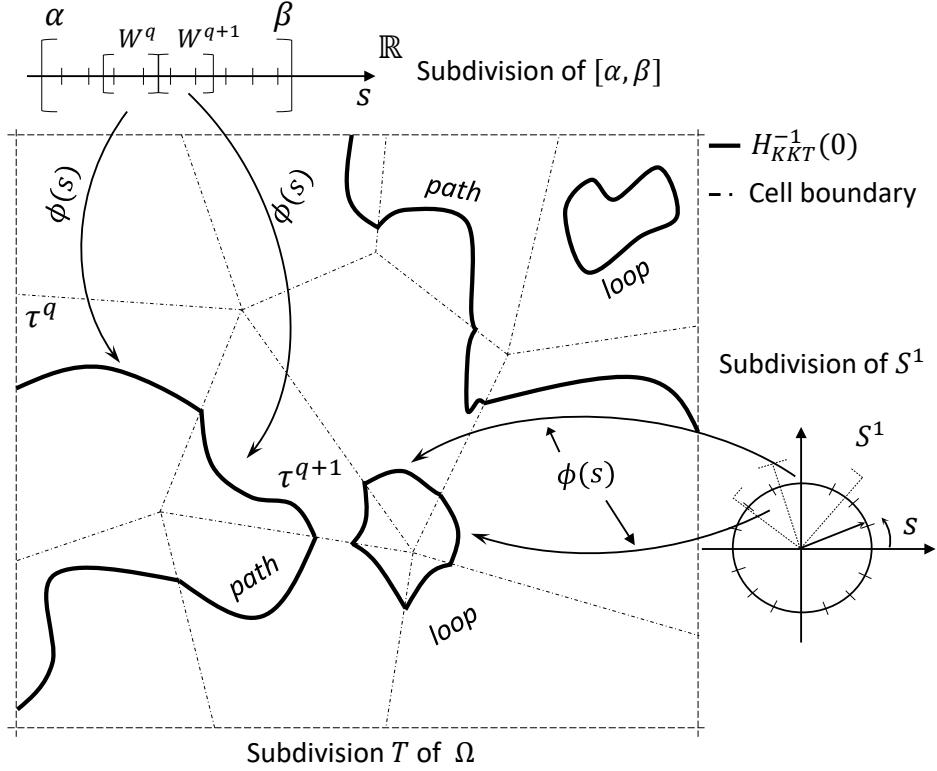
#### 4.2.4 Global Optimization

Consider the following NLP:

$$\mathbf{P} : \quad \min\{f(\vec{x}) \mid \vec{x} \in \Omega\} \quad (4.15a)$$

$$\Omega = \{\vec{x} \in \mathbb{R}^n \mid h_i(\vec{x}) = 0, \quad i \in I_h, \quad g_j(\vec{x}) \leq 0, \quad j \in I_g\} \quad (4.15b)$$

where the same assumptions hold for  $f, h_i$  and  $g_j$  as in PNLP problem (4.7). Conventional numerical tools based in mathematical analysis guarantee the convergence to a (local) minimizer  $\vec{x}^*$  of  $\mathbf{P}$  provided that the initial guess is within the domain of convergence for the



**Figure 4.4.** Piecewise continuously differentiable paths and loops of  $H_{KKT}^{-1}(0)$  in  $\mathbb{R}^{n+l+p+1}$ .

particular method employed. By utilizing Homotopy principles, we can convert  $\mathbf{P}$  into a PNLP using an appropriate imbedding. A numerical algorithm with global convergence properties can then be constructed, given that the conditions discussed previously hold. Consider the following one-parameter PNLP:

$$\mathcal{P}(t) : \quad \min\{\mathcal{F}(\vec{x}, t) \mid \vec{x} \in \Omega(t), t \in [0, 1]\} \quad (4.16a)$$

$$\Omega(t) = \{\vec{x} \in \mathbb{R}^n \mid \mathcal{H}_i(\vec{x}, t) = 0, i \in I_h, \mathcal{G}_j(\vec{x}, t) \leq 0, j \in I_g\} \quad (4.16b)$$

For  $\mathcal{P}(t)$  to be a homotopy imbedding of  $\mathbf{P}$ , we require that:

1. NLP  $\vec{\mathcal{P}}(0)$  possesses a trivial solution.
2.  $\vec{\mathcal{P}}(1) = \mathbf{P}$

If  $\mathbf{P}$  can be naturally parameterized, then  $\mathcal{P}(t) = \mathbf{P}(t)$ ,  $\mathcal{F}(\vec{x}, t) = f(\vec{x}, t)$  and a stationary point, not necessarily unique, of  $\mathbf{P}(0)$  is usually easy to compute. Artificial imbeddings can also be

employed when  $\mathbf{P}$  cannot be naturally parameterized. In such cases, a trivial stationary point for  $\mathcal{P}(0)$  is easy to compute by construction. Below we list two artificial imbeddings that are frequently used in practice:

- $\mathcal{F}(\vec{x}, t) = tf(\vec{x}) + \frac{1}{2}(1-t)\|\vec{x} - \vec{x}_0\|^2$
- $\mathcal{F}(\vec{x}, t) = tf(\vec{x}) + \frac{1}{2}(1-t)\|f(\vec{x}) - f(\vec{x}_0)\|^2$

### 4.3 Naturally Parameterized NLP Framework for Mechanics

We will now formulate a naturally parameterized nonlinear programming (NPNLP) framework for solid and structural mechanics applications and establish the conditions that guarantee the global convergence to a solution, using homotopy and PNLP principles discussed previously. The parameterization is considered natural because we assume that i) underlying physical quantities are parameterizable and ii) a solution for the initial state, not necessarily unique, is relatively easy to obtain. This solution can then be used as the starting point of the NPNLP deformation process.

Consider the hybrid NLP element program involving the discretized TPE (2.21) and the discretized (equality) constraints (2.28):

$$\mathbf{P} : \quad \min\{\Pi(\vec{d}) \mid \vec{d} \in \Omega\} \quad (4.17a)$$

$$\Omega = \{\vec{d} \in \mathbb{R}^n \mid h_i(\vec{d}) = 0, i = 1, \dots\} \quad (4.17b)$$

In the derivations that follow we assume a set of inequality constraints is also introduced. Furthermore, to simplify the exposition, we assume a cell subdivision of the parameter domain and work only with the set of inequality constraints that are active,  $\vec{g}_a$ . Since this allows us to deal only with equality constraints, we can consider vector  $\vec{h}$  to include both the standard hybrid element constraints (2.32a),(2.32b) and  $\vec{g}_a$ . Consequently, the corresponding vector of Lagrange

multipliers,  $\vec{\lambda}$ , will be augmented with the non-zero components of vector  $\vec{\mu}$ , as defined in the previous section. Under these assumptions, the general form of the naturally parameterized NLP (NPNLP) program can be stated as follows:

$$\mathcal{P}(t) : \min\{\Pi(\vec{d}, t) \mid \vec{d} \in \Omega(t), t \in \mathbb{R}\} \quad (4.18a)$$

$$\Omega(t) = \{\vec{d} \in \mathbb{R}^n \mid h_i(\vec{d}, t) = 0, i = 1, \dots, l\} \quad (4.18b)$$

where in  $\vec{d}$  we have collected all state variables associated with the structural system (e.g. strains, displacements etc.). The Lagrangian function of  $\mathcal{P}(t)$ , counterpart to (2.30), is:

$$\mathcal{L}(\vec{d}, \vec{\lambda}, t) = \Pi(\vec{d}, t) + \vec{h}^T \vec{\lambda} \quad (4.19)$$

with  $\vec{h}(\vec{d}, t) = [\vec{h}^A(\vec{d})^T \ \vec{h}^B(\vec{d})^T \ g_a(\vec{d}, t)^T]^T$ . The corresponding parameterized KKT system for a particular  $t$ , considering again  $\vec{z} = [\vec{d}^T \ \vec{\lambda}^T]^T, \vec{z} \in \mathbb{R}^{n+l}$ :

$$\vec{H}(\vec{z}, t) = \begin{bmatrix} \nabla_{\vec{d}} \mathcal{L}(\vec{z}, t) \\ \vec{h}(\vec{d}, t) \end{bmatrix} = \vec{0} \quad (4.20)$$

Since  $\vec{H} : \mathbb{R}^{n+l+1} \rightarrow \mathbb{R}^{n+l}$ , we can apply homotopy continuation principles to system 4.20 in order to numerically track a component (path or loop) in the current cell. We define the set of equilibrium configurations as follows:

$$\mathcal{E} = \vec{H}^{-1}(\vec{0}) = \{(\vec{z}, t) \in \mathbb{R}^{n+l+1} \mid \vec{H}(\vec{z}, t) = \vec{0}\} \quad (4.21)$$

It is implicitly assumed that, for NPNLP  $\mathcal{P}(t)$  to be a natural homotopy, in addition to conditions 1 and 2 mentioned in the preceding section, stationary points that correspond to values of  $t \neq 1$  should represent actual (intermediate) states of our structural system. If an

artificial imbedding was used in place of (4.18a) instead, then only solutions corresponding to  $t = 1$  could be regarded as physical equilibrium states. In that case, during the numerical continuation, the homotopy path need not be tracked with strict tolerance, since intermediate solutions are of no physical significance. For the NPNLP framework however, all solutions along the path are equilibrium configurations and numerical accuracy is required through the whole process. Therefore, we will refer to solutions  $\vec{d}$  of (4.20) as equilibrium configurations or states. We will not be concerned with their characterization (stable, unstable, neutral). Moreover, initial and target values,  $Y_0$ ,  $Y_1$  respectively, of the parameterized quantity  $Y = Y(t)$  are prescribed such that  $Y(0) = Y_0$  and  $Y(1) = Y_1$ . If a trivial equilibrium state is not apparent for  $t = 0$ , then  $\mathcal{P}(0)$  can be solved as an NLP so that an approximate equilibrium configuration  $\vec{d}_0$  is found. With that known, the initial configuration of the system is completely determined and the deformation process of the NPNLP can begin. As an example, consider the case where the parameterized quantity is the load vector  $\vec{P}$  such that the TPE is  $\Pi(\vec{d}, t) = U_s(\vec{d}) - t\vec{P}^T \vec{u}$ , with  $\vec{u}$  being the global nodal displacement vector. If we consider  $Y(t) = t\vec{P}^T \vec{u}$ , then it is clear that for  $Y(0) = 0$ , a trivial equilibrium state of  $\mathcal{P}(0)$  is the unstressed configuration,  $\vec{d} = \vec{0}$ . In contrast, if only the  $k$ -th component of  $\vec{P}$ ,  $P_k$ , is parameterized such that  $Y(t) = tP_k$ , then an equilibrium configuration for  $\mathcal{P}(0)$  will need to be found through an analysis of the structure loaded with  $\vec{P}_0 = [P_1 \ P_2 \ \dots \ P_{k-1} \ 0 \ P_{k+1} \ \dots \ P_n]^T$ .

#### 4.3.1 Existence and Uniqueness of NPNLP Path

We examine now under what conditions a particular component  $\Gamma_0 \subseteq \mathcal{E}$  emanating from  $\vec{d}_0$  converges to an equilibrium state of  $\mathbf{P}$  at least once. The  $(n + l) \times (n + l + 1)$  Jacobian of the KKT system (4.20), considering variations in  $\lambda$  as well, is:

$$\mathbf{D}\mathbf{H} = [\mathbf{H}_z \ \frac{d\vec{H}}{dt}] \quad (4.22)$$

where  $\mathbf{H}_z$  is the Lagrangian matrix defined by:

$$\mathbf{H}_z = \begin{bmatrix} \mathbf{H} & \nabla \mathbf{h} \\ \nabla \mathbf{h}^T & \mathbf{0} \end{bmatrix} \quad (4.23)$$

and  $\mathbf{H} = \nabla_d^2 \mathcal{L}$  is the Hessian matrix of the Lagrangian function (4.19). In the absence of constraints,  $\mathbf{H}_z = \mathbf{H} = \mathbf{K}$ , where  $\mathbf{K}$  is the global stiffness matrix. The following conditions guarantee that a path  $\Gamma_0$  emanating from the couple  $(\mathbf{d}_0, 0)$  (initial “guess”) converges to a solution of  $\mathbf{P} = \mathcal{P}(1)$  at least once if the following conditions hold[159]:

1. The TPE function,  $\Pi$ , and the constraints,  $h_i$ , should be  $C^k$  and  $C^{k-1}$  function respectively,  $k \geq 2$ .
2. Zero is a regular value of the KKT system (4.20) in each cell.
3. The couple  $(\vec{\mathbf{d}}_0, 0)$  is a unique equilibrium state for  $\mathcal{P}(0)$ .
4.  $\mathcal{E}$  is a compact set for  $t \in [0, 1]$  and for the cell subdivision.
5.  $\mathcal{P}(t)$  satisfies the LICQ in all cells corresponding to  $t \in [0, 1]$ .

Condition 3 is the most restrictive as far as natural homotopies are concerned, since it is quite often that  $\mathcal{P}(0)$  possesses multiple stationary points. This may necessitate to perform additional analyses using different solutions of  $\mathcal{P}(0)$  as starting points, if these points lie on different paths on  $\mathcal{E}$ . In addition, convergence to additional solutions of  $\mathcal{P}(1)$ , when  $t$  is allowed to take values greater than one, might subsequently lead to an accumulation point. This suggests that as  $t \rightarrow \infty$ ,  $\vec{\mathbf{d}} \rightarrow \vec{\mathbf{d}}^*$  and the path is therefore not compact.

Condition 5 can be relaxed so that we assume LICQ holds for  $t$  outside  $[0, 1]$  with the expectation of finding additional equilibrium states for  $\mathcal{P}(1)$ . However, this assumption does not guarantee convergence beyond the first stationary point. Moreover, convergence to the first equilibrium point can be guaranteed even with the weaker Mangasarian-Fromovitz constraint

qualification, as is proven in[158, 159]. In that case, the Lagrange multiplier vectors at stationary points are not unique and the set  $\mathcal{E}$  has different structure. Its projection on  $\vec{d} - t$  space however is the same as the one with LICQ assumed.

Regularity condition 2 implies that no secondary branches (bifurcations) occur along a particular path  $\Gamma_0$ , as this would imply there exist points where  $\text{Rank}(\mathbf{DH}) < n+l$ . Let the tangent space of the current active constraints at point  $(\vec{d}, t)$  be defined as follows:

$$\mathcal{T} = \{\vec{\xi} \in \mathbb{R}^n \mid [\nabla_d \vec{h}]^T \vec{\xi} = \vec{0}\} \quad (4.24)$$

Following Poore[163], Tiahrt& Poore[165] we assume that SCC holds for all  $(\vec{d}, t)$  in the interior of a cell that also satisfy Eq. (4.20). Then,  $\vec{0}$  being a regular value for  $\vec{H}$  in any cell  $\tau = \tau(I^q)$ ,  $q = 1, \dots, Q$ , implies the following conditions:

**A. Loss of LIQC** - In the case, we require that:

- $\text{Dim}(\text{Span}(\{\nabla_d h_1 \dots \nabla_d h_l\})) \geq l - 1$ .
- The matrix  $\mathbf{J}_h$ , has full rank:  $\text{Rank}(\mathbf{J}_h) = l$ , where  $\mathbf{J}_h = \begin{bmatrix} \nabla_d \vec{h}^T & \frac{d\vec{h}}{dt} \end{bmatrix}$

**B. Singularity of Hessian  $\mathbf{H}$  on  $\mathcal{T}$**  - In this case, we require that:

- $\text{Rank}(\nabla_d \vec{h}^T \mathbf{H} \nabla_d \vec{h}) \geq l - 1$ .
- $\text{Proj}_{\mathcal{T}}(\nabla_d(\frac{dL}{d\lambda})) \notin \text{Range}(\nabla_d \vec{h}^T \mathbf{H} \nabla_d \vec{h})$ .  
where  $\nabla_d \vec{h}^T$  is the basis of  $\mathcal{T}$  in terms of the (active) constraint gradients and  $\text{Proj}_{\mathcal{T}}(\vec{w}) = \nabla_d \vec{h} (\nabla_d \vec{h}^T \nabla_d \vec{h})^{-1} \nabla_d \vec{h}^T \vec{w}$  is the orthogonal projection of a vector  $\vec{w} \in \mathbb{R}^n$  on the tangent space  $\mathcal{T}$ .

**C. Loss of LIQC and Singularity of  $\mathbf{H}$  on  $\mathcal{T}$**  - In this case, we require that:

- $\text{Dim}(\text{Span}(\{\nabla_d h_1 \dots \nabla_d h_l\})) \geq l - 1$ .
- $\text{Rank}(\mathbf{J}_h) = l$ .

- $\text{Rank}(\nabla_d \vec{\mathbf{h}}^T \mathbf{H} \nabla_d \vec{\mathbf{h}}) \geq l - 1$ .
- $\text{Rank}(\mathbf{H}_z) \geq n + l - 1$ .

The second condition in A and the second condition in B essentially guarantee that, in the event of violation of LICQ or singularity of the Hessian in the tangent space, for the last column of  $\mathbf{D}\mathbf{H}$  we will have  $\frac{d\vec{\mathbf{H}}}{dt} \notin \text{Range}(\mathbf{H}_z)$  and  $\mathcal{E}$  remains a one-dimensional manifold.

Under conditions 1-5 and the restrictions A,B,C implied by the regularity assumption, the component  $\Gamma_0$  is a path which emanates from the initial configuration  $\mathbf{d}_0$  with  $t = 0$  and converges in finite steps in at least one solution of problem  $\mathbf{P}$  when  $t = 1$ . Moreover, the only singular points on a path in the interior of a cell are fold points. When the path crosses a cell boundary, the SCC is violated. This is essentially a bifurcating behavior , where two curves intersect at that boundary. The branch switching is carried out by enforcing the complementarity slackness condition so that we maintain a direction that leads to feasible points. Numerical continuation can now be utilized to track  $\Gamma_0$ , using an appropriate parameterization and an active set strategy.

#### 4.4 Implementation

We now present a practical and robust algorithm for the implicit path continuation for the active set KKT system which is suited for structural mechanics applications using the NPNLP framework. It follows closely the methodology of the secant length algorithm[170] but is modified by employing a technique found in Crisfield[87] and Batoz & Dhatt [84]. This is done so that the algorithm takes into account the frequently well structured form of the Lagrangian matrix  $\mathbf{H}_z$  while at the same time remaining agnostic with respect to the parameterization. This allows to bypass the expensive computation of the pseudo-inverse that is required in the secant length scheme at the start of every step and instead use the homotopy parameter  $t$  to enforce the constraint exactly.

Let  $\Delta(\cdot)_i$  denote the cumulative changes of a quantity up to iteration  $i$  and  $\delta(\cdot)_i$  the iterative

changes in iteration  $i$ , such that:

$$\Delta(\cdot)_i = \Delta(\cdot)_{i-1} + \delta(\cdot)_i \quad (4.25)$$

where  $\Delta(\cdot)_0 = 0$ . The secant length constraint,  $\|(\vec{z} - \vec{z}_N, t - t_N)\|_2^2 = \Delta s^2$ , leads to the following iterative form:

$$\Delta \vec{z}_i^T \Delta \vec{z}_i + \Delta t_i^2 = \Delta s^2, \quad i = 1, \dots, i_{max} \quad (4.26)$$

In the derivations that follow, it is assumed we are at iteration  $i$  and drop the subscript from the iterative change vectors,  $\delta(\cdot)$ . Using Eq. (4.25) for  $\vec{z}$ ,  $t$ , and substituting into Eq. (4.26) yields the following iterative constraint:

$$\Delta \vec{z}_{i-1}^T \Delta \vec{z}_{i-1} + \delta \vec{z}^T \delta \vec{z} + 2\Delta \vec{z}_{i-1} \delta \vec{z} + \Delta t_{i-1}^2 + \delta t^2 + 2\Delta t_{i-1} \delta t = \Delta s^2 \quad (4.27)$$

From Eq. (4.20), we require  $\vec{H}_i = \vec{H}(\vec{z}_{i-1} + \delta \vec{z}, t_{i-1} + \delta t) = \vec{0}$  which, after using Taylor expansion, leads to the following expression:

$$\vec{H}_{i-1} + [\mathbf{H}_z]_{i-1} \delta \vec{z} + \frac{d\vec{H}}{dt} \Big|_i \delta t = \vec{0} \quad (4.28)$$

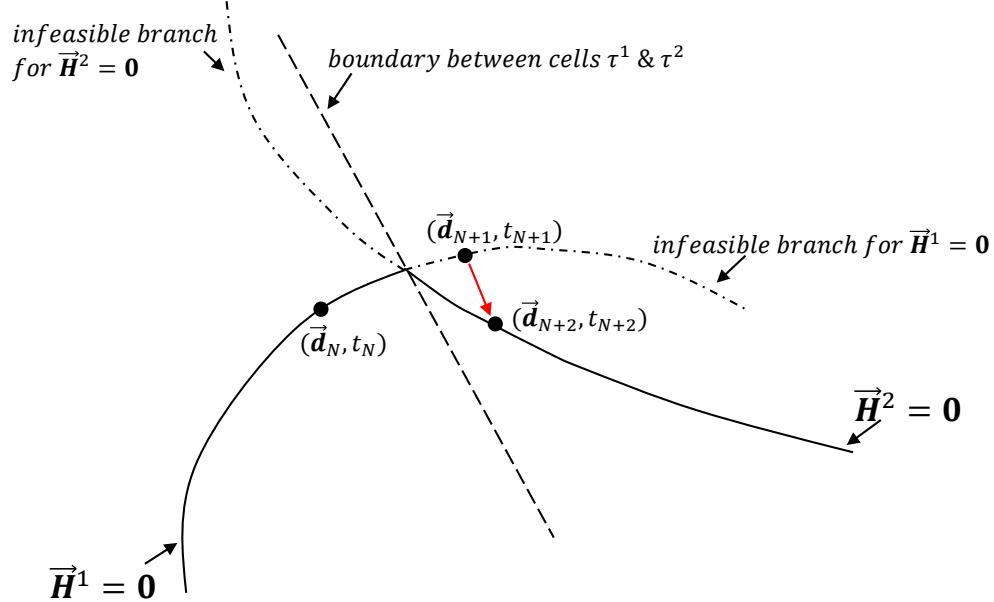
We note here that all quantities in iteration  $i - 1$  are known and that  $\|\vec{H}_{i-1}\|_2 \neq 0$ . Solving Eq. (4.28) for  $\delta \vec{z}$  leads to:

$$\delta \vec{z} = \delta \vec{z}_1 + \delta t \delta \vec{z}_2 \quad (4.29)$$

where

$$\delta \vec{z}_1 = -[\mathbf{H}_z]_{i-1}^{-1} \vec{H}_{i-1}, \quad \text{and} \quad \delta \vec{z}_2 = -[\mathbf{H}_z]_{i-1}^{-1} \frac{d\vec{H}}{dt} \Big|_i \quad (4.30)$$

Substitution of Eq. (4.29) in the iterative constraint (4.27) leads to the following quadratic



**Figure 4.5.** Transition from cell  $\tau^1$  to  $\tau^2$  during numerical continuation. Projection of  $\vec{H}(\vec{z}, t) = \vec{0}$  on  $\vec{d} - t$  space.

equation to be solved for  $\delta t$ :

$$\alpha_1 \delta t^2 + \alpha_2 \delta t + \alpha_3 = 0 \quad (4.31)$$

Coefficients  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  are determined as follows:

- $\alpha_1 = \delta \vec{z}_2^T \delta \vec{z}_2 + 1$
- $\alpha_2 = 2(\delta \vec{z}_1^T \delta \vec{z}_2 + \Delta \vec{z}_{i-1}^T \delta \vec{z}_2 + \Delta t_{i-1})$
- $\alpha_3 = \Delta \vec{z}_{i-1}^T \Delta \vec{z}_{i-1} + \Delta t_{i-1}^2 + \delta \vec{z}_1^T \delta \vec{z}_1 + 2\Delta \vec{z}_{i-1}^T \delta \vec{z}_1 - \Delta s^2$

The root is chosen so that a forward direction of tracking persists. An intuitive criterion proposed by Crisfield[87] suggests selecting the root of Eq. (4.31) that leads to the smallest inner product  $\Delta \vec{z}_i^T \Delta \vec{z}_{i-1}$ , which can be regarded as the iterative form of the continuity condition.

As is the case with the Secant Length parameterization proposed by Menzel & Schwetlick[170], the algorithm presented here bypasses the need to factorize an augmented  $(n+l+1) \times (n+l+1)$  Jacobian by requiring exact satisfaction of the secant length constraint but also avoids the costly numerical computation of the pseudo-inverse. This is done by exploiting the decomposition of

DH in its abstract form (4.22) using the technique of Batoz & Dhatt[84]. However, the form proposed here is more general since the algorithm can handle any parameterization. Moreover, the vector  $\frac{d\vec{H}}{dt}$  is not generally constant and depends on both the specific parameterization as well as the current active set. The algorithm is also suited for general parametric optimization problems where such a decomposition of the Jacobian is amenable.

In the case where the new solution point  $(\vec{z}_{N+1}, t_{N+1})$  violates the complementarity slackness, then a new active set has to be determined, which will lead to a modified  $\vec{H} = \vec{0}$ . Let  $\vec{H}^1$  be the homotopy associated with cell  $\tau^1 = \tau(I^1)$ , and  $\vec{H}^2$  the homotopy with the updated active set on cell  $\tau^2 = \tau(I^2)$ . Then, point  $(\vec{d}_{N+1}, t_{N+1})$ , which corresponds to  $(\vec{z}_{N+1}, t_{N+1})$ , can be used as an initial estimation once the Lagrange multiplier vector has been adjusted according to the current active index set  $I^2$ , and use the condition  $\vec{H}^2 = \vec{0}$  to perform corrections back to the new branch. The infeasible point at step  $N + 1$  can then be discarded from the solution set. Alternatively, other approaches can be implemented, such as interpolation in order to locate the corner point or simply repeat the step from step  $N$  but with reduced step-length so that any solutions off the feasible branches do not lead to significant violation in any constraint. This is illustrated in Fig. 4.5 for steps  $N$ ,  $N + 1$  and  $N + 2$ , where the path is projected on the  $\vec{d} - t$  space.

## 4.5 Numerical Examples

### 4.5.1 Example 1 - Global Optimization

In the first application we are concerned with the global optimization of a 2-D function. In particular, we explore the landscape using PNLP and homotopy principles to generate components that lead to one or more stationary points. Second order optimality principles can be used then to characterize these points. The function we choose is a variation of the Himmelblau function, for which Kuno & Seader[171] provided seven stationary points. The function is:

$$f(x_1, x_2) = 1/2x_1^4 + 1/2x_2^4 + x_1^2x_2 + x_1x_2^2 - 11x_1^2 - 7x_2^2 + 13x_1 + 9x_2 \quad (4.32)$$

with  $-5 \leq x_1, x_2 \leq 5$ . We use two natural imbeddings:

- $f_1(x_1, x_2, t) = 1/2x_1^4 + 1/2x_2^4 + x_1^2x_2 + x_1x_2^2 - 11x_1^2 - 7x_2^2 + 13x_1 + g(t)x_2$
- $f_2(x_1, x_2, t) = 1/2x_1^4 + 1/2x_2^4 + x_1^2x_2 + x_1x_2^2 - 11x_1^2 - 7x_2^2 + 13tx_1 + 9tx_2$

In both cases, the stationary points of Eq. (4.32) and  $f_i(x_1, x_2, 1)$ ,  $i = 1, 2$ , coincide. Function  $g(t)$  in  $f_1$  is defined in such a way so that  $g(1) = 9$  and  $g(0) = 50$ . This results in a linear function  $g(\lambda) = -41\lambda + 50$ . We chose  $g(0) = 50$  so that a (starting) stationary point of  $f_1(x_1, x_2, 0)$  is not close enough to a stationary solution of Eq. (4.32). Moreover, we let  $t$  get values larger than 1 with the hope that more than one stationary points is found per component followed.

The first order optimality condition requires  $\vec{H}_i(x_1, x_2, t) = \nabla_{\mathbf{x}} f_i = 0$ ,  $i = 1, 2$ , for any variation in  $\lambda$ :

$$\vec{H}_1(x_1, x_2, t) = \begin{bmatrix} 2x_1^3 + 2x_1x_2 + x_2^2 - 22x_1 + 13 \\ 2x_2^3 + x_1^2 + 2x_1x_2 - 41t + 50 \end{bmatrix}, \quad \vec{H}_2(x_1, x_2, t) = \begin{bmatrix} 2x_1^3 + 2x_1x_2 + x_2^2 - 22x_1 + 13 \\ 2x_2^3 + x_1^2 + 2x_1x_2 + 9t \end{bmatrix} \quad (4.33)$$

The  $2 \times 3$  Jacobians associated with  $\vec{H}_i$  are given, in block form, as  $\mathbf{D}\mathbf{H}_i = [[\mathbf{H}_{\mathbf{x}}]_i \frac{d\vec{H}}{dt}]$ , with:

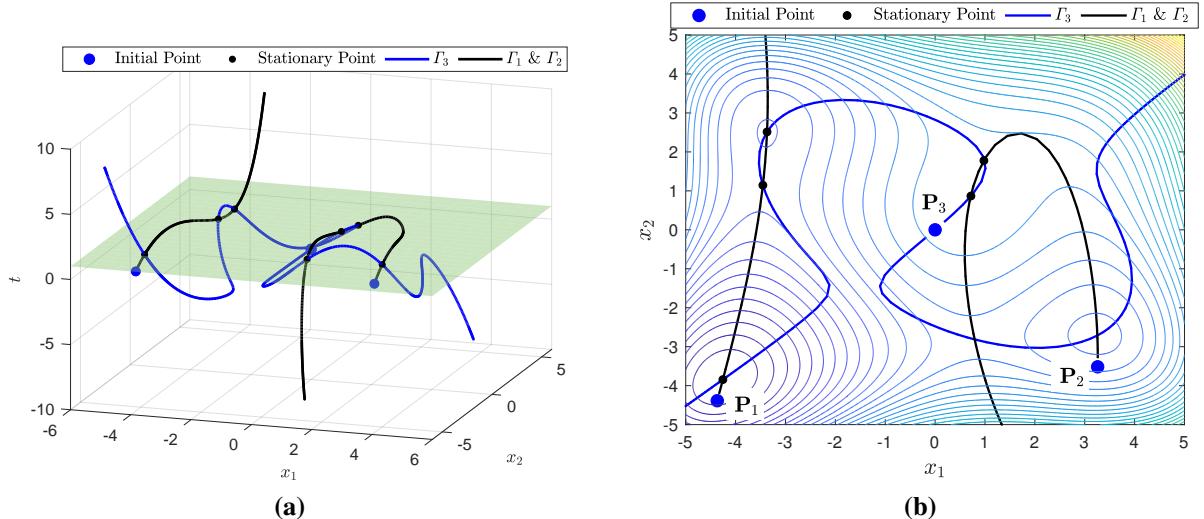
$$[\mathbf{H}_{\mathbf{x}}]_1 = [\mathbf{H}_{\mathbf{x}}]_2 = \begin{bmatrix} 6x_1^2 + 2x_2 - 22 & 2x_1 + 2x_2 \\ 2x_1 + 2x_2 & 6x_2^2 + 2x_1 \end{bmatrix}, \frac{d\vec{H}}{dt} \Big|_1 = \begin{bmatrix} 0 \\ -41 \end{bmatrix}, \frac{d\vec{H}}{dt} \Big|_2 = \begin{bmatrix} 0 \\ 9 \end{bmatrix} \quad (4.34)$$

Two paths are generated,  $\Gamma_1$  and  $\Gamma_2$ , using imbedding  $f_1$ , emanating from points  $\mathbf{P}_1 = (-4.3676, -4.3868)$  and  $\mathbf{P}_2 = (3.2589, -3.5165)$ . These points are stationary points of the initial program  $f_1(x_1, x_2, 0)$  and they are found using an artificial Newton homotopy. The initial points that were then used as starting points in the algorithm described in Sec. 4.4 were  $(\mathbf{P}_1, 0)$ ,  $(\mathbf{P}_2, 0)$  for  $\Gamma_1$  and  $\Gamma_2$  respectively. The step-length used was  $\Delta s = 0.4$  and  $t_{max}$  was set to 5. For the second imbedding, an obvious stationary solution of  $f_2(x_1, x_2, 0)$  is  $\mathbf{P}_3 = (0, 0)$ . We used the point  $(\mathbf{P}_3, 0)$  to generate path  $\Gamma_3$  with  $\Delta s = 0.4$  and  $t_{max} = 12$ , which we track in both directions. As can

**Table 4.1** Stationary solutions of Eq. (4.32).

Kuno & Seader[171]	$\Gamma_1$	$\Gamma_2$	$\Gamma_3$
$x_1$	$x_2$		
-4.255	-3.845	✓	✓
-3.454	-1.144	✓	✓
-3.369	2.514	✓	✓
0.987	1.801		✓
0.713	0.856		✓
0.793	-2.814		✓
3.252	-2.712		✓

be seen from Table (4.1), path  $\Gamma_1$  finds three stationary points,  $\Gamma_2$  finds four and  $\Gamma_3$  finds all seven stationary points. In figure (4.6a) these components are depicted in the  $\mathbb{R}^3$  space and in figure (4.6b) we show their projections on  $\mathbb{R}^2$ , interpreting them as the trajectory followed on the landscape of  $f(x_1, x_2)$ .



**Figure 4.6.** (a) Paths  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma_3$  in  $\mathbb{R}^3$ , (b) interpretations of paths as trajectories on the target TPE function (4.32).

### 4.5.2 Example 2 - Global Sensitivity

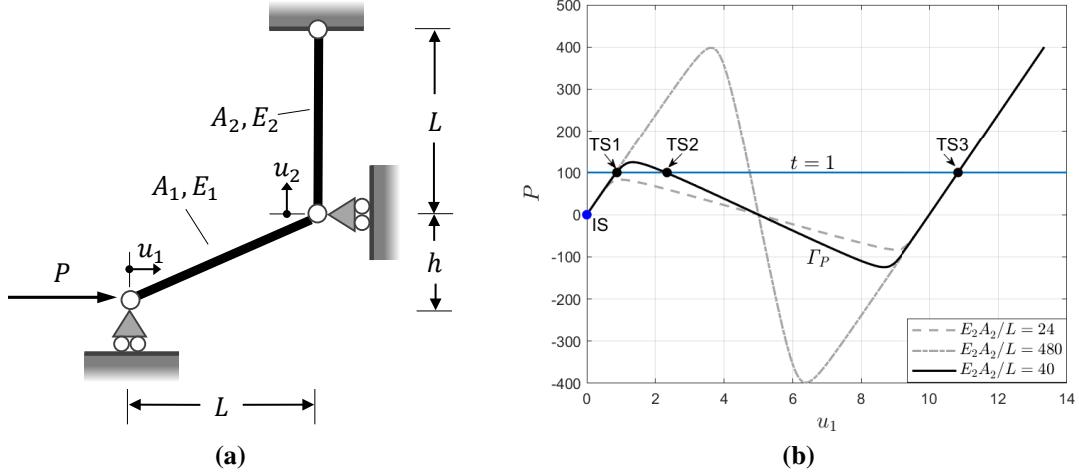
In this application we investigate the global sensitivity of a two degree of freedom truss structure with respect to the axial stiffness of one member. The example, which has appeared in different forms in various investigations[172, 173], is adopted from [111] and modified in accordance with Fig. ???. Assuming linear elastic constitutive law, the system TPE is as follows:

$$\Pi(u_1, u_2) = \frac{1}{2} \frac{E_1 A_1}{L} \left( \sqrt{(L - u_1)^2 + (h + u_2)^2} - \sqrt{L^2 + h^2} \right)^2 + \frac{1}{2} \frac{E_2 A_2}{L} u_2^2 - P u_1$$

with the Young's moduli being  $E_1 = E_2 = 30$ , cross-sectional area for member 1 is  $A_1 = 20$ ,  $L = 5$ ,  $h = 2$  and external load  $P = 100$ . To facilitate the exposition, we first parameterize with respect to external load in a linear fashion,  $P(t) = 100t$ , and solve for three different values of  $A_2$ :  $A_2 = \{4A_1, A_1/3, A_1/5\}$ . We will then conduct a parametric study with  $A_2$  being the “natural” parameter, while the external load will remain fixed at  $P = 100$ . The equilibrium paths from the initial analyses, which correspond to the three distinct values of  $A_2$ , will be used as a reference for the global sensitivity of the structure to (large) variations of  $A_2$ , starting from the system with  $A_2 = 4A_1$  and setting as Target System(TS) the one with  $A_2 = A_1/3$ . The case where  $A_2 = A_1/5$  was included to highlight the fact that below a certain threshold in terms of  $A_2$ , the system no longer possesses multiple solutions for  $P = 100$ . The load parameterized NPNLP is:

$$\mathcal{P}_P(t) : \min \{\Pi(u_1, u_2, t) \mid u_1, u_2 \in \mathbb{R}, t \in \mathbb{R}\} \quad (4.35)$$

The first order optimality condition requires that  $\vec{\mathbf{H}}(u_1, u_2, t) = \nabla_{\mathbf{u}} \Pi = \vec{0}$  for all  $t$ . Again, the resulting Jacobian  $\mathbf{DH}$  is of dimension  $2 \times 3$ , where  $\mathbf{DH} = [\mathbf{H}_u \ \mathbf{H}_t]$ , where  $\mathbf{H}_t$  indicates differentiation of  $\vec{\mathbf{H}}$  with respect to  $t$ . In the absence of constraints, the Lagrangian matrix  $\mathbf{H}_u$  corresponds to the stiffness matrix of the structure while  $\mathbf{H}_t = \nabla_{tu} \Pi = [-P \ 0]^T$ . For each distinct  $A_2$ ,  $\mathcal{P}_P(0)$  possesses a trivial solution, namely, the unstressed configuration:



**Figure 4.7.** (a) Truss structure, (b) equilibrium paths for  $A_2 = \{4A_1, A_1/3, A_1/5\}$  projected on the  $u_1 - P$  plane.

$(u_1, u_2)_0 = (0, 0)$ . Starting from this state and applying the continuation algorithm described in Sec. 4.4 with  $t_{max} = 4$ , we generate the three equilibrium paths depicted in Fig. 4.7b. We now proceed to investigate the global sensitivity of the system with respect to  $A_2$ . We define a nominal value for  $A_2$ ,  $\bar{A}_2 = A_1$ , and induce variations by introducing the parameterization  $A_2(t) = (-11/3t + 4)\bar{A}_2$ . In this way, the Target States (TS) have been chosen to correspond to  $A_2(t = 1) = A_1/3$  and are also depicted in Fig. 4.7b. The resulting NPNLP is:

$$\mathcal{P}_{A_2}(t) : \min\{\Pi(u_1, u_2, t) \mid u_1, u_2 \in \mathbb{R}, t \in \mathbb{R}\} \quad (4.36)$$

Again, the natural homotopy equation  $\vec{H} = \vec{0}$  results from the first order conditions, with  $\mathbf{H}_u$  acting as the structure stiffness, however, in this case we have for  $\mathbf{H}_t$ :

$$\mathbf{H}_t = -\frac{11}{3} \frac{E_2 \bar{A}_2}{L} \begin{bmatrix} 0 \\ u_2 \end{bmatrix} \quad (4.37)$$

The initial NLP  $\mathcal{P}_{A_2}(0)$ , which corresponds to  $A_2(0) = 4\bar{A}_2 = 4A_1$ , has three stationary points for  $P = 100$ . We use two of them as Initial States (IS) to start the continuation process for  $\mathcal{P}_{A_2}(t)$ :

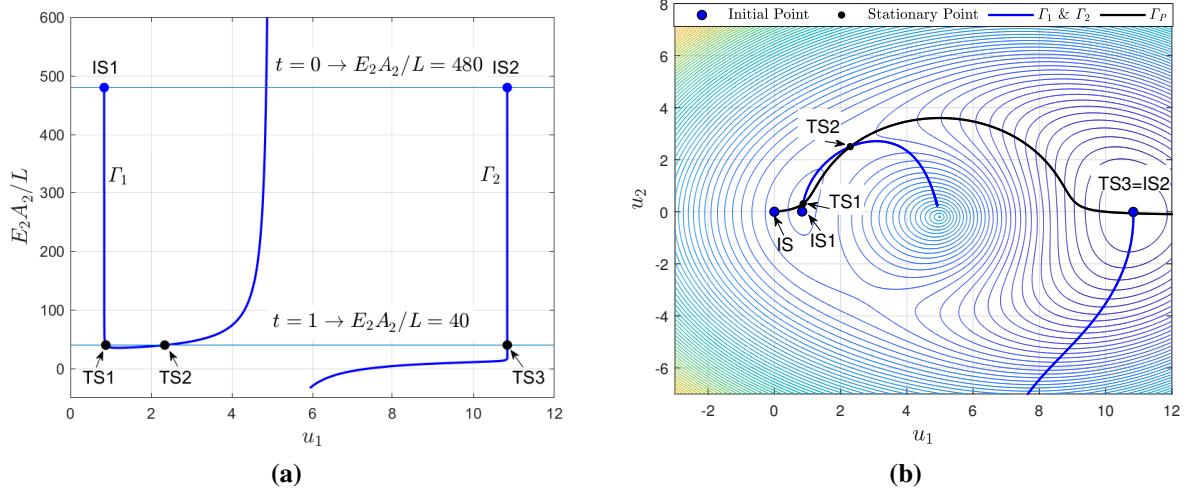
**Table 4.2** Initial States for paths  $\Gamma_1$ ,  $\Gamma_2$  and Target States reached.

	$u_1$	$u_2$	$\Gamma_1$	$\Gamma_2$
IS1	0.836	0.010	×	
IS2	10.836	-0.006		×
TS1	0.867	0.310	✓	
TS2	2.302	2.507	✓	
TS3	10.836	-0.060		✓

$$1. \text{ IS1} : (u_1, u_2) = (0.836, 0.010)$$

$$2. \text{ IS2} : (u_1, u_2) = (10.836, -0.006)$$

The paths generated from IS1 and IS2, shown in Fig. 4.8a, are  $\Gamma_1$  and  $\Gamma_2$  respectively. As can be seen from this figure and Table 4.2, the target NLP  $\mathcal{P}_{A_2}(1)$  has three stationary points with  $\Gamma_1$  passing through two of them (TS1 and TS2), while  $\Gamma_2$  reaches only one, TS3. This can also be illustrated by projecting  $\Gamma_1$  and  $\Gamma_2$  on the target TPE function, as can be seen in Fig. 4.8b, where we have also included the equilibrium path,  $\Gamma_P$ , associated with the load parameterized target system discussed earlier,  $\mathcal{P}_P(t)$  with  $A_2 = A_1/3$ . Clearly, the points of intersection between this equilibrium path and paths  $\Gamma_1$  and  $\Gamma_2$  are precisely the points TS1, TS2 and TS3. Moreover, the persistence of vertical orientation of paths beyond a certain point hints to the existence of an accumulation point. As can be seen in Fig. 4.8a, for  $\Gamma_1$ , we have  $|t| \rightarrow \infty$  and  $(u_1, u_2) \rightarrow (4.950, 0.078)$  and, if traced in the opposite direction,  $(u_1, u_2) \rightarrow (0.835, 0.001)$  which are indications of accumulation points. That is, the initial point IS1 is already an accumulation point. Physically, this means that after a certain value of axial stiffness for the vertical member,  $u_1$  is only due to deformation of the inclined truss member. Similarly,  $\Gamma_2$  also has an accumulation point if traced backwards. These accumulation points can also be illustrated in Fig. 4.8a, where the equilibrium paths converge before TS1 and after  $u_1 \approx 9.2$ . In addition, we see from Figs. 4.7b, 4.8a that when axial stiffness  $E_2 A_2 / L$  becomes even lower, then there exists only one solution for  $P = 100$ .



**Figure 4.8.** (a) projection of components  $\Gamma_1$  &  $\Gamma_2$  on the  $u_1 - E_2 A_2(t)/L$  plane, (b) paths  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma_P$  projected on the TPE of the target NLP  $\mathcal{P}_{A_2}(1)$ .

#### 4.5.3 Example 3 - Frictionless Contact

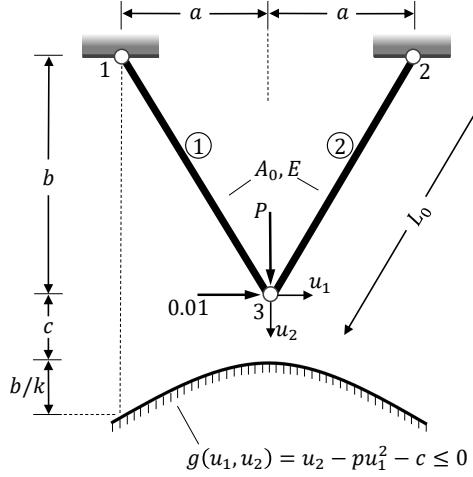
In this example we consider a two-dimensional structure comprised of two bars, shown in Fig. 4.9, and investigate the response in the presence of a frictionless rigid surface described by function  $g$  in terms of the displacements  $u_1, u_2$  of node 3. This configuration has previously been investigated under stability considerations[174, 175]. Here, we assume the following nonlinear elastic constitutive law:

$$\sigma_{PK2} = \frac{1}{2}E\ln(1 + \epsilon_G) \quad (4.38)$$

where  $E$  is the Young's modulus,  $\sigma_{PK2}$  is the second Piola-Kirchhoff stress and  $\epsilon_G$  is the work conjugate Green's strain. It is equivalent to the constitutive law adopted in the relevant example in [175] when Poisson's ratio is 0.5. An initial imperfection is assumed by prescribing a small horizontal load of magnitude 0.01 and subsequently the response is parameterized with respect to the vertical load  $P$ .

The structure TPE is then:

$$\Pi(u_1, u_2, t) = U_1 + U_2 - 0.01u_1 - tPu_2 \quad (4.39)$$



**Figure 4.9.** Initial geometry for Example 3.

where  $U_i = EA_0/4\ln(1 + \delta_i/L_0)\delta_i$  is the strain energy of bar  $i$  and  $\delta_i$  its elongation, with  $A_0$ ,  $L_0$  being the initial cross-section area and length respectively. The elongations are determined as follows:

- $\delta_1 = \sqrt{(b + u_2)^2 + (a + u_1)^2} - L_0$
  - $\delta_2 = \sqrt{(b + u_2)^2 + (a - u_1)^2} - L_0$

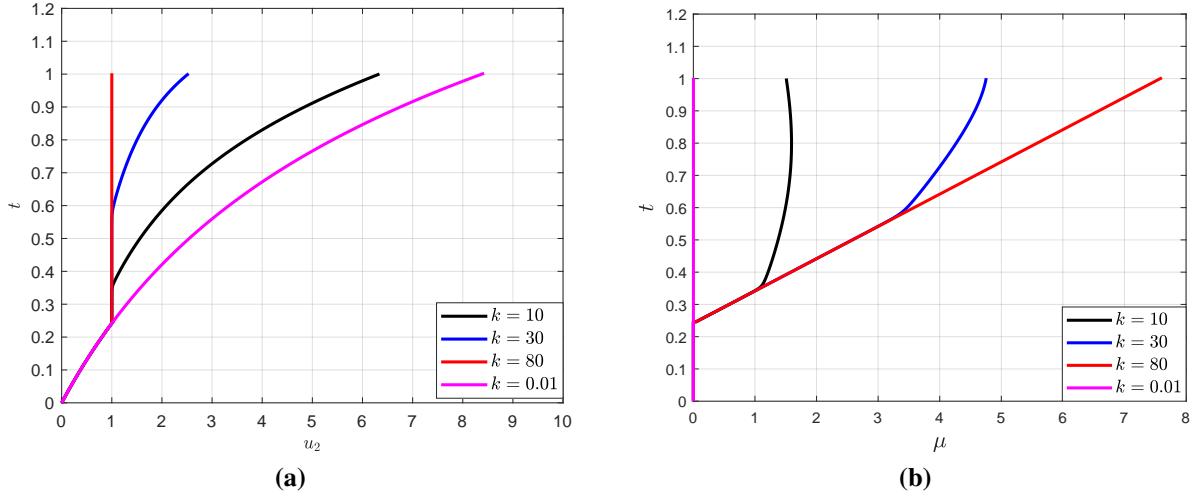
and  $L_0 = \sqrt{a^2 + b^2}$ . The constraint surface is defined as:

$$g(u_1, u_2) = u_2 - pu_1^2 - c \leq 0 \quad (4.40)$$

where  $p = b/(ka^2)$ ,  $k \in (0, \infty)$ , is a parameter controlling the curvature of  $g$ . For  $k \rightarrow \infty$ , the contact surface becomes horizontal at vertical distance  $c$  from node 3, while letting  $k \rightarrow 0$  effectively makes  $g$  inactive for any  $(u_1, u_2)$ . The material and geometric properties were chosen as follows:  $A_0 = 1$ ,  $E = 10$ ,  $a = 1$ ,  $b = 3$ ,  $c = 1$  and the target state is defined when  $P = 10$ . The NPNLP problem is then expressed as follows:

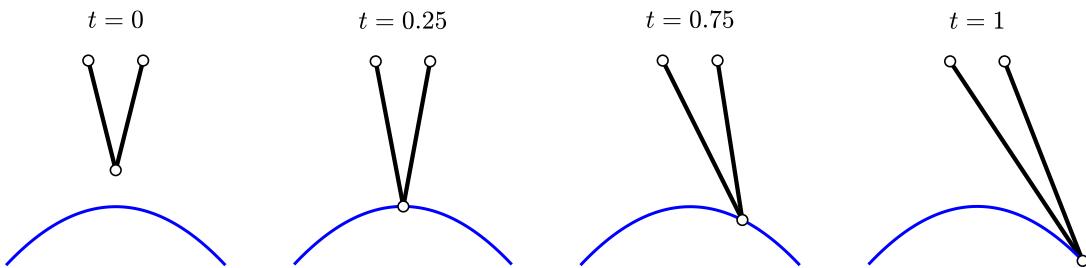
$$\mathcal{P}_P(t) : \min\{\Pi(u_1, u_2, t) \mid u_1, u_2 \in \Omega, t \in \mathbb{R}\} \quad (4.41a)$$

$$\Omega = \{u_1, u_2 \in \mathbb{R} \mid g(u_1, u_2) \leq 0, \} \quad (4.41b)$$



**Figure 4.10.** Projection of path (a) on  $u_2 - t$  plane and (b)  $\mu - t$  plane.

The Lagrangian function for this problem,  $\Omega \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , is defined as  $\mathcal{L}(u_1, u_2, \mu, t) = \Pi(u_1, u_2, t) + \mu g(u_1, u_2)$ , with  $\mu$  being the Lagrange multiplier for  $g$ . The initial point provided to the continuation algorithm is  $\mathbf{P}_0 = (0.0317, 0)$ , which is the unique stationary point for  $\mathcal{P}_P(0)$  and can be sufficiently approximated considering linear response. At that point the constraint  $g$  is inactive and  $\mu$  is set to zero. For values  $u_2 \gtrsim 1$ , the active set changes, which is indicated by the presence of a non-smooth point on the path. Constraint  $g$  becomes active and  $\mu$  is nonzero. The projected paths on the  $u_2 - t$  and  $\mu - t$  planes for four levels of  $k$  are depicted in Fig. 4.10, with the case  $k = 0.01$  corresponding to an effectively flat contact surface. The transition from the first active index set  $I^1 = \{\emptyset\}$  to the second,  $I^2 = \{1\}$  happens  $t \approx 0.25$ . The configuration profiles at four distinct levels of  $t$ , shown in Fig. 4.11, correspond to a surface  $g$  determined with  $k = 30$ .



**Figure 4.11.** Configuration profiles during deformation.

#### 4.5.4 Example 4 - Nonlinear Analysis of Shallow Arch

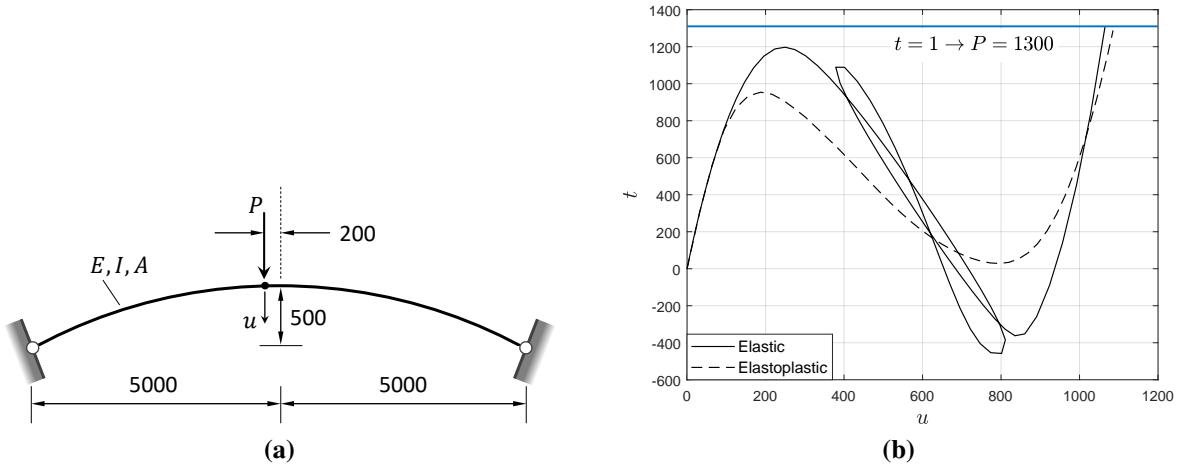
In this application we are concerned with geometrically nonlinear static response of a shallow arch using a constrained NPNLP framework, for both elastic and inelastic behavior. In this application, the parameterized variable is the external load and, for this reason, the proposed algorithm used here coincides with the spherical arc-length method[87].

The structure configuration, seen in Fig. 4.12a, and material properties are taken from Clarke & Hancock[176]: the Young's modulus is  $E = 200$ , the beam cross-sectional area is  $A = 10^4$  with moment of inertia  $I = 10^8$ , while the maximum value for the external load is  $P_{max} = 1300$ . For the numerical discretization, we used  $N_e = 21$  straight NLP hybrid beam elements. The discretized TPE, which is of the form (4.17), is parameterized with respect to the applied load,  $\vec{P}$ , and the resulting NPNLP takes the following form:

$$\begin{aligned} \min \Pi(\vec{u}, \vec{\epsilon}, \vec{\gamma}, \vec{\kappa}, t) &= U(\vec{u}, \vec{\epsilon}, \vec{\gamma}, \vec{\kappa}) - t \vec{P}^T \vec{u} \\ \text{s.t. } \vec{h}(\vec{u}, \vec{\epsilon}, \vec{\gamma}, \vec{\kappa}) &= \vec{0} \\ \vec{u} \in \mathbb{R}^{3(N_e+1)}, \vec{\epsilon}, \vec{\gamma}, \vec{\kappa} &\in \mathbb{R}^{n_i N_e}, t \in \mathbb{R} \end{aligned} \quad (4.42)$$

where  $n_i$  is the number of integration points in each element,  $\vec{\epsilon}$ ,  $\vec{\gamma}$  and  $\vec{\kappa}$  are the vectors of axial strain, shear strain and curvature respectively. Vector  $\vec{h}$  contains the discretized equality constraints (2.32a),(2.32b).

Convergence within a step is reached when the norm of the KKT system (4.20) becomes smaller than a prescribed tolerance,  $\epsilon_{tol}$ . For the inelastic analysis, the multiaxial constitutive fiber model proposed in the previous Chapter was used, however, since the structure is fairly slender, the interaction effects between shear and flexure are negligible.



**Figure 4.12.** (a) Shallow arch structure and loading, (b) equilibrium path using the NPNLP framework.

## 4.6 Summary

In this chapter, we have utilized homotopy continuation principles in the formulation of a naturally parameterized NLP framework for the numerical solution of structural and solid mechanics problems. The discretized TPE is treated as the objective function and any additional constraints can be incorporated in the program directly. The regularity and smoothness conditions are established for the resulting parameterized KKT system, which ensure that the solution space is comprised of smooth or piecewise smooth components. These components (paths or loops) connect an initial state with at least one solution of the system we designate as target and can be traced using numerical continuation techniques. These implicitly defined curves represent how the system responds to large variations of the designated parameter. In addition, the intrinsic relation of these globally convergent methods with the various ad hoc incremental schemes in mechanics is also emphasized and highlighted through the lens of the underlying homotopy theory (see Appendix F). Continuation past the first solution point can often lead to more system solutions, although no firm mathematical basis has been established to guarantee this for arbitrary systems of equations. Moreover, while the formulation reveals how such algorithms can handle fold type turning points along the path, the case of bifurcating branches

is not considered in this work. However, for such cases the proposed framework can tackle secondary branches easily by using perturbation techniques. Lastly, a numerical continuation algorithm is proposed which bypasses the expensive numerical linear algebra associated with the computation of the pseudoinverse or Augmented Jacobian approaches and that is agnostic with regards to the parameterization. The efficacy of the proposed NPNLP framework and continuation scheme is demonstrated through a diverse selection of applications such as global unconstrained optimization, global sensitivity of a simple structure, a frictionless contact problem in nonlinear elastostatics and a load parameterized continuation of a shallow arch for both elastic and inelastic response.

## CHAPTER 5

### WEIGHTED LEAST SQUARES PREDICTOR FOR NUMERICAL CONTINUATION

#### 5.1 Introduction

In this chapter, an efficient prediction scheme for numerical continuation of systems of the form  $\vec{F} = \vec{0}$ , where  $\vec{F} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  is developed. The formulation does not require second order information, yields predictor directions consistent with the local geometry of the solution set and remains linear in complexity irrespective of the Jacobian structure. It can be implemented in any variant of path-tracking algorithms and uses stabilized local weighted least squares to fit a local polynomial on the last  $k \geq 2$  solution points, with the weighting function defined so that solution points closer to the current one are assigned higher weight. This leads to a versatile scheme where the order of approximation is decoupled from the points supplanted and can be easily adjusted and updated, if further criteria are enforced. Moreover, the prediction can be based either on extrapolation from the fitted polynomial or its tangent, which is trivial to compute. Since the solution set of such systems is a curve, the parameterization best suited for the proposed algorithm is the arc-length of the curve.

The chapter is structured as follows: in the first section we provide a summary numerical continuation. In the second section, the formulation of the Weighted Least Squares (WLS) prediction scheme is introduced, followed by a section dealing with implementation details. The chapter concludes with a section dedicated to numerical tests. We demonstrate that predictions based on the proposed scheme are competitive compared to more expensive approaches utilizing the tangent predictor in terms of total steps, function evaluations and linear solves. Furthermore, it is shown that they can facilitate the continuation of the tracking process even in cases where conventional prediction schemes lead to premature termination.

## 5.2 Numerical Continuation

We focus our attention to undetermined nonlinear systems of the form<sup>1</sup>:

$$\vec{F}(\vec{y}) = \vec{0} \quad (5.0)$$

where  $\vec{F} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$  is a sufficiently smooth map. Numerical continuation attempts to solve these systems, assuming that the regularity assumptions hold. We have already introduced the necessary theoretical concepts pertaining to such systems in the previous chapter. In what follows, we assume that  $\vec{F}$  is a sufficiently smooth map and  $\vec{0}$  is a regular value for system (5.2). Under these assumptions, the solution set  $\mathcal{S}$  is a one-dimensional manifold in  $\mathbb{R}^{n+1}$  consisting of connected components homeomorphic to the real line (paths) and the unit circle (loops) and with the same smoothness as  $\vec{F}$ . For  $\mathcal{S}$  we have:

$$\mathcal{S} = \{\vec{y} \in \mathbb{R}^{n+1} \mid \vec{F}(\vec{y}) = \vec{0}\}, \quad \text{and} \quad \text{rank}(\mathbf{J}) = n \quad \forall \vec{y} \in \mathcal{S} \quad (5.0)$$

where  $\mathbf{J}$  is the Jacobian of  $\vec{F}$ . Then, by the implicit function theorem there exists  $\vec{\phi}(s) : \mathbb{R} \rightarrow \mathbb{R}^{n+1}$  such that  $\vec{F}(\vec{\phi}(s)) = \vec{0}$ . In other words, paths or loops in  $\mathcal{S}$  implicitly defined by Eq. (5.2) can be parameterized with respect to  $s$  and, thus,  $\vec{y}(s) \equiv \vec{\phi}(s)$ . Differentiating Eq. (5.2) with respect to  $s$ , we arrive at the following Initial Value Problem (IVP):

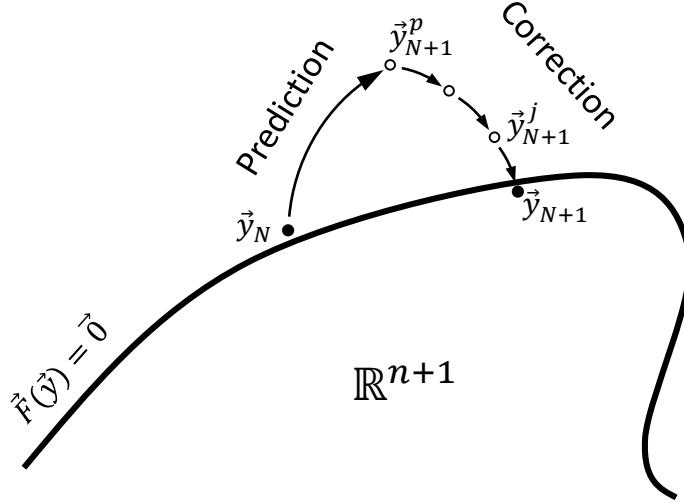
$$\begin{aligned} \mathbf{J}(\vec{y})\dot{\vec{y}} &= 0 \\ \vec{y}(0) &= \vec{y}_0, \quad \vec{F}(\vec{y}_0) = \vec{0} \end{aligned}$$

where  $(\cdot)$  denotes differentiation with respect to  $s$  and  $\dot{\vec{y}}$  is the tangent vector to the solution curve at  $\vec{y}(s)$ .

The implicit function theorem forms the basis of numerical continuation[64, 177, 65, 73,

---

<sup>1</sup>In this chapter, we use  $\vec{y} \in \mathbb{R}^{n+1}$  to denote any solution vector to an undetermined system  $\vec{F} = \vec{0}$  and should not be confused with the internal field vector  $\vec{y}_i$  introduced in Chapter 2, Sec. 2.4.



**Figure 5.1.** Advancing from step  $N$  to step  $N + 1$  using a PC approach.

168, 178, 66], which aim to track solution sets in  $\mathcal{S}$ . The idea is to generate a sequence of points  $\{\vec{y}_1, \vec{y}_2, \dots\}$ , with  $\vec{F}(\vec{y}_i) \approx \vec{0}$  by solving system (5.1) in a step-by-step fashion using a Predictor-Corrector (PC) scheme. The orientation of tracking is chosen by assigning a sign to the tangent vector and the positive orientation is characterized by condition  $\det([\frac{\mathbf{J}}{\dot{\vec{y}}}]) > 0$ . If  $s$  is assumed to be the arc-length of the solution curve, then  $\dot{\vec{y}}(s)$  is a unit tangent vector and system (5.1) is supplemented by condition  $\|\dot{\vec{y}}(s)\|_2 = 1$ , where  $\|\cdot\|_2$  denotes the Euclidian norm.

Different initial points  $\vec{y}_0$  might lead to a different path(loop) to be tracked and each point generated by PC methods is considered to be in the proximity of the solution curve. As shown in Fig. 5.1, to advance from the current step  $N^2$  with known solution  $\vec{y}_N$  to step  $N + 1$ , a prediction for  $\vec{y}_{N+1}$  is first made,  $\vec{y}_{N+1}^p$ , followed by a correction phase whereby the initial prediction is iteratively improved, usually by employing a Newton type iterative procedure to enforce condition (5.2). Satisfaction of this condition is usually tested using an appropriate norm and a prescribed tolerance.

Assuming that the path is parameterized by  $s$  so that  $\vec{y} = \vec{y}(s)$ , these two phases can be succinctly stated as follows:

---

<sup>2</sup>The use of  $N$  to designate the step counter in this chapter should not be confused with the same symbol used in Chapter 2 to represent the axial stress resultant in a cross-section.

1. Prediction:  $\vec{y}_{N+1}^p = \vec{G}(s_N + \Delta s)$
2. Correction:  $\vec{y}_{N+1}^{j+1} = \vec{y}_{N+1}^j + \mathbf{B}^j \vec{F}(\vec{x}_{N+1}^j), j = 0, 1, \dots, j_{max}, \vec{y}_{N+1}^0 = \vec{y}_{N+1}^p$

Both  $\vec{G}$  and  $\mathbf{B}$  depend on the predictor and corrector algorithms used. For example, if the Euler predictor is used, then  $\vec{G} = \vec{y}(s)$  and  $\vec{G}(s_N + \Delta s) \approx \vec{y}_N + \Delta s \dot{\vec{y}}_N$  around  $\vec{y}_N$ . In addition, if  $\mathbf{B} = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}$ , then the correction phase arrives at the least squares solution for  $\vec{y}_{N+1}$ , with  $\mathbf{B}$  being the Moore-Penrose inverse of  $\mathbf{J}$ . In general, the determining characteristics for a PC algorithm are:

- Choice of parameterization
- Type of predictor
- Type of corrector
- Determination and control of step-length  $\Delta s$

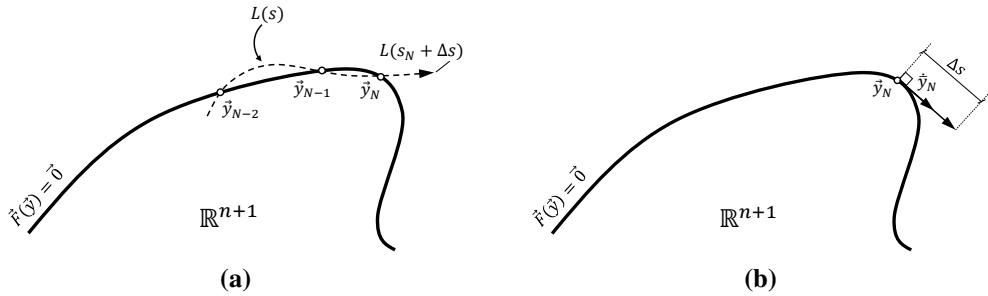
While the most computationally costly part in PC methods is the correction phase, efficient step-length adjustment schemes in conjunction with fast and reliable prediction phases can also contribute significantly to the performance of the PC algorithm.

Predictors can be categorized in three families:

1. Predictors based on functional extrapolation
2. Predictors based on derivative information of  $\vec{F}$
3. A combination of 1 and 2

As an example of type 1 prediction, a Lagrange interpolating polynomial,  $L(s)$ , can be used to extrapolate from a set of the  $k$  most recent solution points  $\{\vec{y}_{N-k+1}, \dots, \vec{y}_N\}$  such that  $\vec{y}_{N+1}^p = L(s_N + \Delta s)$ [177, 179]. The Euler predictor using the tangent  $\dot{\vec{y}}_N$  as a direction is the simplest case of the second type. Finally, a case of type 3 predictors are those derived from interpolations that utilize derivative information, such as the Hermite interpolating polynomials. In Fig. 5.2 the two generic cases of predictors are illustrated.

In general, predictors based on extrapolation of an interpolating function tend to be sensitive to variations in accuracy of the computed set  $\{\vec{y}_i\}_{i=1}^N$  and although they are computationally in-



**Figure 5.2.** Two types of predictors: (a) interpolatory and (b) Euler predictor.

expensive, the quality of prediction worsens as the step-length becomes larger. For that reason, predictors of order higher than two are rarely used[180, 181]. On the other side, type 2 or 3 predictors[182, 183, 184, 185, 186, 179, 187], tend to yield better approximations for larger step-lengths but are considerably more expensive. Tangent vectors need to be extracted from the nullspace of  $\mathbf{J}$  at the beginning of each step and the determination of higher order derivatives require additional linear systems to be solved. Alternatively, finite difference schemes can be employed to approximate higher order quantities, but these too require additional functional evaluations. An alternative way to categorize predictors is based on whether they rely solely on information at  $\vec{y}_N$  or incorporate previous steps as well[188]. Since the predictor developed in this chapter is by construction a multistep formulation, the categorization presented here will help highlight type 1 and type 2 predictor variants that arise from the proposed WLS scheme.

### 5.3 Weighted Least Squares Predictor

In this section we formulate the Weighted Least Squares (WLS) predictor. Let  $k, m \in \mathbb{Z}^+, k > m$ , be parameters that designate the last  $k$  steps, including the current step,  $N$ , and the degree of polynomial space,  $\mathcal{P}_m(\mathbb{R}_{\geq 0}, \mathbb{R})$ , respectively. We assume convergence has been achieved at step  $N$  and all relevant quantities for the last  $k$  steps are known. For ease of notation, we restrict our presentation to a particular component of  $\vec{y} = [y_1 \cdots y_l \cdots y_{n+1}]^T$ ,  $y_l$ , and omit the index from derivations. Therefore, in what follows, quantities  $y_N, y_{N-1}\mathbb{R}$  are to be understood as the  $l$ -th components of  $\vec{y} \in \mathbb{R}^{n+1}$  at steps  $N$  and  $N - 1$  respectively unless explicitly stated

otherwise.

Let  $\bar{y}(s) \in \mathcal{P}_m$  be the fitted polynomial at the last  $k$  points  $\{y_i\}_{i=N-k+1}^N$  using a WLS approach, where  $y_i = y(s_i)$ , such that:

$$\bar{y}(s) = b_0 + b_1 s + \cdots + b_m s^m = \vec{p}^T \vec{b} \quad (5.1)$$

where  $\vec{p} = [1 \ s \ \dots \ s^m]^T$  and  $\vec{b} = [b_0 \ b_1 \ \dots \ b_m]^T$  are the polynomial basis and coefficient vectors respectively. The “energy” function associated with the WLS fitting in this case is:

$$J = \sum_{j=N-k+1}^N w(s_j) [\bar{y}(s_j) - y_j]^2 \quad (5.1)$$

where  $w(s)$  is the weight function, unspecified for now. Minimization of  $J$  with respect to  $\vec{b}$  yields the so-called normal equations for the least squares solution:

$$\mathbf{A} \vec{b} = \vec{f} \quad (5.1)$$

with Eq. (5.3) substituted in (5.3). The moment matrix  $\mathbf{A}$  and “force” vector  $\vec{f}$  are given by:

$$\begin{aligned} \mathbf{A} &= \sum_{j=N-k+1}^N w_j \vec{p}(s_j) \vec{p}(s_j)^T \\ \vec{f} &= \sum_{j=N-k+1}^N w_j y_j \vec{p}(s_j) \end{aligned}$$

Since the path is parameterized by the arc-length  $s$ , which is taken here as the independent parameter of the WLS fitting, a stabilization measure is required for the case that  $s$  becomes excessively large. In such a scenario, the moment matrix  $\mathbf{A}$  becomes ill-conditioned. To deal with this issue, we transform the polynomial basis and work instead in  $\mathcal{P}_m([-1, 1], \mathbb{R})$ . That is:

$$\bar{y}(s) = \vec{c}^T \vec{\rho} \quad (5.2)$$

where now  $\vec{\rho} = [\rho_0 \ \rho_1 \ \dots \ \rho_m]$  is the coefficient vector for the transformed basis and  $\vec{c}$  is:

$$\vec{c} = \left[ 1 \ \frac{s - s_r}{h} \ \left( \frac{s - s_r}{h} \right)^2 \ \dots \ \left( \frac{s - s_r}{h} \right)^m \right]^T \quad (5.2)$$

with  $s_r = (s_N + s_{N-k+1})/2$ ,  $h = (s_N - s_{N-k+1})/2$  and  $s \in [s_{N-k+1}, s_N]$ . The transformation between the basis vectors  $\vec{p}$  and  $\vec{c}$  is carried out by operator  $\Phi$ :

$$\vec{c} = \Phi \vec{p} \quad (5.2)$$

where

$$\Phi = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -\frac{s_r}{h} & \frac{1}{h} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ \frac{(-s_r)^m}{h^m} & \binom{m}{m-1} \frac{(-s_r)^{m-1}}{h^m} & \dots & -\binom{m}{1} \frac{s_r}{h^m} & \frac{1}{h^m} \end{bmatrix} \quad (5.2)$$

Similarly, for the coefficient vectors we have:

$$\vec{b} = \Phi^T \vec{\rho} \quad (5.2)$$

Using Eq. (5.3) and  $\vec{p} = \Phi^{-1} \vec{c}$  from Eq. (5.3), the normal equations in the transformed basis are:

$$\mathbf{A}_c \vec{\rho} = \vec{f}_c \quad (5.2)$$

where  $\mathbf{A}_c$ ,  $\vec{f}_c$  are determined from Eq. (5.2) with  $\vec{p}(s_j) \rightarrow \vec{c}(s_j)$ . Using Eqs. (5.3), (5.3), and that  $\vec{\rho} = \mathbf{A}^{-1} \vec{f}_c$  from Eq. (5.3), the stabilized weighted least squares fit for component  $y$  in the original space is given by:

$$\bar{y}(s) = \vec{p}(s)^T \vec{b} \quad (5.2)$$

where

$$\vec{b} = \Phi^T \mathbf{A}_c^{-1} \vec{f}_c \quad (5.2)$$

Notice that  $\mathbf{A}_c$  is common for all components  $y_l$ ,  $l = 1, \dots, n+1$  in  $\vec{y}$  and only the forcing the “force” vector  $\vec{f}_c$  has to be evaluated for each component. That is, only the coefficient vector is component dependent:  $\vec{b} \rightarrow \vec{b}(y_{N-k+1}, \dots, y_N)$ .

### 5.3.1 Matrix Formulation of WLS Approximation

We now derive the matrix relations of the WLS process, which incorporate all components. This can be represented succinctly by collecting all the involved quantities corresponding to different components of  $\vec{y}$  into matrices. The local approximation to solution curve  $\vec{y}(s)$  is as follows:

$$\vec{y} \approx \bar{y}(s) = \begin{bmatrix} \bar{y}_1(s) \\ \bar{y}_2(s) \\ \vdots \\ \bar{y}_{n+1}(s) \end{bmatrix}$$

which holds for  $s \in [s_{N-k+1}, s_N]$  and all solution points corresponding to that interval<sup>3</sup>. Let  $\mathbf{P}$  be defined as follows:

$$\mathbf{P} = \begin{bmatrix} \vec{p}(s_{N-k+1}) & \dots & \vec{p}(s_N) \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 \\ s_{N-k+1} & \dots & s_N \\ \vdots & \ddots & \vdots \\ s_{N-k+1}^m & \dots & s_N^m \end{bmatrix} \quad (5.2)$$

with its counterpart in the transformed basis given by

$$\mathcal{C} = \Phi \mathbf{P} \quad (5.2)$$

where  $\mathcal{C}$  is defined in accordance with  $\mathbf{P}$ :  $\mathcal{C} = \begin{bmatrix} \vec{c}(s_{N-k+1}) & \dots & \vec{c}(s_N) \end{bmatrix}$ . We also define  $\mathbf{W} = \text{diag}(w_{N-k+1}, \dots, w_N)$ . From Eqs. (5.3), (5.2) and (5.3), the coefficient vector for

---

<sup>3</sup>Note that  $\bar{y}(s)$  is a vector. It denotes an approximation to vector  $\vec{y}$ . The overhead arrow is omitted in order to avoid excessive use of notation.

component  $l$  in  $\bar{\vec{y}}, \vec{b}_l$ , can be expressed in the following way:

$$\vec{b}_l = \Phi^T \mathbf{A}_c^{-1} \mathcal{C} \mathbf{W} \begin{bmatrix} y_{N-k+1,l} \\ y_{N-k+2,l} \\ \vdots \\ y_{N,l} \end{bmatrix}_{(k \times 1)} \quad (5.2)$$

By extension, a concise expression for  $\bar{\vec{y}}$ , whose component form is given by Eq. (5.3), is then:

$$\bar{\vec{y}} = \vec{p}(s)^T \mathcal{B} \quad (5.2)$$

where

$$\mathcal{B} = \Phi^T \mathbf{A}_c^{-1} \mathcal{C} \mathbf{W} \mathbf{Y} \quad (5.2)$$

and  $\mathbf{Y}$  is a  $k \times (n + 1)$  matrix whose  $l$ -th column contains the values of the  $l$ -th component of  $\vec{y}$  at the last  $k$  points supplied to the algorithm (see Eq. (5.3.1)):

$$\mathbf{Y} = \begin{bmatrix} y_{N-k+1,1} & y_{N-k+1,2} & \dots & y_{N-k+1,n+1} \\ y_{N-k+2,1} & y_{N-k+2,2} & \dots & y_{N-k+2,n+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N,1} & y_{N,2} & \dots & y_{N,n+1} \end{bmatrix} \quad (5.2)$$

We observe from Eq. (5.3.1) (or (5.3)) that derivatives up to order  $m$  can be computed directly by simple differentiation of the polynomial basis vector  $\vec{p}(s)$ :

Component form	Vector form
$\dot{y} = \dot{\vec{p}}(s)^T \vec{b},$	$\dot{\vec{y}} = \dot{\vec{p}}(s)^T \mathcal{B}$
$\ddot{y} = \ddot{\vec{p}}(s)^T \vec{b},$	$\ddot{\vec{y}} = \ddot{\vec{p}}(s)^T \mathcal{B}$
$\vdots$	$\vdots$

with

$$\begin{aligned}\dot{\vec{p}}(s) &= \begin{bmatrix} 0 & 1 & 2s & \dots & ms^{m-1} \end{bmatrix} \\ \ddot{\vec{p}}(s) &= \begin{bmatrix} 0 & 0 & 2 & \dots & m(m-1)s^{m-2} \end{bmatrix} \\ &\vdots\end{aligned}$$

### 5.3.2 WLS Prediction Variants

Given an increment in the arc-length  $\Delta s$ , a straightforward prediction scheme is given by extrapolating from the fitted curve,  $\bar{\vec{y}}$  at  $s_N$ :

$$\vec{y}^p = \bar{\vec{y}}(s_N + \widetilde{\Delta s}) \quad (5.-2)$$

where  $\widetilde{\Delta s}$  is introduced to account for the fact that  $s$  is not actually the arc-length of the solution curve  $\vec{y}(s)$  and use of  $\Delta s$  can sometimes lead to unpredictable behavior. A stabilization procedure for is presented in the next part of this section.

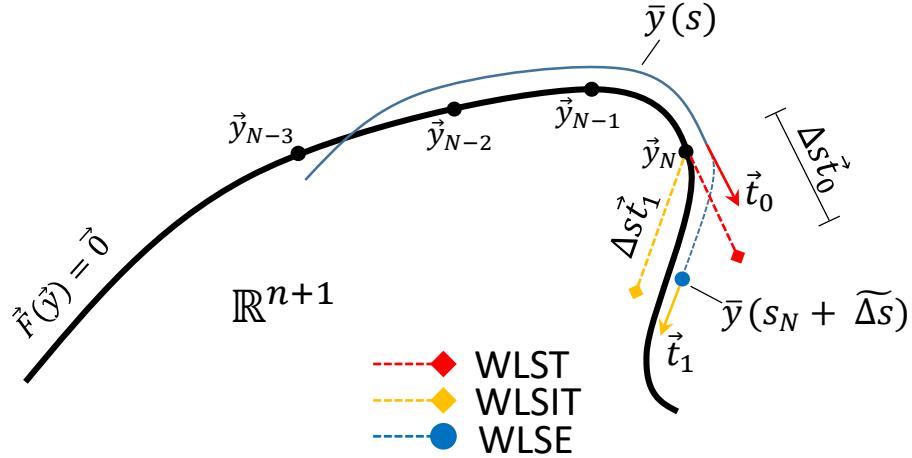
We will refer to prediction given by Eq. (5.3.2) as Weighted Least Squares Extrapolation (WLSE) predictor. A second approach is to derive the prediction direction from first order information of the local WLS curve, which is trivial to retrieve. We propose the following general formula:

$$\vec{y}^p = \vec{y}_N + \Delta s \vec{t}_z \quad (5.-2)$$

where  $\vec{t}_z$  is a unit vector determined as follows:

$$\vec{t}_z = \frac{\dot{\vec{y}}(s_N + z\widetilde{\Delta s})}{\|\dot{\vec{y}}(s_N + z\widetilde{\Delta s})\|_2} \quad (5.-2)$$

with  $\dot{\vec{y}}(s) = \dot{\vec{p}}(s)^T \mathcal{B}$  and  $z \in [0 1]$ . For  $z \neq 0$ , the direction is derived by extracting the tangent of the fitted curve ahead of the current solution point, whereas for  $z = 0$  the normalized tangent of  $\vec{y}(s_N)$  is used instead and this yields a WLS approximation to the standard Euler predictor.



**Figure 5.3.** Geometric interpretation of predictors (i)WLSE (blue), (ii) WLSIT (light blue,  $z = 1$ ), and (iii) WLST (red,  $z = 0$ ).

Normalization is required because the  $\dot{\vec{y}}$  is not a unit vector. When  $z > 0$ , will refer to this prediction scheme as the Weighted Least Squares Implicit Tangent (WLSIT) predictor. In the case that  $z = 0$ , we term it Weighted Least Squares Tangent (WLST) predictor. Notice that, in any case, WLSIT involves extrapolation and, therefore, the steplength adjustment procedure for  $\tilde{\Delta}s$  is required here as well. For illustrative purposes, the predictors WLSE, WLST and WLSIT with  $z = 1$  are depicted in Fig. 5.3.

### 5.3.3 Arc-length Stabilization

Given a step-length  $\Delta s$ , the WLSE predictor estimates the next solution point using Eq. (5.3.2). To determine  $\tilde{\Delta}s$  given  $\Delta s$ , we impose condition Eq. (5.3.3):

$$\|\vec{y}^p - \vec{y}_N\|_2^2 - \Delta s^2 = 0 \quad (5.-2)$$

where in place of  $\vec{y}^p$  (see Eq. (5.3.2)) we use its first order approximation:

$$\vec{y}^p \approx \vec{y}(s_N) + \dot{\vec{y}}(s_N)\tilde{\Delta}s \quad (5.-2)$$

Let  $\vec{y}_0 = \vec{y}_N$ ,  $\bar{y}_0 = \bar{y}(s_N) = \vec{p}(s_N)^T \mathcal{B}$  and  $\dot{\bar{y}}_0 = \dot{\bar{y}}(s_N) = \dot{\vec{p}}(s_N)^T \mathcal{B}$ . Then, substituting Eq. (5.3.3) into (5.3.3) yields the following quadratic equation to be solved for  $\widetilde{\Delta s}$ :

$$\|\dot{\bar{y}}_0\|_2^2 (\widetilde{\Delta s})^2 + 2(\bar{y}_0 - \bar{y}_0)^T \dot{\bar{y}}_0 \widetilde{\Delta s} + \|\bar{y}_0 - \bar{y}_0\|_2^2 - \Delta s^2 = 0 \quad (5.2)$$

This leads to two roots, of which the smallest positive one is chosen:

$$\widetilde{\Delta s}_{1,2} = \frac{-2(\bar{y}_0 - \bar{y}_0)^T \dot{\bar{y}}_0 \pm \sqrt{\Delta}}{2\|\dot{\bar{y}}_0\|_2^2} \quad (5.2)$$

where the discriminant is  $\Delta = 4\|\bar{y}_0 - \bar{y}_0\|_{\mathbf{N}}^2 - 4\|\dot{\bar{y}}_0\|_2^2(\|\bar{y}_0 - \bar{y}_0\|_2^2 - \Delta s^2)$ , with  $\mathbf{N} = \dot{\bar{y}}_0 \dot{\bar{y}}_0^T$ .

Condition  $\Delta > 0$  leads to the following requirement for the user-defined step-length  $\Delta s$  if the WLSE predictor is used:

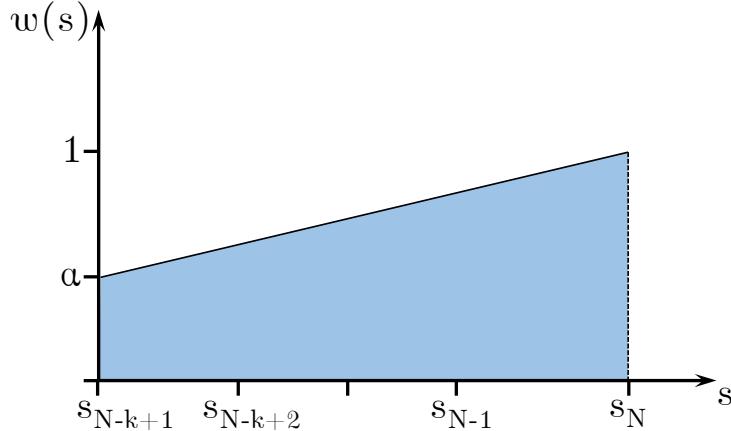
$$\Delta s > \sqrt{\|\bar{y}_0 - \bar{y}_0\|_2^2 - \frac{\|\bar{y}_0 - \bar{y}_0\|_{\mathbf{N}}^2}{\|\dot{\bar{y}}_0\|_2^2}} \quad (5.2)$$

### 5.3.4 The Weight Function

We now discuss the weighting function  $w(s)$ . The reasoning for introducing a weighting in the least squares fitting is as follows: since with the “exact” tangent predictor the direction is determined by the neighborhood of the current solution  $\vec{y}_N$  in the differential sense, we need to assign different weights to the preceding  $k - 1$  solution points we provide for the least squares fitting. A reasonable option for  $w(s)$ , which is depicted in Fig. 5.4, is a generalization of the unit ramp function:

$$w(s) = \alpha + \frac{s - s_{N-k+1}}{s_N - s_{N-k+1}}(1 - \alpha), \quad \alpha \in [0, 1]$$

By choosing  $\alpha = 0$ , the unit ramp function is obtained, while for  $\alpha = 1$ , all solution points are assigned unit weight,  $w(s) = 1$ , with  $s \in [s_{N-k+1}, s_N]$ . With this weighting plan, it is ensured points closer to the current solution are assigned higher importance, with  $\vec{y}_N$  having the highest



**Figure 5.4.** Weight function  $w(s)$  of the WLS predictor.

weight by construction. If the unit ramp is used, it is crucial that the implementation ensures there are enough points for the least squares algorithm to commence. This is because the point that corresponds to step  $N - k + 1$  is now assigned a zero weight and, thus, it doesn't contribute in the moment matrices  $\mathbf{A}$ ,  $\mathbf{A}_c$  (see Eqs. (5.3), (5.2)). This issue arises when the minimum points for the desired degree of approximation are supplied.

## 5.4 Implementation

### 5.4.1 Algorithm

We outline the basic steps in the implementation of the WLS predictor in pseudocode format. We assume all quantities at the current step  $N$  are known.

As can be seen from Algorithm 1, the WLS process requires the polynomial degree  $m$  we use for the fitting, the number of solution points  $k$  we take into account starting from the current one at step  $N$ , along with the curve length up to these points. We also need to specify values for parameter  $\alpha$  (see Eq. (5.3.4)) and  $z$ , in case the WLSIT predictor is selected. Finally, a step-length,  $\Delta s$ , also needs to be provided. The most expensive action is the computation of the coefficient matrix  $\mathcal{B}$ . More specifically, the number of floating point operations is commensurate to the term  $2(m+1)k(n+1)$  or, simply,  $2(m+1)kn$ , where  $n$  is the size of  $\vec{y}$  after ignoring one component. In other words, it is of  $\mathcal{O}(n)$  (time) complexity. Coupled with a degree control

---

**Algorithm 1:** Weighted Least Squares Predictor pseudocode

---

**Input:**  $m, k, \{y_i\}_{i=N-k+1}^N, \{s_i\}_{i=N-k+1}^N, \Delta s, \alpha, z$   
Determine  $\Phi, \mathbf{Y}$  and initialize  $\mathbf{A}_c, \mathbf{W}, \mathcal{C}$

**for**  $i = k, k-1, \dots, 1$  **do**

- $s_i \leftarrow s_{N-i+1}$
- Get shifted basis at  $s_i$ ,  $\vec{c}(s_i)$ , and add to  $i$ -th column in  $\mathcal{C}$
- Determine weight,  $w_i$ , and add it to component  $(i, i)$  in  $\mathbf{W}$
- $\mathbf{A}_c \leftarrow \mathbf{A}_c + w_i \vec{c}(s_i) \vec{c}(s_i)^T$

**end for**

$\mathcal{B} \leftarrow \Phi^T \mathbf{A}_c^{-1} \mathcal{C} \mathbf{W} \mathbf{Y}$

Determine  $\tilde{\Delta s}$  *// Step-length stabilization*

*// WLSE*

$s \leftarrow s_N + \tilde{\Delta s}$

Get polynomial basis at  $s$ ,  $\vec{p}(s)$

$\vec{y}_{N+1}^p \leftarrow \vec{p}(s)^T \mathcal{B}$

*// WLSIT*

$s \leftarrow s_N + z \tilde{\Delta s}$

Get first derivative of polynomial basis at  $s$ ,  $\dot{\vec{p}}(s)$

$\vec{v} \leftarrow \dot{\vec{p}}(s)^T \mathcal{B}$

$\vec{t}_z \leftarrow \vec{v} / \|\vec{v}\|_2$

$\vec{y}_{N+1}^p \leftarrow \vec{y}_N + \Delta s \vec{t}_z$

**return**  $\vec{y}_{N+1}^p$

---

check, whereby  $m$  can be adjusted based on local properties of the path tracked so far, we expect high values for the coefficient only in highly nonlinear regions.

#### 5.4.2 Arc-Length Measurement

Prediction schemes that rely on some form of extrapolation require measurement of the current curve length,  $s$ , as the solution process progresses. The most direct and easy way to compute the curve length at discrete points is by summation of lengths of the line segments connecting them. After convergence is achieved at step  $N$ , the length of the segment between  $\vec{y}_{N-1}$  and  $\vec{y}_N$  is approximated as follows:

$$s_N = s_{N-1} + Ds \quad (5.-2)$$

where

$$Ds = \|\vec{y}_N - \vec{y}_{N-1}\|_2 \quad (5.-2)$$

If Eq. (5.2) represents the finite-dimensional approximation of some operator equation, then it is recommended[189] that we account for the fact that, as is customary, the first  $n$  components are members of some function space (e.g. a Banach space) and, therefore, Eq. (5.4.2) should in some sense approximate the underlying  $L^2$  norm:

$$Ds = \|\vec{y}_N - \vec{y}_{N-1}\|_{L^2} = \sqrt{h_m^d \sum_{i=1}^n (y_{i,N} - y_{i,N-1})^2 + (y_{n+1,N} - y_{n+1,N-1})^2} \quad (5.-2)$$

where the  $n+1$  component in  $\vec{y}$  is considered as an additional parameter and  $h_m, d$  are the mesh spacing and the spatial dimension of the problem respectively.

### 5.4.3 Discussion

It is clear that, in general,  $\bar{y}(s_j) \neq \vec{y}_j$  for  $j = N - k + 1, \dots, N$ , which is expected since this is not, necessarily, an interpolatory approximation. The interpolatory condition can be retained if the relation between the number of solution points supplied,  $k$ , and the degree of the polynomial space,  $m$ , is  $k = m + 1$ . Furthermore, the degree of approximation is uncoupled from the number of points supplied, while in conventional interpolatory schemes this is not the case. We suspect this is one of the reasons some researchers claim higher order predictors based on extrapolation generally behave poorer than the secant predictor[180, 181]. In addition, the WLS predictor does not rely on derivative information of  $\vec{F}$  and, therefore, its performance in terms of computational cost is independent of the structure of the Jacobian. This is particularly important for problems without properties like sparsity or bandedness to exploit numerically. For the WLSIT variant, we instead use the local WLS approximation to extract derivatives when and if necessary. For step-length,  $\Delta s$ , of reasonable magnitude, the local WLS fitting is able to give a better approximation of the local geometry, especially closer to sudden changes

in curvature of the solution curve. This is precisely due to the fact that interpolation is not enforced. Furthermore, since extraction of the actual tangent vector from the nullspace of the Jacobian is not required, WLS predictions can couple well with correctors that also bypass Jacobian evaluations, such as nonlinear conjugate gradient corrections[188]. However, in that case, secondary branches due to bifurcation will have to be tracked using a (global) perturbation technique. Since differentiation of the WLS curve is trivial, the tangent vectors to the solution curve at solution points can be approximated by the normalized tangents to the local WLS curve.

Another attractive feature of the proposed formulation is that  $m$  and  $k$  can be adjusted during the continuation process. This provides additional options as far as calibration of the algorithm near turning points is concerned. That is, instead of reducing the step-length,  $m$  or  $k$  can instead be increased in order to represent the local curvature better. In contrast, when the curvature is small, switching to a linear basis ( $m = 1$ ) is usually better. Furthermore, for step-lengths not excessively large, the tangent and curvature on the local WLS curve can be determined analytically and treated as approximations to those of the actual solution curve, provided the polynomial basis is of adequate order. In addition, parameter  $z$  involved in the WLSIT variant can also be varied depending on the local geometry. It has been observed that, if only  $\vec{y}_N$  is located past a turning point, using the WLSIT predictor with  $z = 1$ , or even  $z > 1$  yields much better predictions compared to both WLSE and WLST. Conversely, if  $\vec{y}_{N-k+1}$  lies before a turning point and the rest are located beyond it, then WLST gives better predictions and in most cases, using tangents at  $s < s_N$  yield even better ones (case  $z < 0$ ).

Since at least 2 points need to be passed to the WLS predictor, the first 2 steps of the continuation are carried out using the tangent predictor or an approximation of it. In general, at any step there need to be at least  $k$  solution points available to pass to the WLS predictor. Moreover, prediction schemes based on extrapolation tend to be sensitive to measurement of the arc-length of the solution curve (see Eq. (5.4.2)) and the approximation quality of solution points[190, 191]. Although this can affect the prediction quality of the WLSE variant for larger prescribed step-lengths, the WLSIT is generally more stable due to normalization imposed on

the tangent to the local WLS curve.

## 5.5 Numerical Examples

We now illustrate the performance of the proposed predictor formulation with five examples.

We use both WLS variants described in 5.3, where we also showcase results for both the WLST and WLSIT predictors ( $z = 0$  and  $z = 1$  respectively). For each WLS predictor we also include a selection of cases with regards to parameters  $(m, k)$ , the order of polynomial basis and the number of points passed to the algorithm. For comparison purposes, we also include results using i) Euler predictor with the exact tangent vector and ii) two-step and three-step Adams-Bashforth predictors[190]. For reference, these predictors are given by the following formulas<sup>4</sup>:

- Euler predictor:  $y^p = y_N + \Delta s \dot{y}_N$
- Two-step Adams-Bashforth (AB2):  $y^p = y_N + \Delta s \frac{1}{2} (3\dot{y}_N - \dot{y}_{N-1})$
- Three-step Adams-Bashforth (AB3):  $y^p = y_N + \Delta s \frac{1}{12} (23\dot{y}_N - 16\dot{y}_{N-1} + 5\dot{y}_{N-2})$

While step-length  $\Delta s$  control and adjustment is an important aspect of numerical continuation, its influence on the overall performance is not considered here and in each problem a constant step-length is prescribed. The results are listed in tables, where we use the following abbreviations:

- Ns Number of points computed on the solution curve (steps)
- Nf Number of function evaluations
- NI Number of solved linear systems
- Ni Maximum number of iterations that occurred in a step

For each problem and for each prediction scheme, we perform two analyses: one with step-length that is considered small and one that is large. The specific values for  $\Delta s$  considered small or large are problem-dependent. In all problems, however, we considered the  $n + 1$ -th variable

---

<sup>4</sup>Expressions for a one-dimensional problem.

to be the varying “homotopy” parameter, which we designated  $t$ . In that sense, system (5.2) can be recast as follows<sup>5</sup>

$$\vec{\mathbf{F}}(\vec{x}, t) = \vec{0} \quad (5.2)$$

and its Jacobian, considered to have maximal rank at all points, is  $\mathbf{J} = \begin{bmatrix} \mathbf{F}_x & \mathbf{F}_t \end{bmatrix}$ . Data such as tolerance  $\epsilon_{tol}$ , initial value for  $t$ ,  $t_0$ , and its maximum prescribed value,  $t_{max}$  are included in the table caption for each problem.

### 5.5.1 Numerical Example 1

This example is taken from Kubíček[192], where the dependence of a system solution to a varying parameter is explored. The nonlinear system in question is  $\vec{\mathbf{F}}(\vec{x}, t) = \vec{0}$ ,  $F : \mathbb{R}^4 \times [0, 1] \rightarrow \mathbb{R}^4$ , with  $\vec{\mathbf{F}} = [f_1 \ f_2 \ f_3 \ f_4]^T$  and:

$$\begin{aligned} f_1 &= t(1 - x_3)\exp\left(\frac{10x_1}{1 + 10x_1/\gamma}\right) - x_3 \\ f_2 &= tB(1 - x_3)\exp\left(\frac{10x_1}{1 + 10x_1/\gamma}\right) + \beta_1\theta_{c1} - 10(1 + \beta_1)x_1 \\ f_3 &= x_3 - x_4 + t(1 - x_4)\exp\left(\frac{10x_2}{1 + 10x_2/\gamma}\right) \\ f_4 &= 10x_1 - 10(1 + \beta_2)x_2 + tB(1 - x_4)\exp\left(\frac{10x_2}{1 + 10x_2/\gamma}\right) + \beta_2\theta_{c2} \end{aligned}$$

and values for parameters  $\gamma$ ,  $B$ ,  $\beta_1$ ,  $\beta_2$ ,  $\theta_{c1}$ ,  $\theta_{c2}$  are taken from the aforementioned investigation. The continuation process is initialized from point  $x_0 = (0, 0, 0, 0)$  with corresponding  $t_0 = 0$ . Parameter  $\alpha$  in the weight function (see Eq. (5.3.4)) is taken equal to 0.2. The correction algorithm used here is the one proposed in Chapter 4, Sec. 4.4. The results for this problem are listed in Table 5.1.

---

<sup>5</sup>Again, the use of vector  $\vec{x} \in \mathbb{R}^n$  should not be confused with the position vector of the neutral axis in the deformed configuration, as was the case in Chapter 2.

**Table 5.1** Results for Example 1,  $\epsilon_{tol} = 10^{-6}$ ,  $t_0 = 0$ ,  $t_{max} = 1$ .

Predictor	$(m, k)$	$\Delta s = 0.05$				$\Delta s = 0.2$			
		Ns	Nf	Nl	Ni	Ns	Nf	Nl	Ni
Euler	-	99	413	434	3	-	-	-	-
WLSE	(2, 4)	99	428	460	3	21	177	270	30
	(3, 5)	99	424	450	3	29	180	242	14
WLST	(2, 4)	99	408	420	3	-	-	-	-
	(3, 5)	99	408	418	3	-	-	-	-
WLSIT	(2, 4)	99	429	462	3	17	135	202	30
	(3, 5)	99	428	458	3	-	-	-	-
AB2	-	99	403	414	3	-	-	-	-
AB3	-	99	404	416	3	-	-	-	-

### 5.5.2 Numerical Example 2

In this example we use a Newton homotopy imbedding to find all stationary points of the Him-melblau function:

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (5.2)$$

with  $-5 \leq x_i \leq 5$ ,  $i = 1, 2$ . At a stationary point the gradient of  $f$  is zero:

$$\nabla f(x) = \begin{bmatrix} 2x_1^3 + 2x_1x_2 - 21x_1 + x_2^2 - 7 \\ x_1^2 + 2x_1x_2 + 2x_2^3 - 13x_2 - 11 \end{bmatrix} = 0 \quad (5.2)$$

and numerical continuation is used on the following homotopy imbedding:

$$\vec{F}(x, t) = \nabla f(x) - (1 - t)\nabla f_0 \quad (5.2)$$

where  $\nabla f_0$  is the value of the gradient at the initial point provided,  $x_0$ , which we take to be  $x_0 = (4.81, -4.81)$ . In artificial imbeddings such as the Newton homotopy (5.5.2), a root for the target system is found each time  $t$  attains the value 1, assuming no backtracking happens. Equation (5.5.2) locates nine roots for problem (5.5.2) in the prescribed domain, provided in Table 5.3, and we examine the how the listed prediction schemes performed in finding all of

**Table 5.2** Results for Example 2,  $\epsilon_{tol} = 10^{-6}$ ,  $t_0 = 0$ ,  $t_{max} = 1.2$ .

Predictor	$(m, k)$	$\Delta s = 0.1, \alpha = 0.1$				$\Delta s = 1.0, \alpha = 0.1$				$\Delta s = 1, 0, \alpha = 0.5$			
		Ns	Nf	Nl	Ni	Ns	Nf	Nl	Ni	Ns	Nf	Nl	Ni
Euler	-	304	1210	1209	2	-	-	-	-	-	-	-	-
WLSE	(2, 3)	304	1207	1201	2	-	-	-	-	-	-	-	-
	(2, 4)	304	1211	1207	2	49	323	451	30	-	-	-	-
	(2, 5)	304	1211	1207	2	-	-	-	-	-	-	-	-
	(3, 5)	304	1142	1067	2	32	157	185	4	32	156	183	4
WLST	(2, 3)	304	1209	1205	2	-	-	-	-	-	-	-	-
	(2, 4)	304	1209	1203	2	49	325	455	30	65	436	613	30
	(2, 5)	304	1213	1209	2	-	-	-	-	-	-	-	-
	(3, 5)	304	1212	1207	2	32	181	233	30	-	-	-	-
WLSIT	(2, 3)	304	1209	1205	2	-	-	-	-	-	-	-	-
	(2, 4)	304	1211	1207	2	-	-	-	-	-	-	-	-
	(2, 5)	304	1210	1203	2	-	-	-	-	-	-	-	-
	(3, 5)	304	1211	1205	2	32	153	177	4	32	153	177	4
AB2	-	304	1200	1189	2	32	145	167	4	32	145	167	4
AB3	-	304	1086	961	2	32	143	163	4	32	143	163	4

them. The results are listed in Table 5.2, where we used two different weightting paramterers,  $\alpha$ , for the larger step-length. Only analyses that i) found all 9 solutions and ii) converged to  $t = t_{max} = 1.2$  were deemed successfull. The same correction scheme was used here as in Example 1.

The cases WLSE(2, 4) and WLST(2, 4) with  $a = 0.1$  and WLST(2, 4)  $a = 0.5$ , passed through all solutions but also backtracked before eventually converging. The case of WLSE(2, 5),  $a = 0.5$ , while it converges, it passes through only 5 solutions and, therefore, it is not considered

**Table 5.3** Stationary points of Eq. (5.5.2).

Root	Coordinates			Coordinates		
	$x_1$	$x_2$	Root	$x_1$	$x_2$	
1	3.5856	-1.8546	6	-0.1282	-1.9697	
2	3.3844	0.0826	7	-3.7792	-3.2829	
3	3.0059	1.9860	8	-3.0733	-0.0819	
4	0.0867	2.8841	9	-2.8050	3.1311	
5	-0.2681	-0.9329				

a successfull analysis. We note that, due to the excessively large step-length of  $\Delta s = 1$  for this problem, only the cubic WLS predictors manage to perform on par with the Adams-Bashforth ones.

### 5.5.3 Numerical Example 3

We consider the following fixed point problem[193, 194, 182]:

$$F_i(x, t) = x_i - t \left[ \cos\left(i \sum_{j=1}^{10} x_j\right) \right] = 0 \quad (5.-2)$$

or  $F(x, t) = 0$ ,  $F : \mathbb{R}^{11} \rightarrow \mathbb{R}^{10}$ . This is solved using the Probability-one homotopy[195, 71] for system (5.5.3):

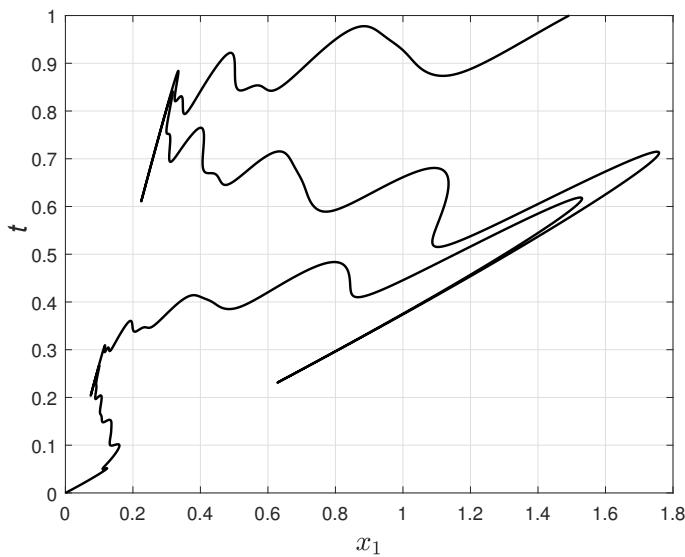
$$\vec{H}(\vec{x}, t) = t \vec{G}(\vec{x}) + (1 - t)(\vec{x} - \vec{x}_0) \quad (5.-2)$$

The resulting solution path exhibits more than 40 turning points with respect to  $t$  and is considered very difficult to solve. Starting from  $(\vec{x}_0, t_0) = (\vec{0}, 0) \in \mathbb{R}^{11}$ , we follow the path until the first intersection with hyperplane  $t = 1$ . The so-called Normal Flow corrector[178] is used in this test case, where the Moore-Penrose inverse of the  $n \times (n + 1)$  Jacobian of Eq. (5.5.3) is used during the Newton iterations.

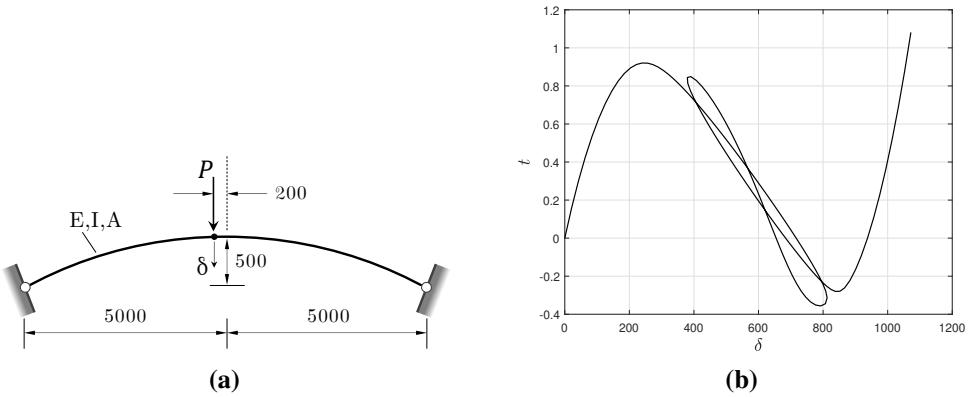
The results are shown in Table 5.4, where now for both step-lengths the optimal performance of the WLS predictors with respect to variations in the weighting parameter  $\alpha$  are listed. This is because even for the smaller  $\Delta s$ , convergence was difficult to achieve. For reference, in an investigation[194] of the same problem with the same arc-length parameterization but with step-length control, the smallest reported step-length was 0.000195 and on average, it was between 0.1 and 0.2. The solution path of problem (5.5.3), projected on the  $t - x_1$  plane, is shown in Fig. 5.5.

**Table 5.4** Results for Example 3,  $\epsilon_{tol} = 10^{-6}$ ,  $t_0 = 0$ ,  $t_{max} = 1$ , Normal Flow.

Predictor	$(m, k)$	$\Delta s = 0.15$					$\Delta s = 0.3$				
		$\alpha$	Ns	Nf	Nl	Ni	$\alpha$	Ns	Nf	Nl	Ni
Euler	-	-	-	-	-	-	-	449	2091	1196	26
WLSE	(2, 3)	-	1165	4920	2592	30	-	-	-	-	-
	(2, 4)	*	-	-	-	-	0.40	291	1337	756	24
	(2, 5)	1.00	554	2378	1270	12	0.30	189	942	564	28
	(3, 5)	0.20	472	2042	1098	20	1.00	289	1363	785	26
WLST	(2, 3)	-	-	-	-	-	-	576	2761	1611	26
	(2, 4)	0.50	598	2521	1326	14	0.65	326	1585	934	29
	(2, 5)	0.80	608	2591	1375	13	0.95	297	1460	866	26
	(3, 5)	0.40	625	2739	1489	21	0.90	305	1458	848	30
WLSIT	(2, 3)	-	-	-	-	-	-	-	-	-	-
	(2, 4)	0.05	592	2507	1324	10	0.35	294	1402	815	24
	(2, 5)	1.00	601	2564	1362	9	0.80	302	1447	843	22
	(3, 5)	*	-	-	-	-	*	-	-	-	-
AB2	-	-	758	3156	1643	11	-	337	1546	875	12
AB3	-	-	651	2788	1489	18	-	-	-	-	-



**Figure 5.5.** Solution path of Eq. (5.5.3).



**Figure 5.6. (a)** Shallow arch structure and loading, **(b)** equilibrium path.

#### 5.5.4 Numerical Example 4

Here, we investigate the performance of the proposed predictors in a practical application pertaining to elastostatics. More specifically, we are concerned with the geometrically nonlinear elastic response of the double pinned shallow arch shown in Fig. 5.6a. The structure is discretized into 42 hybrid NLP finite elements and solved for the external load applied with a geometric imperfection, as shown in Fig. 5.6b. The geometry and material properties considered here are the same as in [176]. The parameterized equilibrium system has the following form:

$$\vec{R}(\vec{u}, t) = \vec{F}_{int}(\vec{u}) - t\vec{P} \quad (5.-2)$$

where  $\vec{F}_{int}$  is the vector of internal forces at the nodes,  $P = 1300$  is the prescribed external load and  $\vec{u}$  is the nodal displacement vector. There are a total of 43 nodes, with 41 nodes in the interior. The total number of primary degrees of freedom (problem dimension) is  $n = 41 \cdot 3 + 2 = 125$ . Thus,  $R : \mathbb{R}^{126} \rightarrow \mathbb{R}^{125}$ . The equilibrium path, shown in Fig. 5.6b, is tracked from the undeformed configuration ( $0 \in \mathbb{R}^{126}$ ) up to the first full application of external load  $P = 1300$ , at which point,  $t = 1$ . The results are listed in Table 5.5, where  $a = 0.2$  for all WLS prediction variants.

**Table 5.5** Results for Example 4,  $\epsilon_{tol} = 10^{-6}$ ,  $t_0 = 0$ ,  $t_{max} = 1$ ,  $\alpha = 0.2$ .

Predictor	$(m, k)$	$\Delta s = 100$				$\Delta s = 280$			
		Ns	Nf	Nl	Ni	Ns	Nf	Nl	Ni
Euler	-	99	311	523	4	69	258	448	5
WLSE	(2, 3)	99	314	432	4	35	161	256	21
	(2, 4)	99	317	439	4	35	150	235	9
	(2, 5)	99	322	450	4	35	150	235	9
	(3, 5)	99	316	438	4	-	-	-	-
WLST	(2, 3)	99	312	418	4	35	151	234	13
	(2, 4)	99	314	433	4	35	154	241	11
	(2, 5)	99	319	444	4	35	156	246	11
	(3, 5)	99	317	440	4	35	159	252	15
WLSIT	(2, 3)	99	319	442	4	35	170	273	25
	(2, 4)	99	320	445	4	35	157	248	13
	(2, 5)	99	324	454	4	35	154	243	9
	(3, 5)	-	-	-	-	-	-	-	-
AB2	-	99	309	520	4	46	175	305	11
AB3	-	99	306	514	4	49	188	327	12

## 5.6 Summary

a fast and reliable multistep prediction scheme was developed to facilitate the tracking of implicitly defined curves using numerical continuation coupled with predictor-corrector methods. It is based on a weighted least squares process, whereby the solution point at the current step, along with a prescribed number of preceding points are used to fit a local polynomial. Due its non-interpolating feature, it can represent the local geometry of the solution curve more faithfully. In addition, it exhibits linear time complexity with regards to the problem dimension, it does not require any derivative information and is versatile in that it allows for different prediction options within its framework. Moreover, it can provide derivative information as an approximation to that of the actual curve at effectively no cost. It is particularly suitable as a prediction toolbox for numerical continuation since it provides reliable estimates for the next solution point which are competitive with other approaches that use derivative information of the Jacobian, even for moderately large step-lengths, as demonstrated by the examples presented. A stabilization procedure was also provided for problems parameterized by the arc-length of the solution curve.

As can be seen from the numerical examples, the WLS variants offer comparable and can even outperform the Euler and the AB2 & AB3 higher order predictors, which employ the exact tangent. For larger steps in particular, we see that achieving reliable estimates of the next solution point can greatly enhance the performance of the continuation process. The proposed prediction schemes do so without utilizing expensive second order information but by reserving an additional portion of available memory, not large, for storing the a number of previously converged solution points. In practice, WLS prediction is unlikely to use a polynomial order higher than  $m = 4$  and, thus, at each step, we can simply ensure that the last  $k = 5$  solution points are available in memory.

## CHAPTER 6

### CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

#### **6.1 Research Summary and Conclusions**

This research work sought to develop a high-performance, NLP beam element and embed it in a general parametric optimization framework that takes advantage of the element formulation. The proposed element features exact kinematic capabilities coupled with a robust and fast multiaxial stress update algorithm, thereby allowing for shear-flexure interaction without introducing kinematic approximations to displacement and rotational fields. The naturally parameterized NLP framework provides the capability for general and truly global parametric investigations of structural frames. In addition, the numerical continuation algorithm used in the implementation of the path-tracking procedure is equipped with a robust, fast and reliable predictor, also as part of this research. This prediction scheme is based on a local weighted least squares fitting and provides superior computational performance when compared to other conventional prediction schemes. In addition, it is a generally applicable algorithm that offers even more considerable advantages when problems possess dense, non-symmetric Jacobians.

#### **6.2 Limitations**

Limitations of the research material presented herein are primarily a direct result of assumptions set initially during the formulation of the relevant parts. As far as the hybrid element is concerned, the present work is focused on quasi-static planar problems that are conservative. In addition, cross-sections are assumed to remain plane after deformation, even if not perpendicular to the beam centerline. This means that warping effects are not taken into account. Another issue related to cross-section kinematics is that simple, even though not simplistic, shear strain distribution patterns were assumed, such as uniform and quadratic, and energetic consistency

with the exact shear strain energy is established through the use of the shear coefficient.

For the multiaxial constitutive model, one limitation pertains to the linear kinematic hardening rule assumed. This simplification was adopted to ease already cumbersome derivations pertaining to the integration of the rate equations. In addition, the model is valid for cross-section with a uniform material that follows the von Mises yield criterion. Extension to include other material constitutive laws is, however, fairly straightforward.

For the naturally parameterized NLP framework, one limitation is the assumption that no points of nullity higher than one occur during the solution process within a cell. This effectively means that no bifurcating solution curves exist when a global parametric study is carried out. Technically, the numerical algorithm proposed in the end of the relevant Chapter is handling such bifurcating behavior when the solution transitions between cells, but this is specifically adjusted for parametric programs involving inequality constraints where loss of LICQ, singularity of the Hessian in the tangent space, or both occur during the transition. Furthermore, the smoothness assumptions imposed on the TPE and the constraints, while realistic for most mechanics applications, restrict the application of the parametric optimization framework to problems that do not involve severe discontinuities and, therefore, can be modelled using smoothness assumptions.

Lastly, regarding the weighted least squares predictor, a basic limitation is the fact that a least squares process does not result in a consistent prediction scheme. This means that as  $\Delta s \rightarrow 0$ , the prediction yielded by any WLS variant does not converge asymptotically to the tangent vector at the current point. In practice however, this is not an issue since the proposed prediction scheme is meant to provide stability and improve convergence performance for problems where a relatively large step-length is used.

### 6.3 Future Directions

This investigation has been a first attempt to establish a parametric nonlinear programming framework for the general parametric study of planar frames using a high performance, hybrid,

NLP-based element. There are, therefore, many parts that should be examined further and features to be added in future research efforts. Below we list some recommendations:

- For the hybrid beam element, the use of different nonlinear programming solution schemes can further be investigated since the optimization literature, which we adhere to emphasize throughout this study, possesses a quite rich collection of solution techniques. Some other directions include the formulation to support initially curved elements, dynamic effects, and three dimensional settings that necessitate special treatment of the non-commutative rotational field.
- With regards to the section state determination phase, the element capability can be extended to address composite section behavior, such that of a reinforced concrete section. In addition, the constrained stress space return mapping algorithm shown for the von Mises criterion can be extended for other materials. Lastly modifying the relevant rate expressions to account for the effect of nonlinear kinematic hardening is also an additional feature that can be incorporated.
- While parameterization with respect to a natural parameter is crucial for the PNLP framework proposed in this work, examining artificial embeddings, especially in conjunction with appropriate NLP solution methods, might also yield valuable results, especially if intermediate operating states of the structural system are of no particular interest. Another important extension to the present NPNNP framework is bifurcation tracking and branch switching capabilities, which have not been investigated in this work.
- Lastly, more research is required in order to better understand the impact of the polynomial order, number of previously converged points, the weighting approach and how these aspects of the WLS predictor interact with each other and with the underlying problem. Furthermore, it might prove valuable to explore the additional options that the WLS framework provides with respect to other aspects of predictor-corrector algorithms and, in

particular, the step-length adjustment phase. Another direction could be the investigation on the influence of arc-length measurements on the quality of the local approximation.

# **Appendices**

## APPENDIX A

### NOTATION

#### Array Notation for Tensors

We outline the array notation used on relevant tensor quantities for the computer implementation of the work presented in the chapters of this manuscript. The relevant cases are i) the three dimensional state and ii) the plane stress state. For each one, we show how second order stress and strain tensors, on one hand, and fourth order material constitutive tensor on the other, are stored in computer memory as one dimensional and two dimensional arrays (“vectors”).

#### Three Dimensions

For the symmetric second order stress tensor  $\sigma$  we have:

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \rightarrow \vec{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{13} \end{bmatrix}^T$$

Similarly, for the symmetric second order strain tensor we have:

$$\epsilon = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{12} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{13} & \epsilon_{23} & \epsilon_{33} \end{bmatrix} \rightarrow \vec{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & \epsilon_{33} & 2\epsilon_{12} & 2\epsilon_{23} & 2\epsilon_{13} \end{bmatrix}^T$$

and it is assumed that for the engineering strains we have  $\gamma_{ij} = 2\epsilon_{ij}$ ,  $i \neq j$ .

The stress and strain convention is extended to all other second order symmetric tensors involved in the formulation of the stress update algorithm, such as back-stress or effective stress tensors or the plastic strain tensor.

The fourth order elastic constitutive tensor,  $\mathbb{C}$ , is represented as a two dimensional matrix  $\mathbf{C}$  as follows:

$$\mathbf{C} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1112} & C_{1123} & C_{1113} \\ C_{2211} & C_{2222} & C_{2233} & C_{2212} & C_{2223} & C_{2213} \\ C_{3311} & C_{3322} & C_{3333} & C_{3312} & C_{3323} & C_{3313} \\ C_{1211} & C_{1222} & C_{1233} & C_{1212} & C_{1223} & C_{1213} \\ C_{2311} & C_{2322} & C_{2333} & C_{2312} & C_{2323} & C_{2313} \\ C_{1311} & C_{1322} & C_{1333} & C_{1312} & C_{1323} & C_{1313} \end{bmatrix}$$

so that:

$$\text{ARRAY}(d\sigma) = \text{ARRAY}(\mathbb{C} : d\epsilon) = \mathbf{C}d\vec{\epsilon} = d\vec{\sigma}$$

For problems in elasticity and associative plasticity, the material constitutive tensor possesses minor and major symmetries:

$$\mathbb{C}_{ijkl} = \mathbb{C}_{jikl} = \mathbb{C}_{jilk} = \mathbb{C}_{klij}$$

For isotropic linear elasticity,  $\mathbf{C}$  is given as follows:

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

Enforcement of the zero-stress condition  $\sigma_{22} = \sigma_{33}$  at a fiber, after the plastic parameter is determined, requires a local iterative procedure whereby  $\mathbf{C}$  is decomposed in accordance to the active and zero stress components. The local iteration formula for the three dimensional algorithm takes the following form:

$$\begin{bmatrix} \Delta\epsilon_{22} \\ \Delta\epsilon_{33} \end{bmatrix} = - \begin{bmatrix} C_{2222} & C_{2233} \\ C_{3322} & C_{3333} \end{bmatrix}^{-1} \begin{bmatrix} \sigma_{22} \\ \sigma_{33} \end{bmatrix}$$

where for each corrected increment in  $\Delta\epsilon_{22}, \Delta\epsilon_{33}$ , the stress update has to be repeated until  $\max(|\sigma_{22}|, |\sigma_{33}|) \leq \epsilon_{tol}$ , with  $\epsilon_{tol}$  being a user-specified tolerance. When convergence has been achieved, only then state variables for a fiber are updated.

Note that the same convention is used for the continuous or the consistent tangent moduli.

### Plane Stress

Here, the vectorization of stress and strain tensors is implemented as follows:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \rightarrow \vec{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{12} & \epsilon_{22} \end{bmatrix} \rightarrow \vec{\boldsymbol{\epsilon}} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{bmatrix}$$

The matrix representation of  $\mathbb{C}$  associated with the plane-stress condition is:

$$\mathbf{C} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1112} \\ C_{2211} & C_{2222} & C_{2212} \\ C_{1211} & C_{1222} & C_{1212} \end{bmatrix}$$

For isotropic linear elasticity in plane stress conditions, we have for  $\mathbf{C}$ :

$$\mathbf{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (\text{A.0})$$

The local Newton iteration to enforce condition  $\sigma_{22} = 0$  during the fiber stress update, typically required when the plane-stress return mapping algorithm by Simo[10] is used, takes

the following form:

$$\Delta\epsilon_{22} = -\frac{1}{C_{2222}}\sigma_{22}$$

## Analysis Operators

We adopt the notation  $\vec{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  to indicate a mapping  $\vec{F}$  that takes as argument a vector of dimension  $n$ ,  $\vec{x} \in \mathbb{R}^n$ , and maps it onto a vector of dimension  $m$ ,  $\vec{y} \in \mathbb{R}^m$ :

$$\vec{y} = \vec{F}(\vec{x})$$

We note that operations involving the symbol  $\nabla$  map scalars to vectors and vectors to matrices. For example, if  $f : \mathbb{R}^p \rightarrow \mathbb{R}$  and  $\vec{v} : \mathbb{R}^p \rightarrow \mathbb{R}^q$ , then  $\nabla f : \mathbb{R}^p \rightarrow \mathbb{R}^p$  and  $\nabla \vec{v} : \mathbb{R}^p \rightarrow \mathbb{R}^{p \times q}$ . For a scalar function  $f(\vec{x})$ ,  $\vec{x} \in \mathbb{R}^n$  we have:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_n} \end{bmatrix}^T$$

## Jacobian of Smooth Maps

The Jacobian of  $\vec{F}$ , denoted  $\mathbf{J}$ , is an  $m \times n$  matrix expressed as follows:

$$\mathbf{J} = \begin{bmatrix} \nabla_{\vec{x}} F_1^T \\ \vdots \\ \nabla_{\vec{x}} F_m^T \end{bmatrix} = \begin{bmatrix} \frac{\partial \vec{F}}{\partial x_1} & \dots & \frac{\partial \vec{F}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \dots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \dots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

## APPENDIX B

### GRADIENT OF HYBRID NLP LAGRANGIAN

Explicit expressions of the beam formulation are provided in this Appendix.

The Lagrangian function is used here, as presented in Eq. (2.30):

$$\mathcal{L}(\vec{\mathbf{y}}, \vec{\mathbf{d}}_g, \vec{\lambda}) = U - W + \vec{\lambda}^T \vec{\mathbf{h}}_g$$

where it is also reminded that  $\vec{\mathbf{y}} = [\vec{\mathbf{y}}_1^T \quad \vec{\mathbf{y}}_2^T \quad \dots \quad \vec{\mathbf{y}}_m^T]^T$ ,  $\vec{\mathbf{y}}_i = [\vec{\mathbf{q}}_i \quad \phi_i]^T$  and  $\vec{\mathbf{q}}_i = [\epsilon_i \quad \gamma_i \quad \kappa_i]^T$ . The number  $m$  designates the *total* number of integration points in the structure.

The gradient of the strain energy,  $U$ , is expressed as:

$$\nabla U = \begin{bmatrix} \nabla_{\mathbf{y}} U \\ \nabla_{\mathbf{d}_g} U \\ \nabla_{\lambda} U \end{bmatrix}$$

The gradient with respect to the strain vector  $\vec{\mathbf{y}}$  gives the stress resultants at all cross-sections and has the following form for a particular cross-section  $i$ :

$$\nabla_{\mathbf{y}_i} U = \begin{bmatrix} w_i N_i \\ w_i V_i \\ w_i M_i \\ 0 \end{bmatrix} = w_i \begin{bmatrix} \vec{\mathbf{F}}_{sec}^{(i)} \\ 0 \end{bmatrix}$$

it is usually computed by numerical integration of Eqs. (2.55) over the cross-section height. In the case of elastic analysis, the above vector can be computed explicitly for each cross-section, since then  $\vec{\mathbf{F}}_{sec}^{(i)} = [EA\epsilon_i \quad k_s GA\gamma_i \quad EI\kappa_i]^T$ .

The gradients of  $U$  with respect to both  $\vec{d}_g$  and  $\vec{\lambda}$  provide zero vectors:

$$\nabla_{\mathbf{d}_g} U = \vec{0} \in \mathbb{R}^{3N_{nod}} \quad \text{and} \quad \nabla_{\lambda} U = \vec{0} \in \mathbb{R}^s$$

where  $s = m + 3n_{el}$ ,  $n_{el}$  being the number of elements. Finally, we then get:

$$\nabla U = \begin{bmatrix} \nabla_{\mathbf{y}} U \\ \vec{0} \\ \vec{0} \end{bmatrix}$$

We restate the constraints equations from Eqs. (2.32a)-(2.32b):

$$\vec{h}_g = \begin{bmatrix} \vec{h}_g^A \\ \vec{h}_g^B \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \vec{d}_g - \boldsymbol{\Lambda}^T \left[ \sum_{i=1}^n w_i \mathbf{R}_i (\vec{q}_i + \vec{E}_1) - \ell \vec{E}_1 \right] \\ \vec{\phi} - \mathbf{V}_2 \vec{d}_g - \mathbf{T} \vec{\kappa} \end{bmatrix}$$

For notational simplicity, we assume here only one element in the derivations that follow. When more elements are used, their corresponding constraint vectors are stacked in vector form as  $\vec{h} = \begin{bmatrix} \vec{h}_g^1 {}^T & \vec{h}_g^2 {}^T & \dots & \vec{h}_g^{n_{el}} {}^T \end{bmatrix}^T$ . The gradient with respect to the strain vector  $\vec{y}_i$  at a particular cross-section is given by:

$$\nabla_{\mathbf{y}_i} \vec{h}_g^A = \begin{bmatrix} \nabla_{\mathbf{q}_i} \vec{h}_g^A & \frac{d\vec{h}_g^A}{d\phi_i} \end{bmatrix} = -w_i \boldsymbol{\Lambda}^T \begin{bmatrix} \mathbf{R}_i & \mathbf{R}_i \mathbf{X} [\vec{q}_i + \vec{E}_1] \end{bmatrix}$$

where:

$$\frac{d\mathbf{R}_i}{d\phi_i} = \mathbf{R}_i \mathbf{X} \quad \text{and} \quad \mathbf{X} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The gradient of  $\vec{h}_g^B$  with respect to strain vector  $\vec{y}_i$ , is:

$$\nabla_{\mathbf{y}_i} \vec{h}_g^B = \begin{bmatrix} \nabla_{\mathbf{q}_i} \vec{h}_g^B & \frac{d\vec{h}_g^B}{d\phi_i} \end{bmatrix} = \begin{bmatrix} \vec{0} & \vec{0} & -\mathbf{T}\hat{\mathbf{E}}_i & \hat{\mathbf{E}}_i \end{bmatrix}$$

where  $\hat{\mathbf{E}}_i \in \mathbb{R}^{n_q}$  is a unit vector with the  $i$ -th component equal to one. Collectively, we have:

$$\nabla_{\mathbf{y}_i} \vec{h}_g = \begin{bmatrix} -w_i \Lambda^T \mathbf{R}_i & -w_i \Lambda^T \mathbf{R}_i \mathbf{X} [\vec{q}_i + \vec{E}_1] \\ [\vec{0} & \vec{0} & -\mathbf{T}\hat{\mathbf{E}}_i] & \hat{\mathbf{E}}_i \end{bmatrix}$$

The gradient with respect to the displacement vector is derived in a straightforward manner by simple derivations:

$$\nabla_{\mathbf{d}_g} \vec{h}_g = \begin{bmatrix} \mathbf{V}_1 \\ -\mathbf{V}_2 \end{bmatrix}$$

$$\mathbf{V}_1 = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \end{bmatrix}, \quad \mathbf{V}_2 = \begin{bmatrix} \vec{0} & \vec{0} & \vec{1} & \vec{0} & \vec{0} & \vec{0} \end{bmatrix}$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \vec{0}, \vec{1} \in \mathbb{R}^{n_q}, \quad \vec{0} = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^T, \quad \vec{1} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$$

Finally, the gradient with respect to the vector  $\vec{\lambda}$  gives a zero matrix:

$$\nabla_{\lambda} \vec{h}_g = \mathbf{0} \in \mathbb{R}^{(n_q+3) \times (n_q+3)}$$

The total constraints gradient is then expressed in block matrix form as:

$$\nabla \vec{h}_g = \begin{bmatrix} \nabla_{\mathbf{y}_1} \vec{h}_g & \nabla_{\mathbf{y}_2} \vec{h}_g & \dots & \nabla_{\mathbf{y}_{n_q}} \vec{h}_g & \nabla_{\mathbf{d}_g} \vec{h}_g & \nabla_{\lambda} \vec{h}_g \end{bmatrix}$$

Note that in the case of small displacements, only the expression  $\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g^A$  has to be modified, so that the strain-displacement relations reduce to the classical linear Timoshenko theory:

$$\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g^A = -w_i \boldsymbol{\Lambda}^T \begin{bmatrix} 1 & -\phi_i & 0 & -\gamma_i \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

with the vector of constraints  $\vec{\mathbf{h}}_g^A$  being:

$$\vec{\mathbf{h}}_g^A = \mathbf{V}_1 \vec{\mathbf{d}}_g - \boldsymbol{\Lambda}^T \left[ \sum_{i=1}^n w_i [\hat{\mathbf{R}}_i (\vec{\mathbf{q}}_i + \vec{\mathbf{E}}_1) - \vec{\mathbf{g}}_i] - \ell \vec{\mathbf{E}}_1 \right]$$

where:

$$\hat{\mathbf{R}}_i = \begin{bmatrix} 1 & -\phi_i & 0 \\ \phi_i & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \vec{\mathbf{g}}_i = \begin{bmatrix} 0 \\ \epsilon_i \phi_i \\ 0 \end{bmatrix}$$

The vector  $\vec{\mathbf{g}}_i$  is added in order to eliminate the second order term  $\epsilon_i \phi_i$  arising from the imposition of the small displacement assumption  $\sin \phi_i \approx \phi_i$ ,  $\cos \phi_i \approx 1$  in Eq. (2.32a).

The potential energy due to external loads  $\vec{\mathbf{P}}$  is:

$$W = \sum_{j=1}^{N_{nod}} \vec{\mathbf{P}}_j^T \vec{\mathbf{d}}_{g,j} = \vec{\mathbf{d}}_g^T \vec{\mathbf{P}}$$

The gradient of the potential energy is, again, straightforward in its derivation as it only depends on the displacement vector:

$$\nabla W = \begin{bmatrix} \nabla_{\mathbf{y}} W \\ \nabla_{\mathbf{d}_g} W \\ \nabla_{\lambda} W \end{bmatrix}$$

with  $\nabla_{\mathbf{y}} W = \vec{\mathbf{0}} \in \mathbb{R}^m$ ,  $\nabla_{\mathbf{d}_g} W = \vec{\mathbf{P}}$  and  $\nabla_{\lambda} W = \vec{\mathbf{0}} \in \mathbb{R}^{n_g+3}$  for one element. Thus, we have:

$$\nabla W = \begin{bmatrix} \vec{\mathbf{0}} \\ \vec{\mathbf{P}} \\ \vec{\mathbf{0}} \end{bmatrix}$$

Finally, the gradients of the Lagrangian with respect to the three constituent vector arguments,  $\vec{\mathbf{y}}_i$ ,  $\vec{\mathbf{d}}_g$  and  $\vec{\boldsymbol{\lambda}}$  are shown below. These expressions are substituted in Eqs. (2.35):

$$\nabla_{\mathbf{y}_i} \mathcal{L} = w_i \begin{bmatrix} \vec{\mathbf{F}}_{sec}^{(i)} \\ 0 \end{bmatrix} + \begin{bmatrix} -w_i \boldsymbol{\Lambda}^T \mathbf{R}_i & -w_i \boldsymbol{\Lambda}^T \mathbf{R}_i \mathbf{X} [\vec{\mathbf{q}}_i + \vec{\mathbf{E}}_1] \\ [\vec{\mathbf{0}} \quad \vec{\mathbf{0}} \quad -\mathbf{T} \hat{\mathbf{E}}_i] & \hat{\mathbf{E}}_i \end{bmatrix}^T \vec{\boldsymbol{\lambda}}$$

$$\nabla_{\mathbf{d}_g} \mathcal{L} = - \begin{bmatrix} \vec{\mathbf{0}} \\ \vec{\mathbf{P}} \\ \vec{\mathbf{0}} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_1 \\ -\mathbf{V}_2 \end{bmatrix}^T \vec{\boldsymbol{\lambda}}$$

$$\nabla_{\lambda} \mathcal{L} = \vec{\mathbf{h}}_g$$

## APPENDIX C

### HESSIAN OF HYBRID NLP LAGRANGIAN

The Hessian form in Eq. (2.36), is restated here:

$$\mathbf{H} = \begin{bmatrix} \nabla_{\mathbf{y}\mathbf{y}}^2 \mathcal{L} & \mathbf{0} & \nabla_{\mathbf{y}} \vec{\mathbf{h}}_g^T \\ \mathbf{0} & \mathbf{0} & \nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g^T \\ \nabla_{\mathbf{y}} \vec{\mathbf{h}}_g & \nabla_{\mathbf{d}_g} \vec{\mathbf{h}}_g & \mathbf{0} \end{bmatrix}$$

The gradients of  $\vec{\mathbf{h}}_g$  have been derived in the previous section. Thus, for the complete specification of the second order information we need only to determine the second derivative matrix with respect to the strain vector  $\vec{\mathbf{y}}$ . Since different integration points are independent of one another, then it is clear that  $\nabla_{\mathbf{y}_i \mathbf{y}_j}^2 \mathcal{L} = \mathbf{0}$  for  $i \neq j$ . Therefore, the matrix  $\nabla_{\mathbf{y}\mathbf{y}}^2 \mathcal{L}$  has a block diagonal form:

$$\nabla_{\mathbf{y}\mathbf{y}}^2 \mathcal{L} = \begin{bmatrix} \nabla_{\mathbf{y}_i \mathbf{y}_i}^2 \mathcal{L} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \nabla_{\mathbf{y}_i \mathbf{y}_i}^2 \mathcal{L} & \cdots & \mathbf{0} \\ \vdots & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \nabla_{\mathbf{y}_m \mathbf{y}_m}^2 \mathcal{L} \end{bmatrix}$$

and making use of Eq. (2.60) gives:

$$\nabla_{\mathbf{y}_i \mathbf{y}_i}^2 \mathcal{L} = \nabla_{\mathbf{y}_i \mathbf{y}_i}^2 U + \nabla_{\mathbf{y}_i} ([\nabla_{\mathbf{y}_i} \vec{\mathbf{h}}_g]^T \vec{\boldsymbol{\lambda}})$$

As already seen, the second derivative matrix of the strain energy yields the generalized section

stiffness of cross-section  $i$ :

$$\nabla_{\mathbf{y}_i}^2 U = w_i \begin{bmatrix} \mathbf{k}_{sec}^{(i)} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

where  $\mathbf{k}_{sec}^{(i)}$  is given by Eq. (2.58) after numerical integration of the cross-section. We only now need to derive the explicit form for the second term of the right hand side in Eq. (C). To this end, we separate the Lagrange multiplier vector into two parts, which correspond to vectors  $\vec{\mathbf{h}}_h^A$  and  $\vec{\mathbf{h}}_h^B$  respectively:

$$\vec{\boldsymbol{\lambda}} = \begin{bmatrix} \vec{\boldsymbol{\lambda}}^A \\ \vec{\boldsymbol{\lambda}}^B \end{bmatrix}$$

Eq. (B) can be now restated as:

$$\nabla_{\mathbf{y}_i} \mathcal{L} = w_i \begin{bmatrix} \vec{\mathbf{F}}_{sec}^{(i)} \\ 0 \end{bmatrix} - w_i \begin{bmatrix} \mathbf{R}_i^T \boldsymbol{\Lambda} \\ [\vec{\mathbf{q}}_i + \vec{\mathbf{E}}_1]^T \mathbf{X}^T \mathbf{R}_i \boldsymbol{\Lambda} \end{bmatrix} \vec{\boldsymbol{\lambda}}^A + \begin{bmatrix} \vec{\mathbf{0}}^T \\ \vec{\mathbf{0}}^T \\ -\hat{\mathbf{E}}_i^T \mathbf{T}^T \\ \hat{\mathbf{E}}_i^T \end{bmatrix} \vec{\boldsymbol{\lambda}}^B$$

Noticing that the gradient of  $\vec{\mathbf{h}}_g^B$  yields components independent of strains, we conclude that the second derivatives of  $\mathcal{L}$  will include only the part  $\vec{\boldsymbol{\lambda}}^A$ , as:

$$\nabla_{\mathbf{y}_i}^2 \mathcal{L} = \nabla_{\mathbf{y}_i}^2 U + \mathbf{Z}_i$$

where:

$$\mathbf{Z}_i = \begin{bmatrix} \mathbf{0}_{(3 \times 3)} & \vec{\mathbf{t}}_i \\ \vec{\mathbf{t}}_i^T & \vec{\mathbf{t}}_i^T \mathbf{X} [\vec{\mathbf{q}}_i + \vec{\mathbf{E}}_1] \end{bmatrix} \quad \text{and} \quad \vec{\mathbf{t}}_i = -w_i \mathbf{X}^T \mathbf{R}_i^T \boldsymbol{\Lambda} \vec{\boldsymbol{\lambda}}^A$$

and the second derivative matrix of the Lagrangian is now fully determined.

## APPENDIX D

### ITERATIVE CORRECTIONS FOR PLASTIC PARAMETER

First, we express the effective stress vector  $\vec{\zeta}_{n+1}$  purely in terms of  $\lambda$ , using eq. (3.23a) and the fact that  $\vec{n} = \sqrt{\frac{3}{2}} \frac{\mathbf{V}\vec{\zeta}}{\|\vec{\zeta}\|_{\mathbf{V}}}$ :

$$\left[ \mathbf{I} + \frac{\lambda \sqrt{\frac{3}{2}}}{\|\vec{\zeta}_{n+1}\|_{\mathbf{V}}} \left[ \mathbf{C}^{el} \mathbf{V} + H_{kin} \mathbf{I} \right] \right] \vec{\zeta}_{n+1} = \vec{\zeta}_{n+1}^{TR} \quad (\text{D.0})$$

In the first phase of a plastic step we assume that no plastic flow occurs (Elastic Prediction), therefore, for the purposes of the iterative strategy we can consider the following identity:

$$f_{n+1} = 0 \rightarrow \|\vec{\zeta}_{n+1}\|_{\mathbf{V}} = q_n + \lambda \sqrt{\frac{2}{3}} \frac{\partial q_n}{\partial e_n^{pl}} \quad (\text{D.0})$$

In the case of linear isotropic hardening, eq. (D) is exact since  $\frac{\partial q}{\partial e^{pl}} = H_{iso}$ . Otherwise the isotropic modulus used in the context of the fully implicit integration is replaced with the one at the previous step for the purposes of the iterative strategy and only for that. This technique is done so that  $\vec{\zeta}_{n+1}$  is recast only in terms of  $\lambda$ . Substituting (D) into (D) and solving for  $\vec{\zeta}_{n+1}$  we get:

$$\vec{\zeta}_{n+1} = \Omega \vec{\zeta}_{n+1}^{TR} \quad (\text{D.0})$$

Let  $a_1 = E + \frac{3}{2}H_{kin} + \frac{\partial q_n}{\partial e_n^{pl}}$  and  $a_2 = 3G + \frac{3}{2}H_{kin} + \frac{\partial q_n}{\partial e_n^{pl}}$ . Then:

$$\Omega = \left[ \mathbf{I} + \frac{\lambda \sqrt{\frac{3}{2}}}{q_n + \lambda \sqrt{\frac{2}{3}} \frac{\partial q_n}{\partial e_n^{pl}}} \left[ \mathbf{C}^{el} \mathbf{V} + H_{kin} \mathbf{I} \right] \right]^{-1} = \text{diag} \left( \frac{q_n + \lambda \frac{\partial q_n}{\partial e_n^{pl}}}{q_n + \lambda a_1}, \frac{q_n + \lambda \frac{\partial q_n}{\partial e_n^{pl}}}{q_n + \lambda a_2} \right) \quad (\text{D.0})$$

Substitution of (D) into  $f(\vec{\zeta}_{n+1}, q_{n+1}) \equiv f_{n+1} = 0$  yields the scalar equation to be solved

iteratively for  $\lambda$ :

$$f_{n+1} = \sqrt{\frac{3}{2}} \sqrt{(\vec{\zeta}_{n+1}^{TR})^T \Omega V \Omega \vec{\zeta}_{n+1}^{TR}} - q_{n+1} = 0 \quad (\text{D.0})$$

The derivative of  $f_{n+1}$  with respect to  $\lambda$  involves finding the derivative of the diagonal matrix  $\Omega$  and a chain rule with respect to  $q_{n+1}$ . It can be found that the derivative is:

$$\frac{df_{n+1}}{d\lambda} = -\frac{1}{q_n} \left[ \frac{n_{11}\zeta_{11}^{TR}(E + \frac{3}{2}H_{kin})}{(1 + \frac{\lambda a_1}{q_n})^2} + \frac{n_{12}\zeta_{12}^{TR}(3G + \frac{3}{2}H_{kin})}{(1 + \frac{\lambda a_2}{q_n})^2} \right] - (1 + \lambda) \frac{\partial q_{n+1}}{\partial e_{n+1}^{pl}} \quad (\text{D.0})$$

## APPENDIX E

### ANALYTICAL DERIVATION OF CONSISTENT TANGENT MODULUS

For the inversion of the block matrix (3.25a) we perform block elimination considering  $\mathbf{A} = \mathbf{C}^{el} + \lambda\boldsymbol{\Psi} - \lambda^2\boldsymbol{\Psi}_{n+1}\left[H_{kin}\mathbf{V} + \lambda\boldsymbol{\Psi}_{n+1}\right]^{-1}\boldsymbol{\Psi}_{n+1}$  as the Schur complement. Notice that while the matrix  $\lambda\boldsymbol{\Psi}_{n+1}$  is not invertible, the diagonal ones are. Omitting the subscript referring to the step, inversion of system (3.25a) yields:

$$\begin{bmatrix} d\vec{\sigma} \\ d\vec{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}_{11} & \boldsymbol{\Xi}_{12} \\ \boldsymbol{\Xi}_{21} & \boldsymbol{\Xi}_{22} \end{bmatrix} \begin{bmatrix} d\vec{\epsilon} - d\lambda\vec{n} \\ d\lambda\vec{n} \end{bmatrix} \quad (\text{E.0})$$

and define  $\boldsymbol{\Gamma} = \lambda\boldsymbol{\Psi}\left(H_{kin}^{-1}\mathbf{V} + \lambda\boldsymbol{\Psi}\right)^{-1}$ . Then, for the block matrices  $[\boldsymbol{\Xi}_{ij}]$  we have:

- $\boldsymbol{\Xi}_{11} = \mathbf{A}^{-1}$ , the inverse of the Schur complement.
- $\boldsymbol{\Xi}_{12} = \boldsymbol{\Xi}_{11}\boldsymbol{\Gamma}$
- $\boldsymbol{\Xi}_{21} = \boldsymbol{\Xi}_{12}^T$
- $\boldsymbol{\Xi}_{22} = \left(H_{kin}^{-1}\mathbf{V} + \lambda\boldsymbol{\Psi}\right)^{-1} + \boldsymbol{\Gamma}\boldsymbol{\Xi}_{11}\boldsymbol{\Gamma}$

where the fact that matrices  $\boldsymbol{\Psi}$  and  $\boldsymbol{\Gamma}$  are symmetric was used. The vector  $\vec{N}$  in (3.28) is given by:

$$\vec{N} = \frac{[\boldsymbol{\Xi}_{11} + \boldsymbol{\Xi}_{12}]\vec{n}}{\|\vec{n}\|_M} \quad (\text{E.0})$$

with  $\mathbf{M} = \boldsymbol{\Xi}_{11} + \boldsymbol{\Xi}_{22} + \boldsymbol{\Xi}_{12} + \boldsymbol{\Xi}_{21}$ . Finally, parameter  $\delta$  is defined as:

$$\delta = \frac{\frac{\partial q}{\partial e^{pl}}}{\|\vec{n}\|_M^2} \quad (\text{E.0})$$

The calculation can be simplified significantly if second and higher order terms in  $\lambda$  are ignored.

## APPENDIX F

### NUMERICAL PATH-FOLLOWING

Assuming that the smoothness and regularity conditions hold for  $\vec{H}$ , let  $\Gamma_0$  be a path corresponding to a regular initial point  $\vec{x}_0$ . This path can be parameterized as  $\vec{y}(s) = (\vec{x}(s), t(s))$ . Following Klopfenstein[78] who proposed  $s$  to be regarded as the arc-length of path  $\Gamma_0$ , by differentiating the Homotopy equation Eq. (4.5) with respect to  $s \in \mathbb{R}_+$ , we arrive at the following IVP:

$$\mathbf{D}\mathbf{H} \cdot \dot{\mathbf{y}}(s) = 0$$

$$\|\dot{\mathbf{y}}(s)\|_2 = 1$$

$$\mathbf{y}(0) = (\mathbf{x}_0, 0), s \geq 0$$

where  $(\cdot)$  denotes differentiation with respect to  $s$ ,  $\|\cdot\|_2$  is the Euclidean norm and  $\mathbf{D}\mathbf{H} = [\mathbf{D}\mathbf{H}_x \mathbf{D}\mathbf{H}_t]$  is the  $n \times (n + 1)$  Jacobian of  $\vec{H}$ . The integration of IVP (5.1) is typically carried out a by predictor-corrector approach, whereby a prediction around the current point is followed by a correction phase using Newton iterations, until, under an appropriate norm and a prescribed tolerance  $\epsilon_{tol}$ , the condition  $\|\vec{H}\| \leq \epsilon_{tol}$  is satisfied.

Let  $N$  denote the current converged step. The next point on the path is usually approximated using Euler prediction:

$$\vec{y}_{N+1}^p = \vec{y}_N + \Delta s \dot{\vec{y}}_N \quad (\text{F.1})$$

where  $\vec{y}_N$ ,  $\dot{\vec{y}}_N$  are known and  $\Delta s$  is a user-defined step-length. Because, in general,  $\|\vec{H}(\vec{y}_{N+1}^p)\| >> \epsilon$ , a correction phase is followed by applying iteratively:

$$\vec{y}_{N+1}^{j+1} = \vec{y}_{N+1}^j + \mathbf{B}^j \vec{r}^r \quad (\text{F.1})$$

with  $j = 0, 1, \dots, j_{max}$  and  $\vec{y}_{N+1}^0 = \vec{y}_{N+1}^p$ . Operator  $\mathbf{B}$  and vector  $\vec{r}$  depend on the algorithm used. The two most common algorithms used are (i) the Normal Flow method[186, 196, 64] and (ii) the Augmented Jacobian method[197, 186, 66]. In the first method the solution is found by solving the following constrained nonlinear program:

$$\text{Minimize } \|\vec{y} - \vec{y}^p\|_2$$

$$\text{subject to } \vec{H}(\vec{y}) = \vec{0}$$

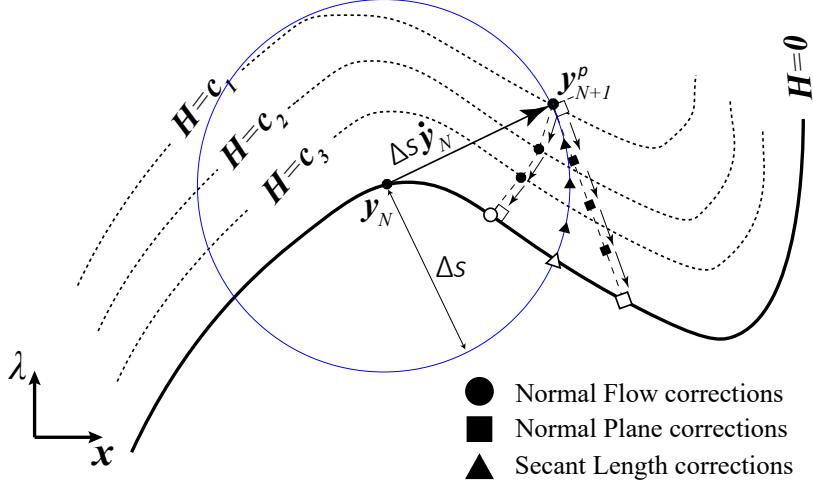
This leads to  $\mathbf{B} = \mathbf{D}\mathbf{H}^\dagger = \mathbf{D}\mathbf{H}^T(\mathbf{D}\mathbf{H} \cdot \mathbf{D}\mathbf{H}^T)^{-1}$ , which is the Moore-Penrose inverse of  $\mathbf{D}\mathbf{H}$ , and  $\vec{r} = -\vec{H}$ . The corrections follow a trajectory that is normal to the perturbed curves  $\vec{H}(\vec{y}) = \vec{c}_{cont}$ .

In the Augmented Jacobian approach, a constraint equation,  $V(\vec{y}) = 0$  is introduced so that, combined with Eq. (F.1), a system of  $n + 1$  equations with  $n + 1$  unknowns is formed and the resulting augmented  $(n + 1) \times (n + 1)$  matrix,  $\mathbf{A}$ , is always non-singular. Some popular choices for  $V$  are listed below:

- $V(\vec{y}) = \dot{\vec{y}}^T(\vec{y} - \vec{y}^p)$ , resulting in the so-called normal plane method.
- $V(\vec{y}) = \|\vec{y} - \vec{y}_N\|_2^2 - \Delta s^2$ , resulting in the secant length method[170].
- $V(\vec{y}) = \vec{e}_i^T(\vec{y} - \vec{y}_N) - 1$ , which results in the switching parameter algorithm proposed in [73], with  $\vec{e}_i$  being the  $i$ -th standard basis vector of  $\mathbb{R}^{n+1}$ .

Intermediate Normal Flow corrections correspond to contours of  $\vec{H}$ ,  $\vec{H}_j$ , where  $\vec{H}_j = \vec{c}_{cont,j}$ ,  $\|\vec{c}_{cont,j}\|_2 > \epsilon_{tol}$ , with  $j$  the iteration counter. Normal Plane corrections lie on a plane orthogonal to the tangent at  $\vec{y}_N$  and at distance  $\Delta s$ , while Secant Length corrections are forced to lie on a sphere of radius  $\Delta s$ , centered at  $\vec{y}_N$ .

In any case, for the Augmented Jacobian approach,  $\mathbf{B} = \mathbf{A}^{-1} = [\mathbf{D}\mathbf{H}^T \quad \vec{q}_c]^{-1}$  and  $\vec{r} = -[\vec{H}^T \quad V]^T$ , where  $\dot{V}(\vec{y}) = \vec{q}_c^T \dot{\vec{y}}$ . The techniques derived from the Augmented Jacobian approach underpin most incremental algorithms that have been developed for tackling stability



**Figure F.1.** Two correction algorithms: Normal Flow and Normal Plane.

problems in computational solid mechanics. [81, 198, 85, 87, 86]. The Normal Flow and two cases of Augmented Jacobian iterates are depicted in figure (F.1), in particular the normal plane and secant length methods. For details on the numerical linear algebra pertaining to these methods, the interested reader can refer to the works of Keller[66, 199], Chan and Keller[200], Griewank[201], Rheinboldt[197], Lin et.al.[102], among others.

## APPENDIX G

### A SIMPLE GUI FOR THE HYBRID NLP ELEMENT

This chapter introduces a Graphical User Interface (GUI) developed as part of this work in order to facilitate analyses using the hybrid NLP element. We detail the various features that are implemented using the AppDesigner utility of MATLAB, version 2022b. The chapter is divided in seven sections, each representing a particular tab of the interface. These seven tabs are : 1) Mesh, 2) Materials, 3) Sections, 4) Boundary Conditions, 5) Analysis Controls, 6) Solve Job and, finally, 7) Post-processing. The first five tabs, some of which can be edited independently, belong to the pre-processing phase. Tab 6 calls the hybrid NLP solver for the given input in tabs 1-5 and tab seven provides post-processing information with respect to the structure response in the form of force-displacement plots and configuration history for all steps. A number of tooltips can be revealed by hovering over relevant fields.

#### **The Mesh Tab**

The first tab of the GUI, depicted in Fig. G.1, is concerned with nodal and element input. We analyze it in terms of parts A to N:

- **A:** The Mesh tab.
- **B:** Facility used to manually add nodes. Numeric values pertaining to the X (horizontal) and Y (vertical) coordinate axes are used in the corresponding fields to define the coordinates of a node. By clicking on the *Add Node* button, the enumerated node is added to the table with its coordinates also listed. The enumeration is automatic. Attempting to add a node with coordinates (X,Y) that are already defined for another node results in an error, as shown in Fig. G.2.

- **C:** Facility used to delete a particular node. The *Label* field takes as input the node label already generated and removes it from the table. The enumeration of remaining nodes is updated accordingly. If node deletion is performed after the mesh is generated, the latter is no longer valid and all elements are removed.
- **D:** Success/Error indicator of an action pertaining to node generation, coupled with a text field for relevant message output (e.g. see G.2).
- **E:** Facility to generate a node set from a **.txt** or **.xlsx** file. The format for this input file is shown in Fig. G.3. If a number of nodes have already been added manually, this action replaces all existing nodes. In contrast, generating a nodal set from an input file and then manually adding additional nodes results in appending the previously generated nodal list.
- **F:** Table that lists all generated nodes along with their coordinates.
- **G:** Facility used to manually add a (hybrid NLP) beam element. Since only one element per member is adequate with the proposed formulation, the only relevant inputs are the start and end node labels, indicated in the relevant fields as *Node i* and *Node j* respectively. Again, adding an element with start and end nodes already defined for an existing element results in an error. In addition, element enumeration is, again, automatic.
- **H:** Element deletion facility. The element label used as input in the relevant field and by clicking the *Delete Element* button results in removal of the particular element from the table. The remaining elements are, again, automatically re-enumerated.
- **I:** Success/Error indicator of an action pertaining to element generation, coupled with a text field for relevant message output.
- **J:** Facility to generate an element partition from a **.txt** or **.xlsx** file. The format for this input file is shown in Fig. G.4. If a number of elements have already been added manually,

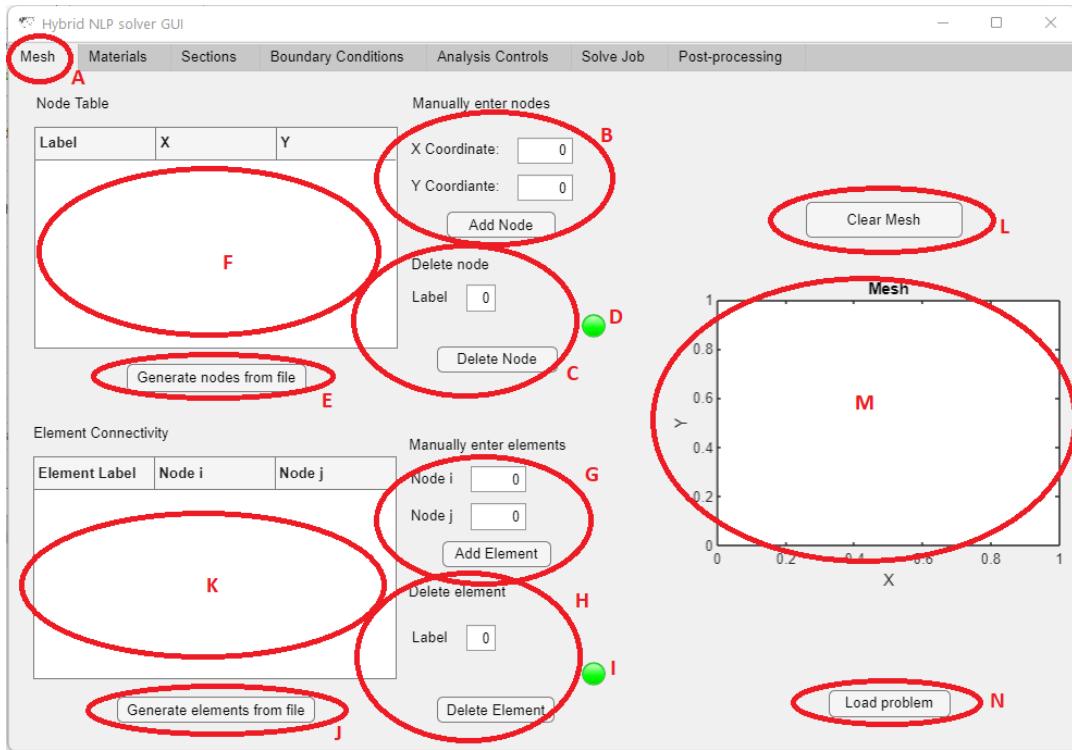
this action replaces all existing elements. In contrast, generating an element partition from an input file and then manually adding additional elements results in appending the previously generated element list.

- **K:** Table that lists all generated elements along with their start and end nodes.
- **L:** Once clicked, the *Clear Mesh* button resets all existing input in the Mesh tab and the user can start over.
- **M:** GUI axes that dynamically update the existing state of nodal and mesh configuration. Figure G.5 shows the Mesh tab when the mesh generation is complete for a simple case of a one bay, one story frame. The nodal labels are places right next to their respective nodes and are typeset in normal font while element labels are approximately placed in the midspan of their corresponding elements and are typeset in italic bold fonts.
- **N:** With the *Load problem* facility, the user can load a problem in its entirety. The appropriate file type is ASCII (extension ".txt") and formatting details for the input file are discussed in Section G of this Appendix. Once the input file is imported, the user can go directly to tab *Solve Job* and proceed with solving the problem.

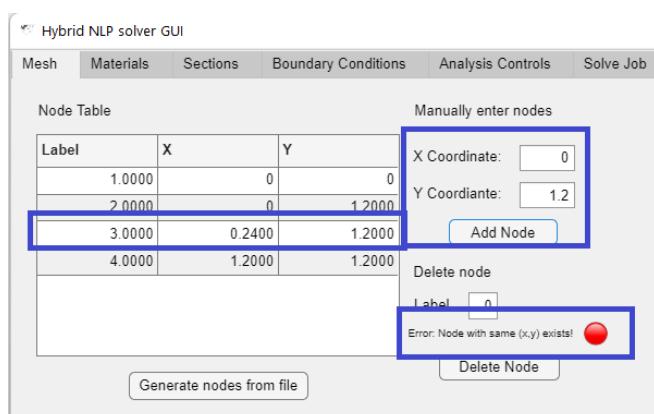
## The Materials Tab

The *Materials* tab follows the *Mesh* tab and does not have any dependencies regarding previous input. Figure FIG outlines all relevant fields in this tab, again, separated by red cycles based on their functionality. We list and describe items A-I below:

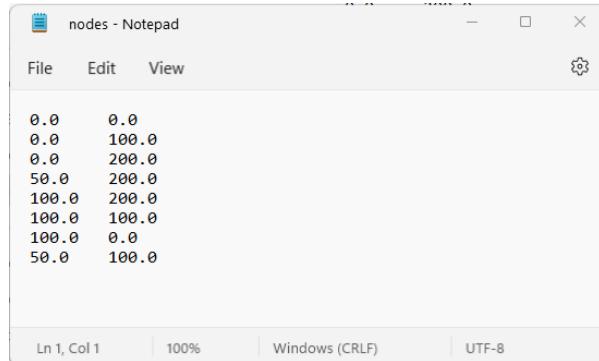
- **A:** The *Materials* tab.
- **B:** Material name field. Takes alphanumeric values. If a material name is not specified, a default name **Material-#** is used, where # takes values 1,2,3... etc. As an example, see Fig. G.7.



**Figure G.1.** The *Mesh* tab with relevant facilities A-M marked in red circles and enumerated with capital english letters.



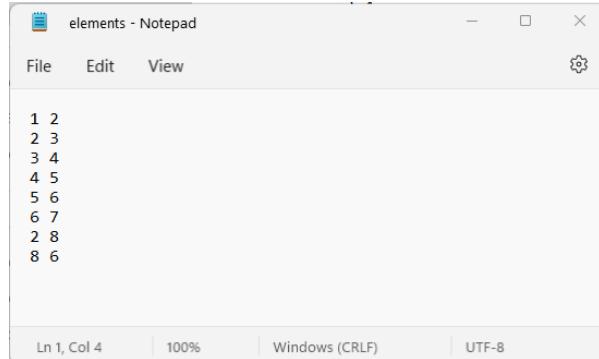
**Figure G.2.** Error when attempting to add a node with coordinates already defined for an existing node.



The screenshot shows a Windows Notepad window titled "nodes - Notepad". The menu bar includes "File", "Edit", and "View". The status bar at the bottom shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8". The main content area contains the following text:

```
0.0      0.0
0.0      100.0
0.0      200.0
50.0     200.0
100.0    200.0
100.0    100.0
100.0    0.0
50.0     100.0
```

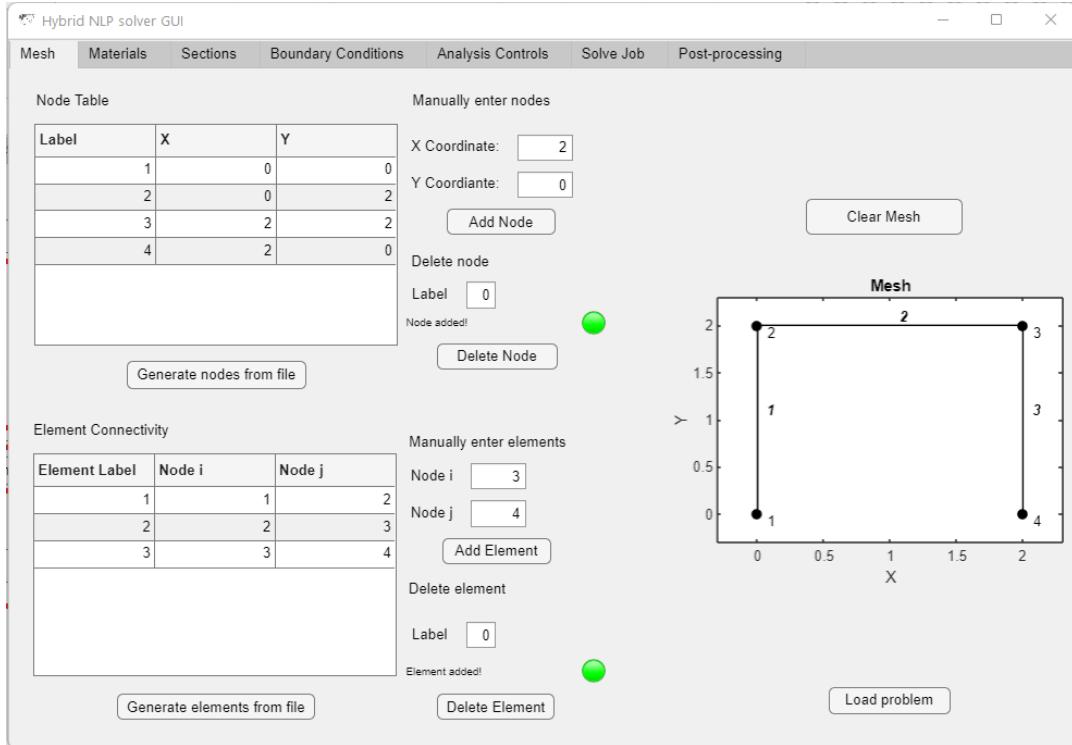
**Figure G.3.** A .txt file with the required format for nodal input. First column is X coordinate, second is Y coordinate. The  $k$ -th row represents Node  $k$ .



The screenshot shows a Windows Notepad window titled "elements - Notepad". The menu bar includes "File", "Edit", and "View". The status bar at the bottom shows "Ln 1, Col 4", "100%", "Windows (CRLF)", and "UTF-8". The main content area contains the following text:

```
1 2
2 3
3 4
4 5
5 6
6 7
2 8
8 6
```

**Figure G.4.** A .txt file with the required format for element input. First column is start node  $i$ , second is end node  $j$ . The  $k$ -th row represents Element  $k$ .



**Figure G.5.** The *Mesh* tab with a completed mesh.

- **C:** Numeric fields pertaining to elastic and plastic properties of the material. In the case of purely elastic analysis, dummy or zero values can be used in the plastic fields, as they are not considered.
- **D:** The *Add* and *Delete* material facilities, along with the Success/Error indicator (green lamp icon). To delete a material defined previously, the user needs to click on it in the *Materials Panel* seen on the left side of the tab (**Item H**, see below), and then click the *Delete* button.
- **E:** Axes that depict the uniaxial stress-strain law for a defined material. In order to activate it, the user will need to click on the specific material in the *Materials Panel* seen on the left side of the tab (**Item H**, see below). An max strain  $\epsilon_{max} = 10\epsilon_y$  is specified purely for plotting purposes. This feature can be seen in Fig. G.8.
- **F:** The *Update Material* facility updates the elastic and plastic properties of an existing material. The material to be updated needs to be selected in the *Materials panel* seen on

the left side of the tab (**Item H**, see below).

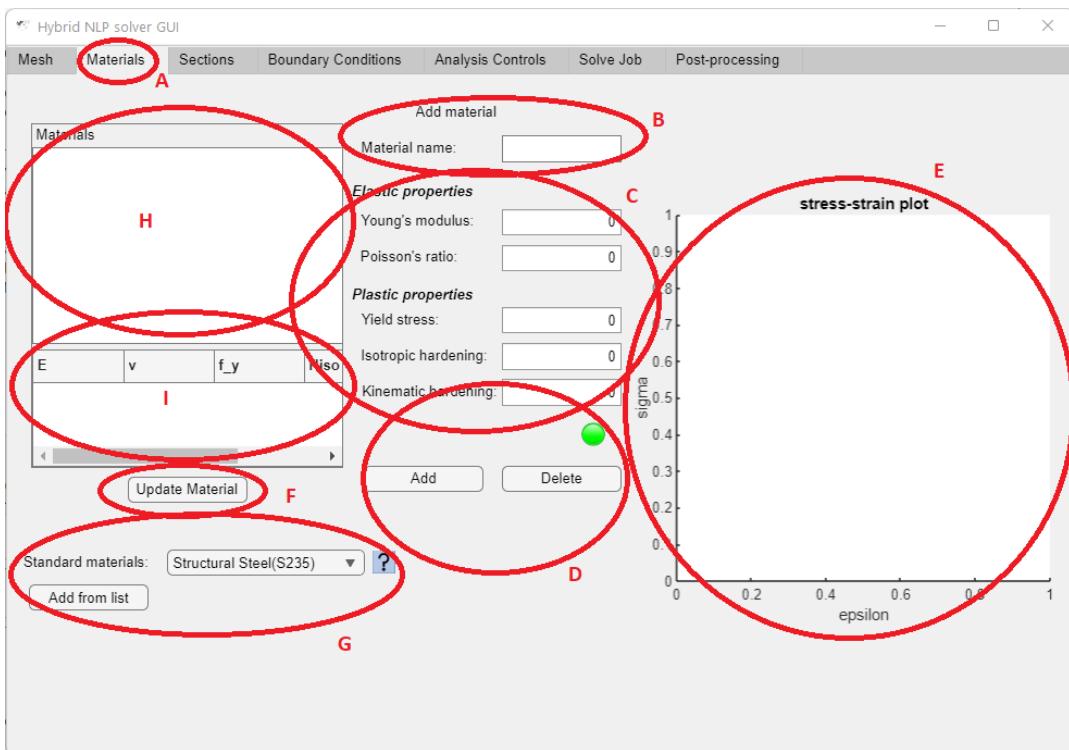
- **G:** A *Standard materials* facility, which is introduced to quickly add predefined, standardized materials from the drop-down list. These materials are assumed to be elastic-perfectly plastic but can be modified using the *Update Material* facility. Two materials are available: 1) Structural Steel(S235) and 2) Aluminum Alloy(6062) (see Fig G.9). In contrast with manually adding materials, duplicate materials from the drop-down are not permitted, as can be seen from Fig. G.10.
- **H:** This is the *Materials* panel. All materials defined and added are listed here. The user needs to click on a specific material on the list in order to i) delete it, ii) update its parameters or iii) view its uniaxial stress-strain plot.
- **I:** The elastic and plastic properties of a material selected on the *Materials* panel are shown in this table. The tabulated quantities correspond to the material fields in **B** as follows: i)  $E$  represents Young's modulus, ii)  $\nu$  represents Poisson's ratio, iii)  $f_y$  represents the yield stress of the material, iv)  $H_{iso}$  represents the isotropic hardening modulus and v)  $H_{kin}$  represents the kinematic hardening modulus.

## The Sections tab

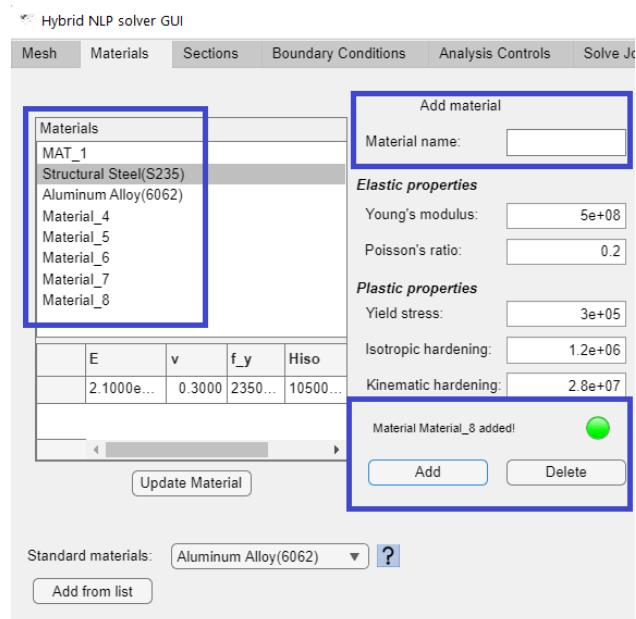
The third tab pertains to cross-section definition. Cross-sections as objects are assigned a name, a shape, numeric values pertaining to the basic dimension of the selected shape (e.g. two basic dimensions for rectangular sections). Moreover, each section is assigned a material. As a consequence the *Sections* tab has a dependency on the *Materials* tab. If a material gets deleted, then all sections that were assigned that material get deleted as well.

We now proceed with the description of the essential parts of the *Sections* tab, shown in Fig. G.11.

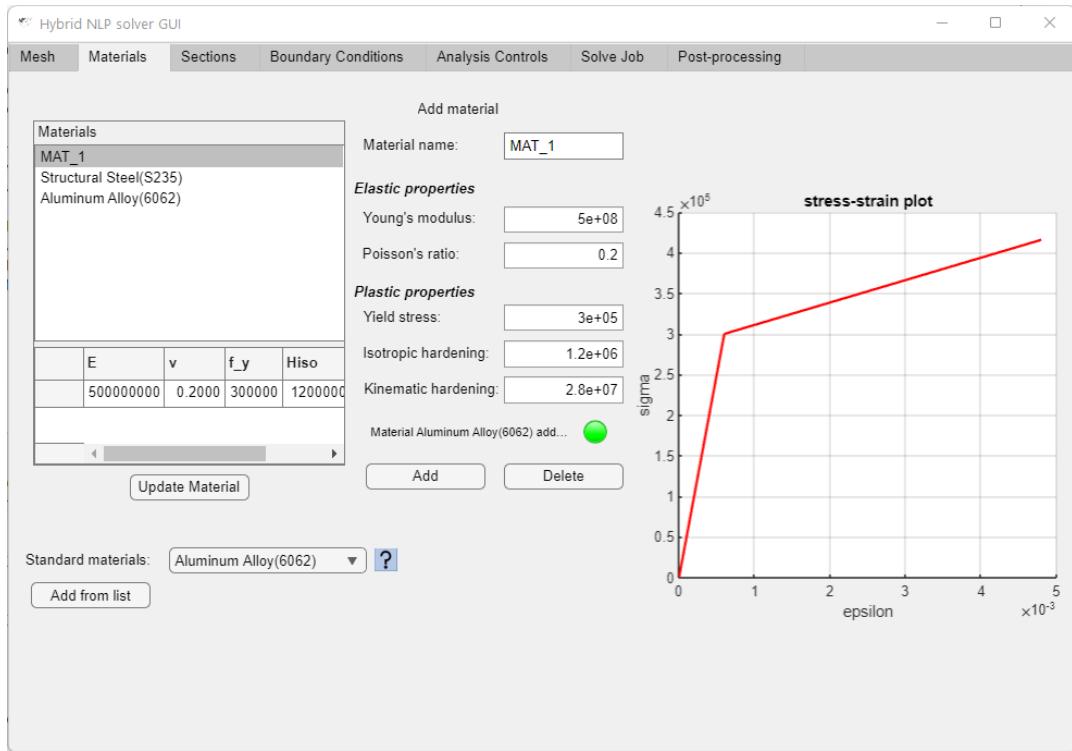
- **A:** The *Sections* tab.



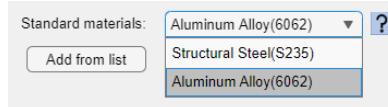
**Figure G.6.** The *Materials* tab with relevant facilities A-M marked in red circles and enumerated with capital english letters.



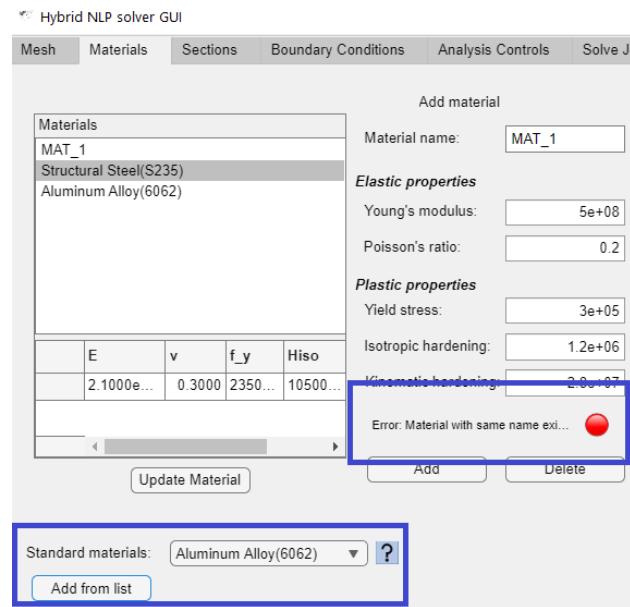
**Figure G.7.** Multiple materials defined without a specified material name.



**Figure G.8.** Potting of uniaxial stress-strain law of material MAT\_1.

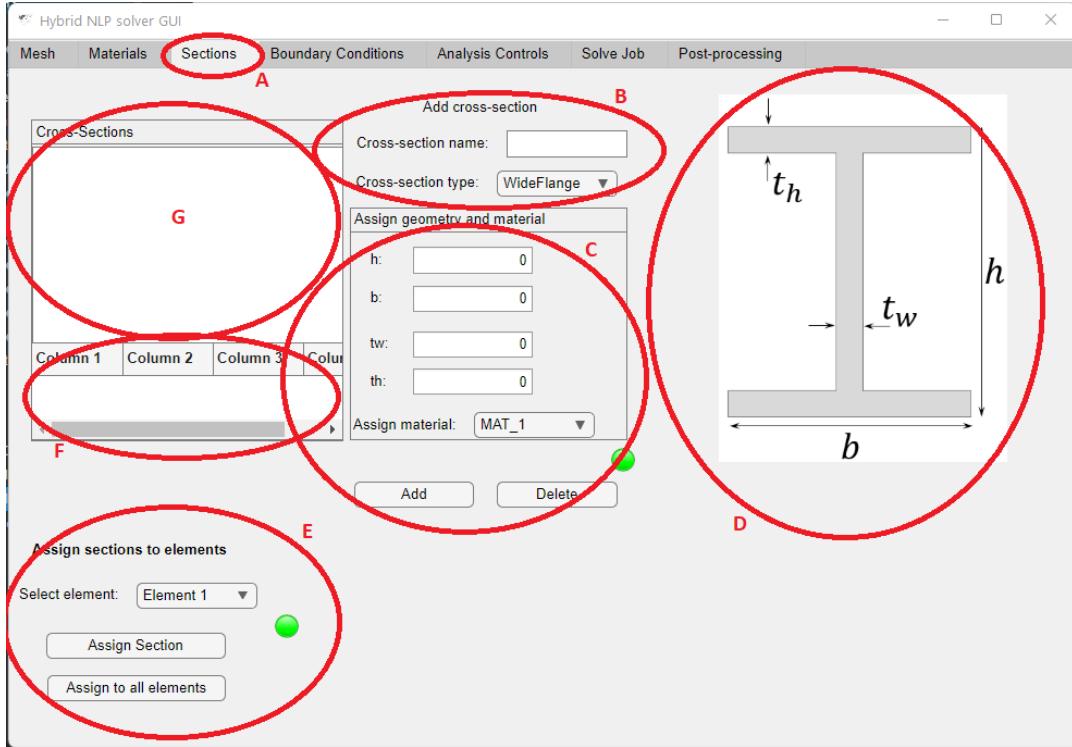


**Figure G.9.** Available predefined materials from the *Standard materials* drop-down list.



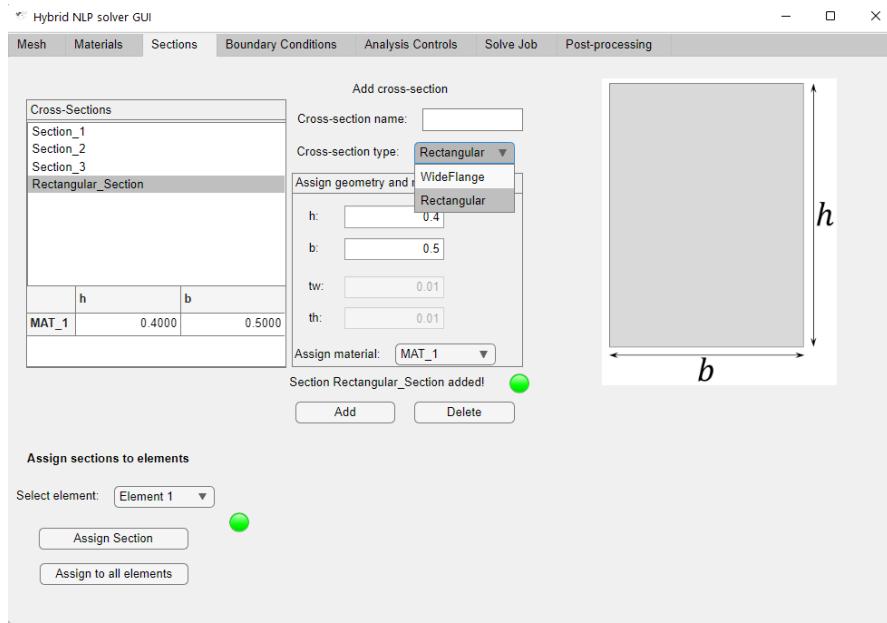
**Figure G.10.** Error when trying to add a standard material that was already added previously.

- **B:** Here, the user assigns a name and a shape to the section. The name field takes alphanumeric values. If not specified, then the name **Section-#** is assigned to the section, where  $\# = 1, 2, 3, \dots$ . The available shapes from the drop-down list are i) rectangular and ii) (symmetric) wide flange geometries.
- **C:** In this section, the user assigns numeric values to the relevant fields that pertain to the basic dimensions of the assigned shape. A rectangular shape has two basic dimensions: height  $h$  and width  $b$ . The wide flange shape has two additional shapes: web thickness  $t_w$  and flange thickness  $t_f$ . In addition, the user also specifies which material will be assigned to the section. As can be seen from Fig. G.12, the section **Rectangular Section** is assigned the material **MAT\_1** defined previously. The drop-down list is also shown, where all other materials defined in the *Materials* tab are available in the list.
- **D:** Axes that depict the cross-section shape selected from the drop-down list. The basic dimensions are also indicated in each case.
- **E:** In this part, the user assigns sections to elements. Tapered element capabilities are not included in the present code, therefore only one section per element is permitted. A section is selected from the *Cross-Sections* panel, which is located at the top left of the tab (see also **Item G** below) and then an element has to be selected that will be assigned the section. Another feature for this part is the capability to assign a specific to all elements, which is done by selecting a section and then clicking on the *Assign to all elements* button. Lastly, a Success/Error indicator is included here as well in order to facilitate the assignment process. For elements that are successfully assigned a section, a message will be shown that states the name of the assigned section, as shown in Fig. G.13 for **Element 1** which was assigned the section **Aluminum Section**. In contrast, for elements that have not been assigned a section, when selected in the drop-down list, the indicator will turn red and an appropriate message will be displayed. This can be seen in Fig. G.14.

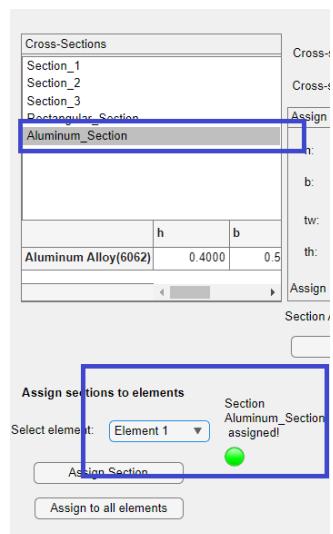


**Figure G.11.** The *Sections* tab with all essential parts marked in red cycles and enumerated.

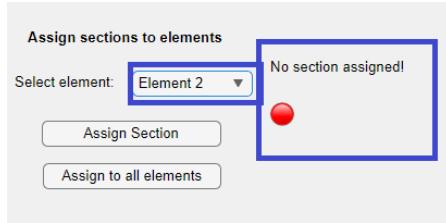
- **F:** Table that lists the basic dimensions for the section currently under selection in the *Cross-Sections* panel (see below).
- **G:** The *Cross-Sections* panel is where all sections created are listed. In order to delete or assign a section to an element, it has to be selected from the panel.



**Figure G.12.** Typical section definition process. A previously defined material **MAT\_1** is assigned to section **Rectangular\_Section**.



**Figure G.13.** Successful section assignment for **Element 1**.

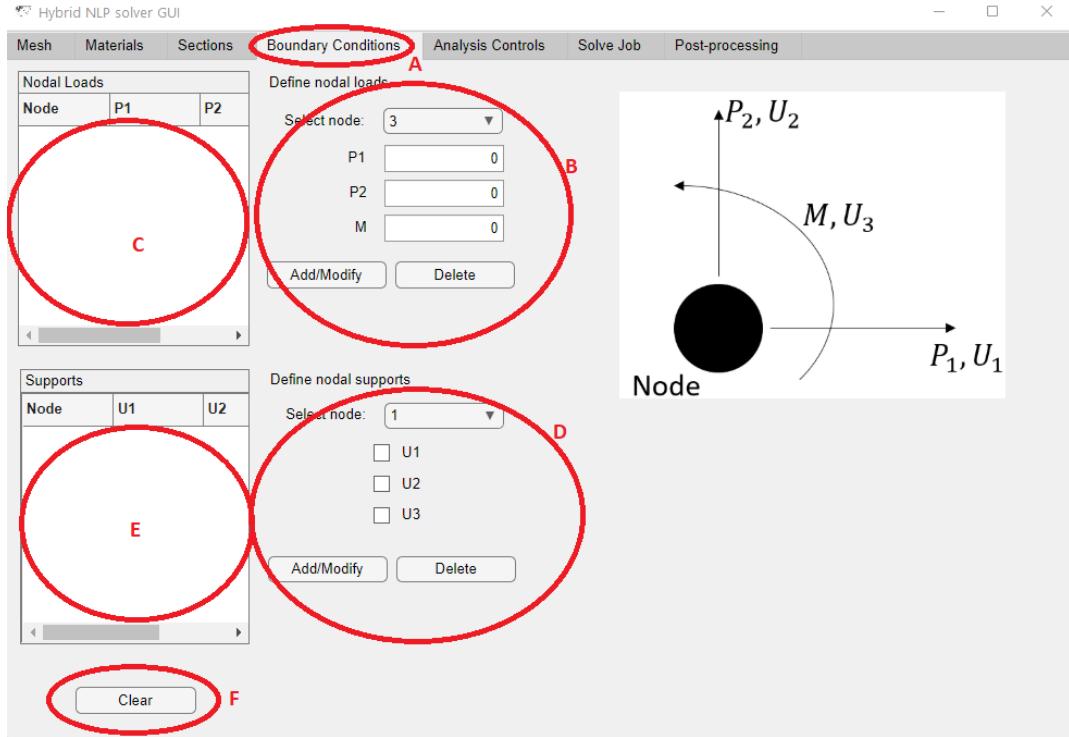


**Figure G.14.** Indication that **Element 2** still needs to be assigned a cross-section.

### The Boundary Conditions tab

In the *Boundary Conditions* tab the user specifies the externally applied nodal loads *and* and the nodal support conditions so that rigid body modes are excluded. The layout of the tab is shown in Fig. FIG and parts from A to F are discussed below:

- **A:** The *Boundary Conditions* tab.
- **B:** Here the user selects a node from the drop-down list at the top to apply an external load at a certain nodal DOF. Once the node is selected, the user can specify the magnitude and sign of the applied force in three available force fields: 1) P1, which is aligned with the global X axis (horizontal direction), 2) P2, which is aligned with the global Y axis (vertical direction) and 3) M, which represents the moment and the direction of the axis which it aligns with is determined by the right-hand rule. In Fig. G.16 an example is shown where horizontal and vertical loads are applied at nodes 2 and 3.
- **C:** This is the *Nodal Loads* table, where all nodes that are assigned a non-zero external load along at least one nodal DOF are listed (see G.16).
- **D:** Here, the user specifies supports conditions for certain nodal DOFs. Again, after selecting a node from the drop-down list, the user specifies which of the three available DOFs, U1, U2 and U3, are fixed. There is a one-to-one correspondence between P1-P2-M and U1-U2-U3 as far as direction of application. However, here, one only needs to activate the available boxes in order to constrain the corresponding displacement DOF.



**Figure G.15.** The *Boundary Conditions* tab with all essential parts marked in red cycles and enumerated.

Figure G.17 shows a case where node 1 is assigned a roller support condition ( $U_1$  and  $U_2$  fixed), whereas node 4 is assigned a clamp support condition ( $U_1$ ,  $U_2$ ,  $U_3$  fixed).

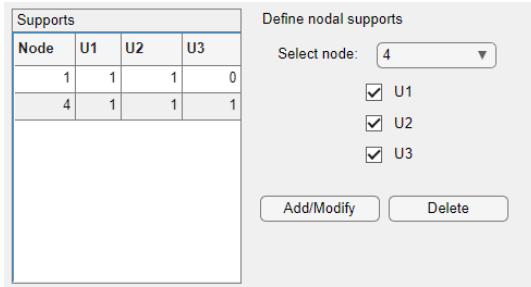
- **E:** Again, the table lists the nodes that have at least one displacement DOF fixed (see Fig. G.17).
- **F:** With the *Clear* button, the user can erase all input in this tab and start over.

Nodal Loads			
Node	P1	P2	M
2	100	-250	0
3	-300	15	0

Select node: 3
Add/Modify
Delete

Define nodal loads		
Node	P1	P2
3	-300	15

**Figure G.16.** Input example for external nodal loading specification.



**Figure G.17.** Input example for nodal support specification.

### The Analysis Controls tab

In the *Analysis Controls* tab, the user can specify settings pertaining to geometric and constitutive assumptions as well as modify the default incremental analysis settings. An option to access advanced features is also given, whereby one can select between different continuation solvers and, if the Homotopy solver is chosen, the user can further specify details such as prediction and correction schemes. The latter gives the option to use the WLS predictor.

The basic layout of the tab is shown in Fig. G.18. The marked components are:

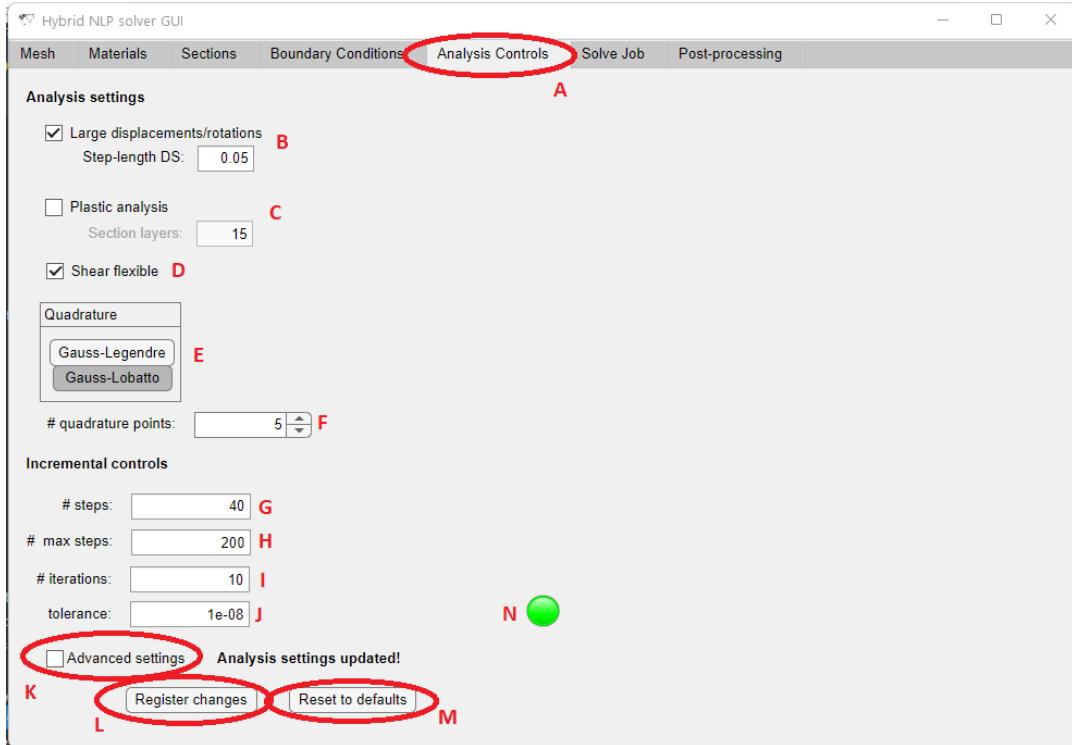
- **A:** The *Analysis Controls* tab.
- **B:** The option to enable geometrically-exact kinematic assumptions. When it is, the equilibrium solver used by default is Crisfield's cylindrical arc-length method. For this reason, the user can also specify a step-length, *DS*.
- **C:** This option pertains to the incorporation of material nonlinearities during the analysis. It is disabled by default. If enabled, the user can also specify the number of cross-section layers. This number is the same for all sections in the structure.
- **D:** Option that pertains to the incorporation shear effects during the analysis. If enabled, which is the default setting, and the plastic analysis option is also enabled, then a quadratic distribution for the shear strains is assumed along the cross-section height.
- **E:** Numerical quadrature options. The user can choose between Gauss-Legendre and

Gauss-Lobatto schemes. The latter tends to be more accurate for frame structures since the end-points of the domain are included in the quadrature set.

- **F:** The user sets the order of numerical quadrature. The minimum allowed is 1 and the maximum is 12.
- **G:** Number of incremental steps to be used if the Load Control equilibrium solver is chosen. When the Cylindrical arc-length is used, this number simply divides the external load vector in order to get a base load.
- **H:** Maximum number of steps. It pertains only to continuation solvers, such as Crisfield's Cylindrical arc-length and the Homotopy solver. The solver will stop 1) when the external load is surpassed or 2) when the maximum number of steps is reached.
- **I:** The maximum number of correction iterations to be performed during an incremental step.
- **J:** The tolerance used to test convergence in term of the Euclidian norm of the out-of-balance (residual) force vector.
- **K:** Option to show the advanced analysis settings (see below).
- **L:** Once the user has specified all analysis controls in tab, they have to register the input by clicking on the *Register changes* button.
- **M:** The user can reset all settings in this tab to their defaults by using the *Reset to defaults* button.
- **N:** Success/Error indicator for this tab.

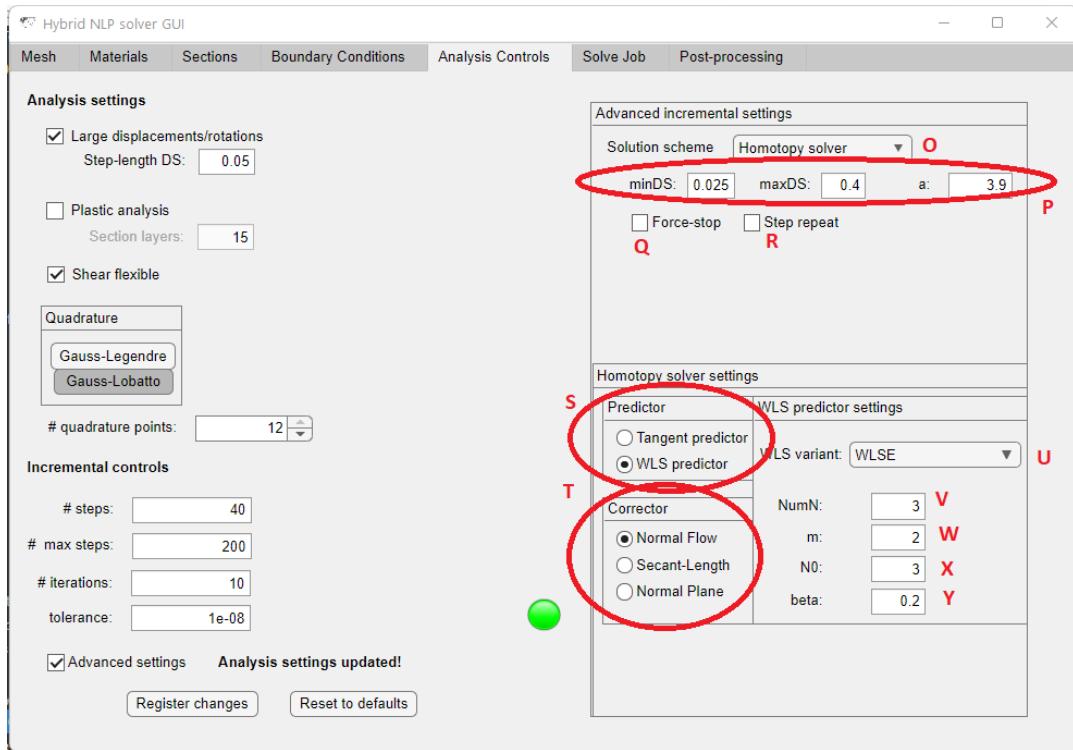
We now discuss the advanced settings panels, shown in Fig. G.19, which become visible only when the *Advanced settings(K)* option is enabled:

- **O:** Drop-down list (see Fig. G.20) where the user can select between three different solution schemes: 1) Load Control, 2) Crisfield's arc-length and 3) Homotopy solver. By default, Crisfield's solver is enabled.
- **P:** Numeric fields pertainig to a) minimum step-length, b) maximum step-length and c) step-length adjustment parameter. The first two limits ensure that the step-length will not become exceedingly small or large. The latter adjust the step-length based on how easily the solution process converged in the last step. It is advised that, in case of plastic analyses, parameter  $a$  be set to values between 4 and 8. These three options are not relevant to the Load Control scheme. The values shown in Fig. G.19 are the default ones.
- **Q:** Option to force Crisfield's arc-length and the Homotopy solver to repeat the last incremental step until the reach load is within a specified tolerance of the externally applied load. If this option is not enabled the solver will stop when the base load mulitplied by the load factor simply exceed the external load vector.
- **R:** Option that, when enabled, forces the Homotopy and Crisfield's arc-length solvers to repeat steps that failed to converge within the specified iteration counter, but with halved step-length. The same step will be repeated until convergence or the  $\min DS$  is reached.
- **S:** The options in the *Homotopy solver settings* become visible only when the Homotopy solver is selected in **O**. The Predictor option allows the user to choose between i) tangent differential predictor and ii) the WLS predictor. If the latter is chosen, an additional panel of options, specific to this precition scheme, appears.
- **T:** Here the user can specify the correction scheme to be used. The three options are: i) Normal Flow[70], ii) Secant-Length[182] and iii) Normal Plane[85].
- **U:** If the *WLS predictor* is selected in **S**, the user can specify which variant of the WLS predictor to use. The available options are: i) WLSE, which is the default, ii) WLST and iii) WLSIT.

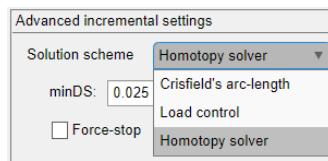


**Figure G.18.** The *Analysis Controls* tab with all essential parts marked in red cycles and enumerated.

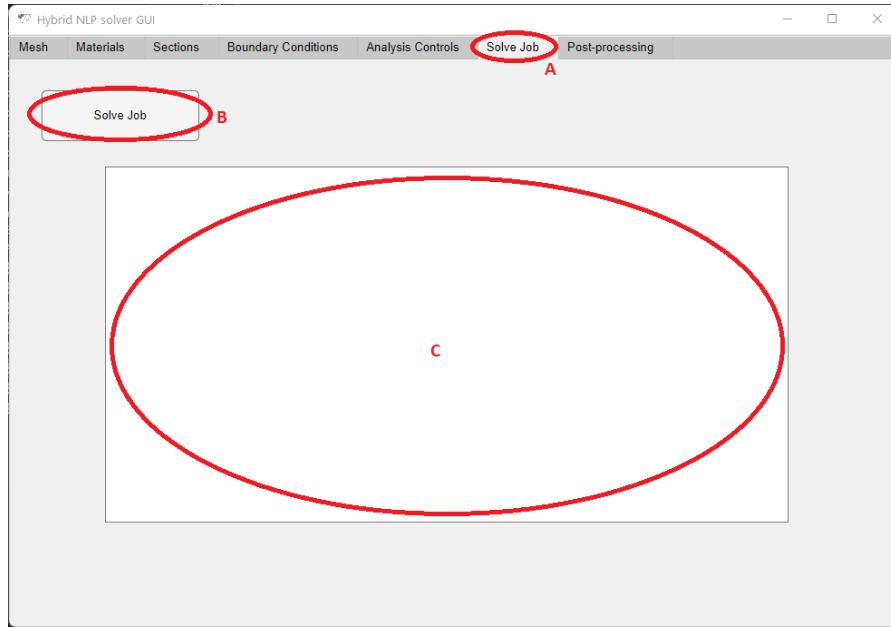
- **V:** This parameter determines the number of load-controlled steps to be performed before the homotopy solver using WLS predictor takes over. A number of simple Newton steps need to be performed in order to generate enough solutions available for the WLS scheme.
- **m:** The degree of polynomial basis used in the WLS fitting. In this version, this remains constant throughout the analysis. Future versions will provide an option to automatically adjust it during the solution process.
- **N0:** The number of previously converged points to consider for the WLS fitting. In this version, this remains constant throughout the analysis. Future versions will provide an option to automatically adjust it during the solution process.
- **beta:** The weight function parameter, with default value 0.2. If set to 1, then there is no weighting during the least squares fitting. Keep this parameter larger than zero.



**Figure G.19.** The *Analysis Controls* tab with the advanced settings enabled.



**Figure G.20.** The *Analysis Controls* tab with the advanced settings enabled.



**Figure G.21.** The *Solve Job* tab.

### The Solve Job tab

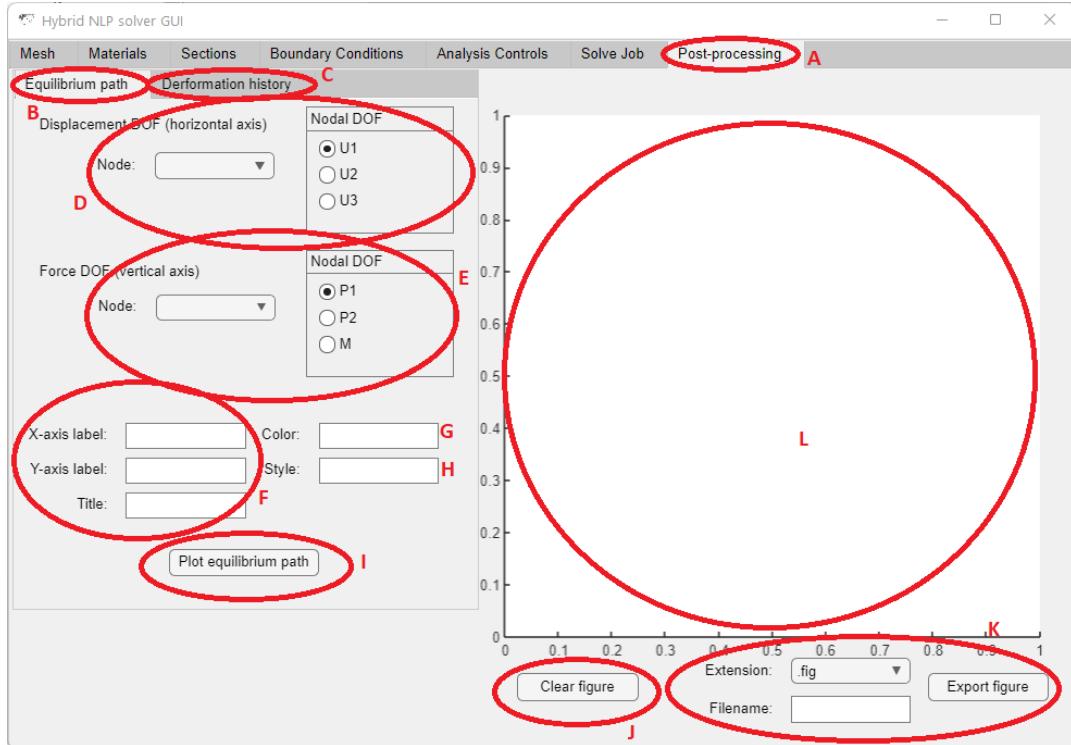
This is the tab where input, if successfully registered, is submitted and the structural problem is solved using the hybrid NLP element. The layout, shown in Fig. FIG, is fairly simple and relevant parts are:

- **A:** The *Solve Job* tab.
- **B:** If all imput is successfully registered, then clicking on the *Solve Job* button will call the hybrid NLP solver. Otherwise, an error message will be shown and the user will be directed to the relevant tab to make corrections.
- **C:** The text box where all analysis settings specified by the user are displayed once the job is submitted. When the analysis is over, the number of steps and total iterations is also displayed.

## The Post-processing tab

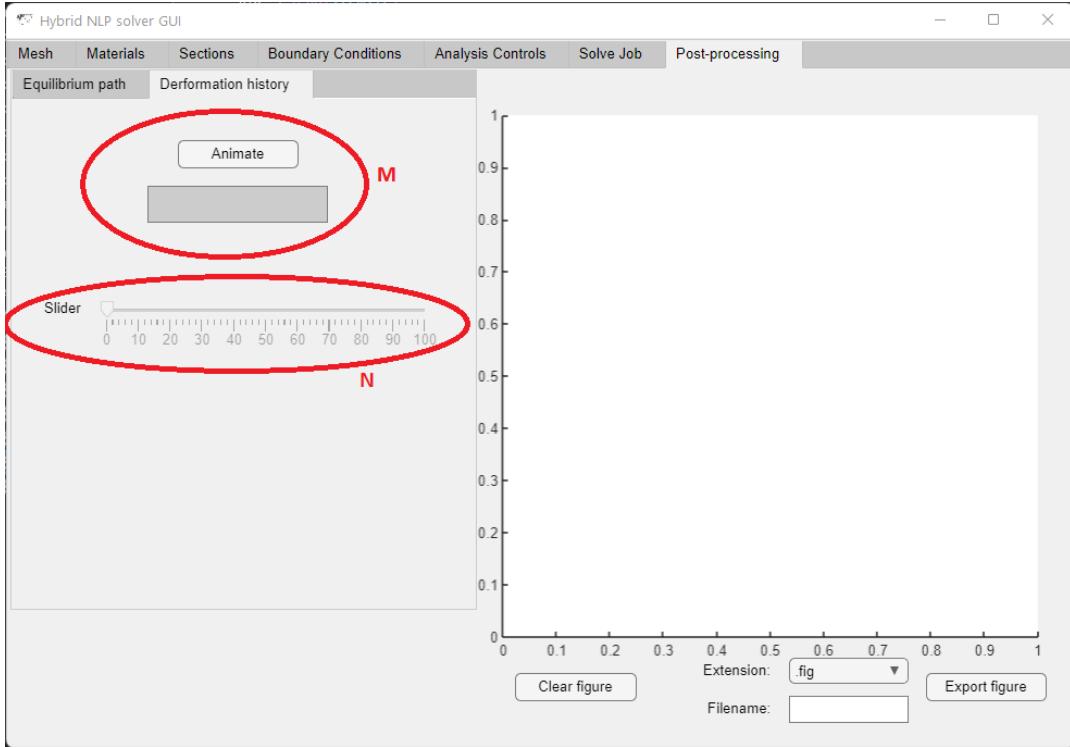
In the *Post-processing* tab the user can choose to plot i) load - displacement equilibrium paths for selected DOFs and ii) the deformation history of the structure. The tab layout is shown in Figs. G.22, G.23, and tab layout, along with the panels pertaining to *Equilibrium path* and *Deformation history*, are discussing below:

- **A:** The *Post-processing* tab.
- **B:** The *Equilibrium path* panel.
- **C:** The *Deformation history* panel.
- **D:** Here, the user selects a node from the drop-down list and a displacement DOF for the (horizontal) X-axis from the *Nodal DOF* button group.
- **E:** Here, in a similar fashion with **D**, the user selects a node from the drop-down list and a DOF from the button group to the right where an external load is specified with non-zero value. The force will be assigned to the (vertical) Y-axis.
- **F:** In these three text fields, the user specifies the labels for X and Y axis and the title for the plot.
- **G:** Here, the user specifies the color of the equilibrium curve. The color format follows native MATLAB rules.
- **H:** In this (text) field, the user specifies the linestyle of the equilibrium curve. This, again, follows standard MATLAB formatting rules. For example, a dashed line is specified as “-”, a line with cycle markers as “-o” etc.
- **I:** Once activated, the *Plot equilibrium path* button plots the equilibrium path for the selected DOFs.
- **J:** Once the *Clear figure* button is activated, it clears the figure in the present tab.



**Figure G.22.** The *Post-processing* tab and the *Deformation history* panel.

- **K:** Option to export the current plot. The user can specify a name in the text field *Filename* and a filetype from the *Extension* drop-down list. The available extensions are: ".pdf" and ".fig", which exports the plot as a MATLAB figure.
- **L:** The embedded axes for this tab where graphic objects are displayed.
- **M:** The *Animate* button in the *Deformation history* panel (see Fig. G.23) generates all the necessary data in order to produce a deformation history animation for the structure. This may take a while in case a plastic analysis or a small step-length general nonlinear analysis is performed. Once the deformation data have been generated, the *Animate* button can be used again to produce the same animation but without repeating the costly post-processing calculations.
- **N:** Slider that can be used to contrast intermediate configuration states for the structure, compared to the last one, which is the last frame produced by the animation. The slider



**Figure G.23.** The *Solve Job* tab and the *Deformation history*.

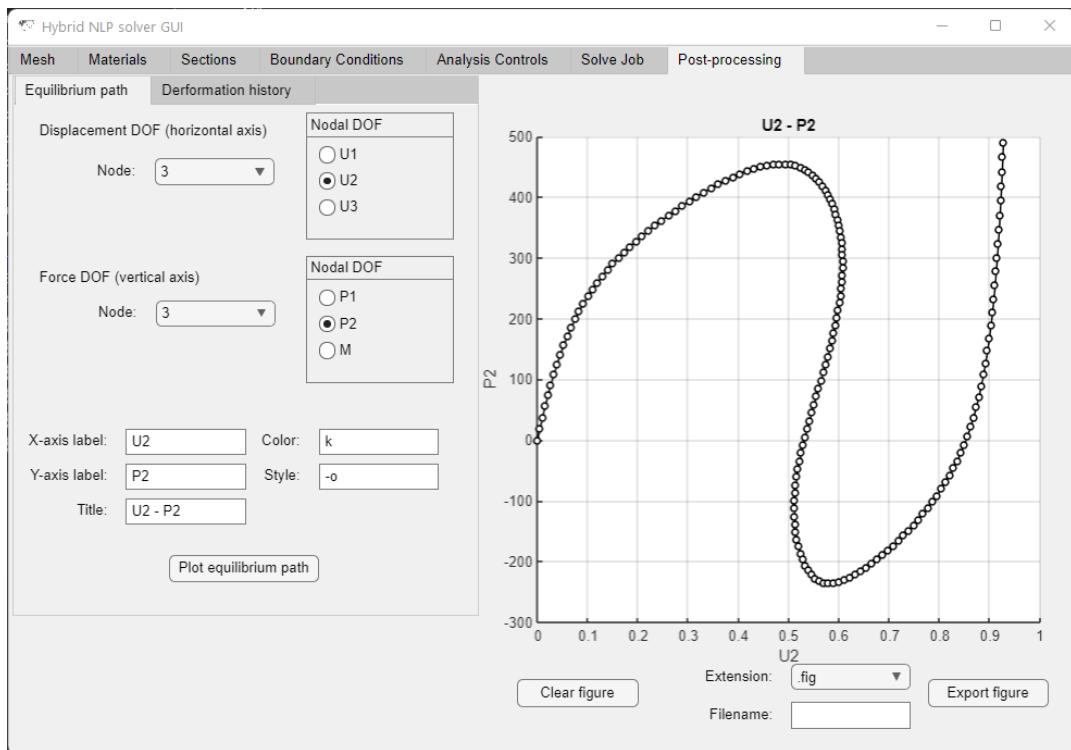
is not enabled when the deformation data have not been produced.

Plotting using these two utilities are shown in Figs. G.24, G.25. Figure G.24 shows the equilibrium path for U2 at node 3 against P2 again at node 3. Figure G.25 on the other hand shows the final configuration of Lee’s Frame along with the configuration at step 12, which is retrieved using the slider utility.

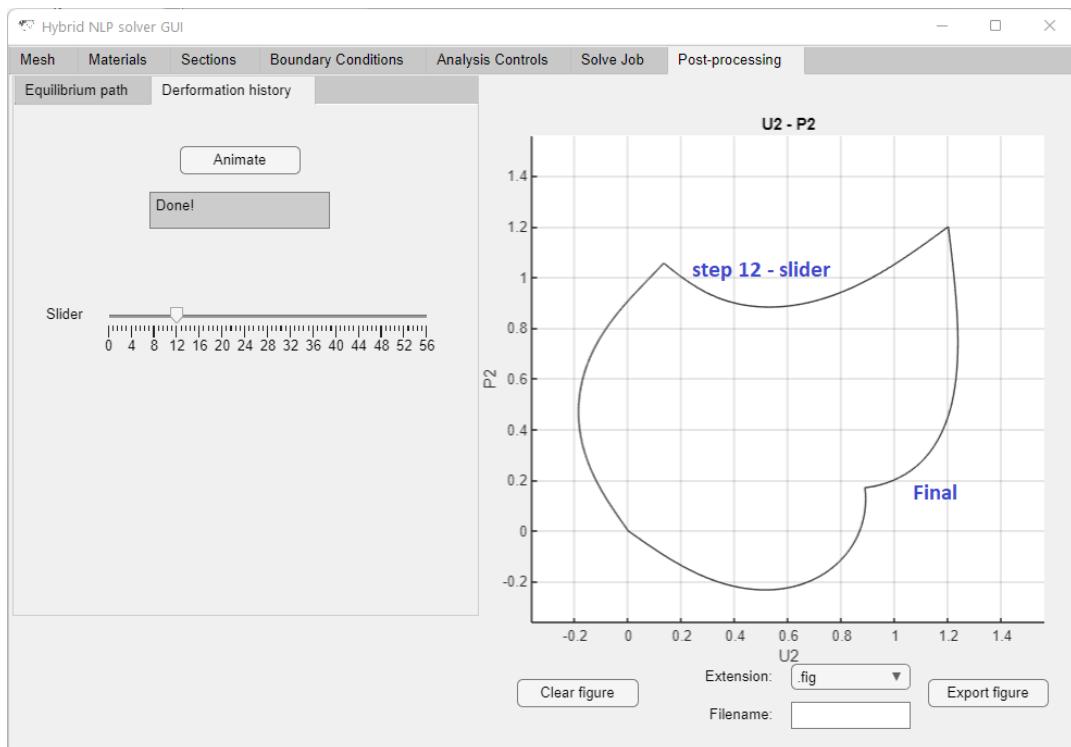
## The Input File

We now discuss the structure of the input file and how the application parses its contents. We highlight again that the input file contains all data required to run an analysis for a particular structure: mesh, materials, cross-sections and element-section assignments, boundary conditions and analysis controls.

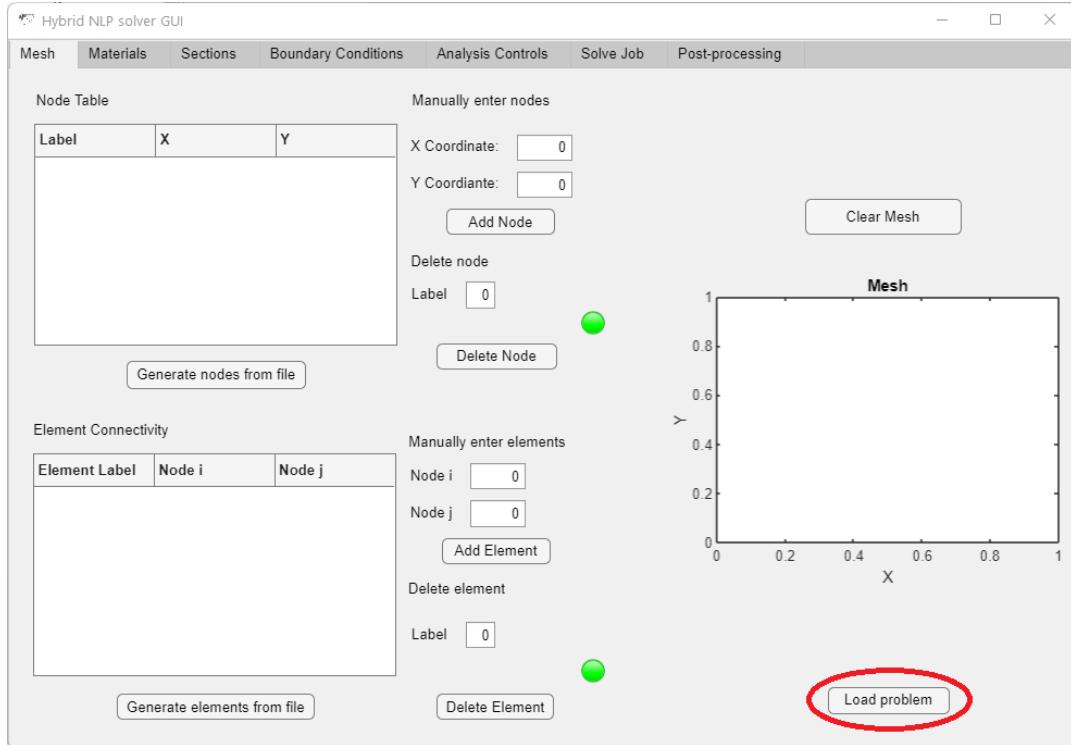
The input file is imported using the *Load problem* found in the *Mesh* tab, which is highlighted in the figure below:



**Figure G.24.** Equilibrium path plot.



**Figure G.25.** Deformation history plot of the structure.



**Figure G.26.** Load problem using the input file.

The input file consists of seven sections or parts and are separated by “.”, parsing the file from top to bottom. Each part pertains to a certain category of inputs. The categories of inputs are listed below and they have to follow this hierarchy:

1. NODES
2. ELEMENTS
3. MATERIALS (6 inputs per material, repeat input for additional materials)
4. SECTIONS (7 inputs per section, repeat input for additional sections)
5. ELEMENT-SECTION ASSIGNMENTS
6. BOUNDARY CONDITIONS
7. ANALYSIS CONTROLS (25 inputs in total)

The input file is loaded and converted into a MATLAB Cell data structure. Inputs that require more than one numeric value, such as nodal coordinates for a node, should be separated by a comma. The use of commas results in a cell structure with columns, where a new column starts after a comma. Lines that start with “#” are treated as comments and are ignored by the parser. In addition, different sections are assigned different “widths”, where width is to be understood as the number of columns the data for a particular section span. For example, the NODE and ELEMENTS input sections require two columns, therefore the parser will ignore anything written after the second column for these two sections. For the MATERIALS or SECTIONS parts, only one column suffices and therefore, comments in these parts can start at the second column. For the BOUNDARY CONDITIONS part we need four columns, that is, the node label and values at the three DOFs at that node. Therefore, the parser will start ignoring input after the fourth column in that section.

### The NODE input

An example of nodal coordinate input on the ASCII input file and how it is stored into a MATLAB cell is shown in Fig. G.27. As can be seen from that figure, the NODE part requires two numeric values per line, where a line represents a node: a X and a Y coordinate, which correspond to the first and second column respectively and are separated by commas. When the nodal input is finalized, the dot character followed by a comma, “.,” should be added in the next line. This tells the parser that nodal input is over and the ELEMENTS input is next.

Note the way the first line is typeset in G.27a. It is crucial that the first line in the input file starts with the comment character # and also includes at least 4 commas. This is done in order to ensure that MATLAB will store all data in a cell with **at least four columns**.

### The ELEMENTS input

Like the NODE part, the ELEMENT input takes two numeric values per line (element), which correspond to nodes  $I$  and  $J$  that the element connects. The relevant portion of the ASCII file

```

File Edit View
#,NODES,1,2
0.0,0.0
0.0,1.2
0.24,1.2
1.2,1.2
..
Ln 1, Col 12

```

(a) ASCII file formatting.

```

6×4 cell array

{ '# NODES' }    {1×1 missing}    {1×1 missing}    {1×1 missing}
{ [ 0] }          { [ 0] }        {1×1 missing}    {1×1 missing}
{ [ 0] }          { [ 1.2000] }   {1×1 missing}    {1×1 missing}
{ [ 0.2400] }    { [ 1.2000] }   {1×1 missing}    {1×1 missing}
{ [ 1.2000] }    { [ 1.2000] }   {1×1 missing}    {1×1 missing}
{ '.' }          {1×1 missing}    {1×1 missing}    {1×1 missing}

```

(b) MATLAB Cell storage.

**Figure G.27.** How NODE part is typeset in the ASCII file and how it is stored in a cell data structure by MATLAB.

is shown in Fig. G.28.

### The MATERIALS input

In this part the user defines a list of materials. Only one column is required and 6 parameters define a material completely. Multiple materials can be defined by repeating the sequence of MATERIALS input, as shown in Fig. G.29 for three materials. The order in which the material parameters are defined needs to be the same as shown in pictures:

```

File Edit View
# ELEMENTS
1,2
2,3
3,4
..
Ln 11, Col 3 100% Windows (CRLF) UTF-8

```

**Figure G.28.** How ELEMENTS part is typeset in the ASCII file.



The screenshot shows a text editor window with the following content:

```

File Edit View
# MATERIALS
Steel01,      #mat_name - FIRST MATERIAL
70.61e9,      #Young's modulus
0.3,          #Poisson's ratio
1.02e9,       #Yield stress
70.61e4,      #Isotropic hardening
70.61e4,      #Kinematic hardening
MAT02,        #mat_name - SECOND MATERIAL
200e9,        #Young's modulus
0.3,          #Poisson's ratio
2e6,          #Yield stress
0,            #Isotropic hardening
0,            #Kinematic hardening
MAT03,        #mat_name - THIRD MATERIAL
1.5e7,        #Young's modulus
0.35,         #Poisson's ratio
4.2e4,        #Yield stress
7.2e2,        #Isotropic hardening
7.2e3,        #Kinematic hardening
.

```

Ln 30, Col 32 | 100% | Windows (CRLF) | UTF-8

**Figure G.29.** How MATERIALS part is typeset in the ASCII file, with multiple materials defined.

1. Material Name (String type)
2. Young's modulus (Numeric type)
3. Poisson's ratio (Numeric type)
4. Yield stress (Numeric type)
5. Isotropic Hardening modulus (Numeric type)
6. Kinematic Hardening modulus (Numeric type)

### The SECTIONS input

Sections are defined in a similar fashion to materials. One section is completely determined by seven parameters which are detailed below and have to be defined in the same order when they are typeset in the ASCII file:

1. Section Name (String type)
2. Cross-section identifier (Numeric type, 1:Symmetric wide flange or 2:Rectangular)



```

File Edit View
# SECTIONS
Sec1,      #section_name - FIRST SECTION
2,         # crossflag, 1: symmetric wide-flange, 2: rectangular
0.02,      # h,   cross-section height
0.03,      # b,   cross-section width
0.02,      # tw,  web thickness
0.02,      # th,  flange thickness
Steel01,   # material of section
Sec1,      #section_name - SECOND SECTION
1,         # crossflag, 1: symmetric wide-flange, 2: rectangular
0.02,      # h,   cross-section height
0.03,      # b,   cross-section width
0.02,      # tw,  web thickness
0.02,      # th,  flange thickness
MAT02,    # material of section
Sec1,      #section_name - THIRD SECTION
2,         # crossflag, 1: symmetric wide-flange, 2: rectangular
0.02,      # h,   cross-section height
0.03,      # b,   cross-section width
0.02,      # tw,  web thickness
0.02,      # th,  flange thickness
MAT03,    # material of section
.

Ln 35, Col 44          100%          Windows (CRLF)          UTF-8

```

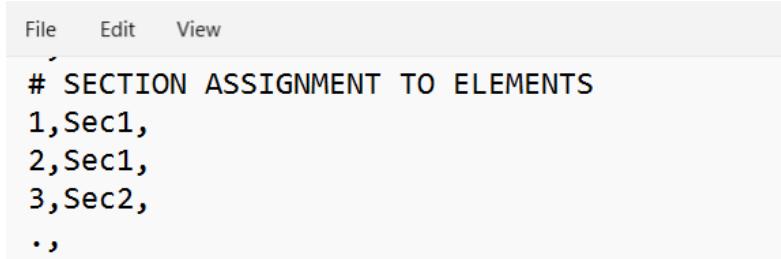
**Figure G.30.** How SECTIONS part is typeset in the ASCII file, with multiple cross-sections defined.

3. Cross-section height (Numeric type)
4. Cross-section width (Numeric type)
5. Web thickness (Numeric type)
6. Flange Thickness (Numeric type)
7. Assigned Material (String type)

In the case where a Rectangular section is defined, the web and flange thicknesses can be assigned dummy values. Again, multiple sections can be defined by repeating the above sequence of parameters. An example where three cross-sections are defined in the SECTIONS part is shown in Fig. G.30 below.

### The ELEMENT-SECTION ASSIGNMENT input

Here the user specifies couples that encapsulate information regarding cross-section assignment to elements. The first input in the couple, and by extension elements in the first column, is the

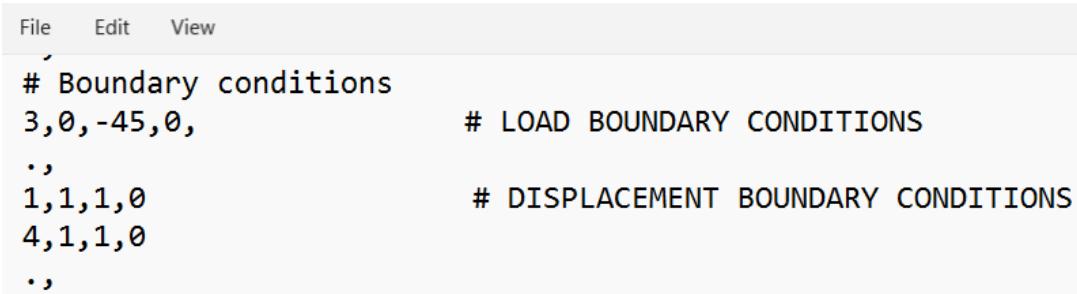


```

File Edit View
# SECTION ASSIGNMENT TO ELEMENTS
1,Sec1,
2,Sec1,
3,Sec2,
.,

```

**Figure G.31.** The ELEMENT-SECTION ASSIGNMENT input block.



```

File Edit View
# Boundary conditions
3,0,-45,0,          # LOAD BOUNDARY CONDITIONS
.,
1,1,1,0              # DISPLACEMENT BOUNDARY CONDITIONS
4,1,1,0
.,

```

**Figure G.32.** Defining load and displacement boundary conditions in the BOUNDARY CONDITIONS block for the ASCII input file.

element label, which is represented by an integer. The second input in the couple is the cross-section name, which is stored in a variable of type "string". Evidently, the number of rows dedicated to this input section should match the number of elements defined previously. The input template for assigning sections to elements is shown in Fig. G.31 below.

### The BOUNDARY CONDITION input

In this part, the user first specifies the loading boundary conditions and then the displacement ones. The two blocks are separated by the dot/comma character ".," as is done for normal input sections. Four entries are required to fully define a boundary condition: a node label, which is a positive integer, and three values corresponding to the three DOFs at that node. Figure G.32 shows an example where Node 3 is assigned a vertical downward load and rollers are assigned to Nodes 1 and 4.

```

File Edit View

. ,
# ANALYSIS CONTROLS
1,          # Enable geometrically exact kinematics, 0=disable
0,          # Enable plasticity, 0=disable
15,         # Number of cross-section layers
1,          # shear flexible flag, 0=disable
1,          # Quadrature flag: 0=Gauss-Legendre, 1=Gauss-Lobatto
5,          # Number of quadrature points
40,         # Number of incremental steps
200,        # Max number of incremental steps
10,         # Max number of iterations per step
1e-8,       # Tolerance
2,          # Solution scheme: 1-Load Control, 2-Arc-length, 3-Homotopy solver
0.05,       # Step-length DS (relevant only for arc-length and homotopy)
0,          # force-stop parameter - disabled
0,          # repeat step if no convergence param - disabled
3.9,        # IMPORTANT TO END INPUT FILE WITH ".,"# Step-length adjustment parameter-OPTIONAL
0.025,      # minDS - OPTIONAL
0.4,        # maxDS - OPTIONAL
. ,
1,          # Predictor : 1-Tangent predictor, 2-WLS predictor
1,          # Corrector : 1-Normal Flow, 2-Secant-Length, 3-Normal Plane
. ,
1,          # ENTER only Predictor = WLS
1,          # WLS variant
3,          # Number of initial load controlled steps
2,          # Degree of polynomial basis
3,          # Number of points supplied to the WLS fitting
0.2,        # Weighting function parameter
. ,
Ln 55, Col 88 | 100% | Windows (CRLF) | UTF-8

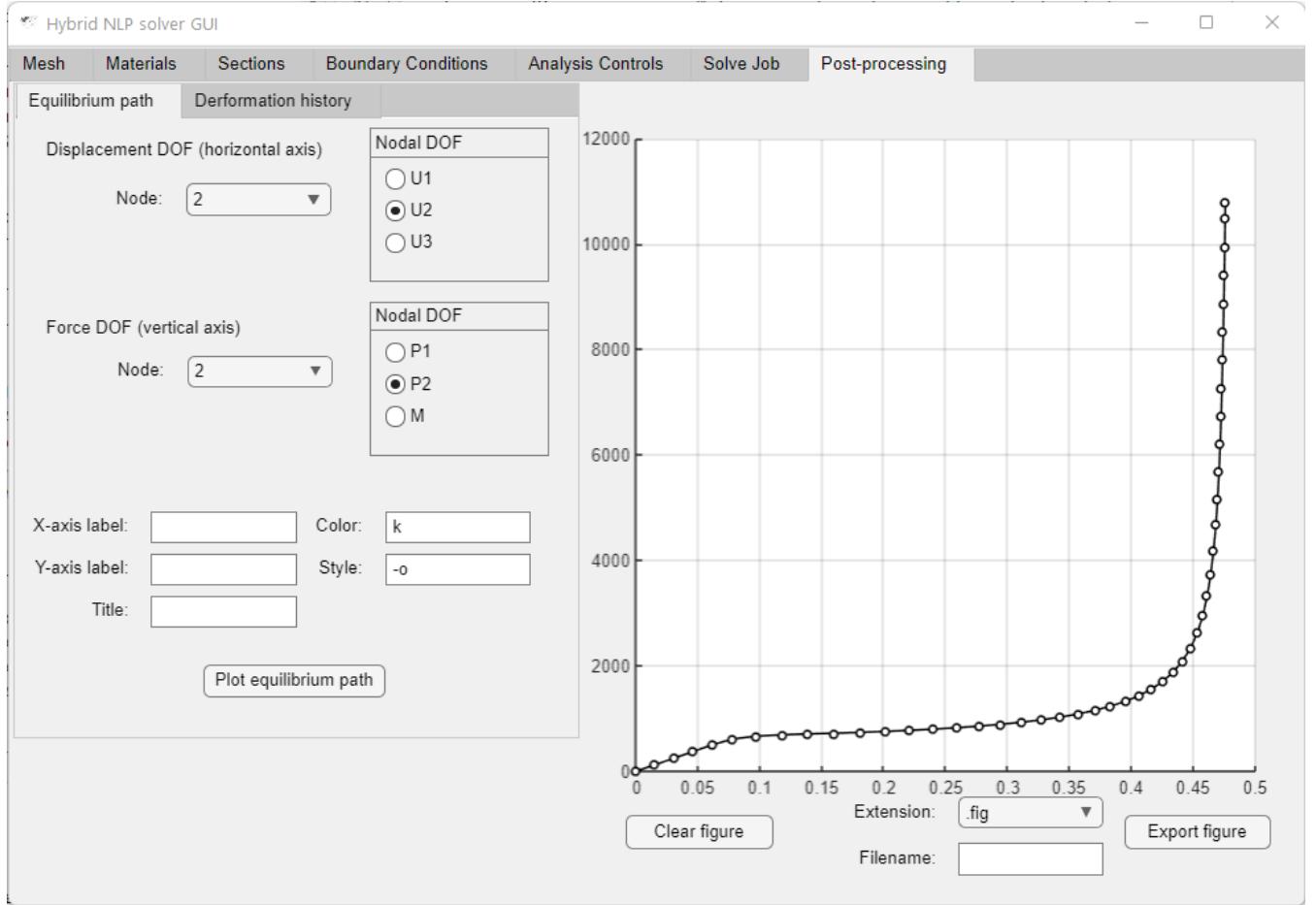
```

**Figure G.33.** The ANALYSIS CONTROLS template block for the ASCII input file.

### The ANALYSIS CONTROLS input

The final input block pertains to the *Analysis Controls* tab in the GUI. 25 input parameters (rows) and only one column are required to fully specify and register the analysis settings. However, some of these options are read if the Homotopy solver is enabled. That leaves us with 18 “core” parameters for this input block. The last three, which pertain to step-length adjustment parameter  $a$  and minimum and maximum step-lengths,  $\text{min}DS$  and  $\text{max}DS$  respectively, are optional. If they are not specified, they are assigned the following default values:

- $a = 3.9$
- $\text{min}DS = DS/4$
- $\text{max}DS = 4DS$



**Figure G.34.** The equilibrium plot for the elastoplastic cantilever, using the GUI post-processor utility.

where  $DS$  is the standard step-length used by the arc-length and homotopy solvers. The template for this input block is shown in Fig. G.33 below:

### A Simple Problem Input File

Here we show the input file and the U2-P2 equilibrium path on the application GUI for the elastoplastic cantilever problem analyzed in Chapter 2, Section 2.5.1.

### Input File for Matlab Source Code

In this final section we will outline the basic structure of the input file when the user intends to use the source code directly. The source code is written in MATLAB and the problem input

The screenshot shows a Windows Notepad window with the title "Cantilever with tip end load - Notepad". The window contains an input file for a finite element analysis, specifically for a cantilever beam. The file includes sections for nodes, elements, materials, sections, element-section assignments, boundary conditions, and analysis controls. The code uses comments starting with '#' and contains numerical values for dimensions and material properties.

```
#,NODES,1,2
0.0,0.0,      # Node 1
0.5,0.0,      # Node 2
.,
# ELEMENTS
1,2,          # Element 1
.,
# MATERIALS
Mat_1,         # material name
2e11,          # Young's modulus
0.3,           # Poisson's ratio
2e9,           # Yield stress
0,             # Isotropic hardening modulus
2e8,           # Kinematic hardening modulus
.,
# SECTIONS
Sec_1,         # section name
2,             # flag for rectangular section shape
0.03,          # cross-section height
0.03,          # cross-section width
0,             # web-thickness, dummy value here
0,             # flange thickness, dummy value here
Mat_1,         # material assigned to the section
.,
# ELEMENT-SECTION ASSIGNMENTS
1,Sec_1,       # element 1 assigned section "Sec_1"
.,
# BOUNDARY CONDITIONS
2,0,4.32e5,0,  # Vertical upwards load at node 2, DOF 2
.,
1,1,1,1,       # Clamped left end of the cantilever
.,
# ANALYSIS CONTROLS
1,              # Enable geometrically exact kinematics, 0=disable
1,              # Enable plasticity, 0=disable
15,             # Number of cross-section layers
1,              # shear flexible flag, 0=disable
1,              # Quadrature flag: 0=Gauss-Legendre, 1=Gauss-Lobatto
5,              # Number of quadrature points
40,             # Number of incremental steps
200,            # Max number of incremental steps
10,             # Max number of iterations per step
1e-8,            # Tolerance
2,              # Solution scheme: 1=Load Control, 2=Arc-length, 3=Homotopy solver
0.05,           # Step-length DS (relevant only for arc-length and homotopy)
0,              # force-stop parameter - disabled
0,              # repeat step if no convergence param - disabled
.,              # IMPORTANT TO END INPUT FILE WITH ","
```

**Figure G.35.** The input file for a simple cantilever problem.

files are matlab functions. Figure G.36 below shows the core pre-processing and processing routines of the **main.m** file, which can be considered as the first and most primitive interface of the hybrid code.

Before the input file is read, the code ensures that the existing data in workspace memory are cleared and pertinent folders are added the search path. In addition, the directory *output* is created if it does not exist already. If it does, all files inside that directory are deleted. This is the folder where we save the files where we write analysis output.

Then the problem input file is called (line 14 in **main.m**). The matlab function does not take any argument since everything is prepared by the user in the respective input file. However, in the case repeated parametric analyses are to be performed, the user will need to modify the function definition in the relevant input file so that it accepts pertinent arguments, if necessary. The input file function returns three Matlab **struct** data structures:

- **snode**: Data structure where the coordinates, elements and DOF status are stored for each node. Its dimension is equal to the total number of nodes.
- **element**: Data structure where all parameters relevant to the all elements are stored. Its dimension is equal to the total number of elements.
- **anlsData**: Data structure where the general analysis settings are stored. It contains 28 fields which will be briefly discussed below.

Next, the *processInput.m* routine is called(line 18 in **main.m**) in order to perform the necessary pre-processing steps on the aforementioned data structures. Finally, the main processing routine, **solveJob.m**, is called once pre-processing is complete. It accepts the **anlsData** and **element** data structures, which it updates during the analysis and then returns them to be used by the post-processing routines.

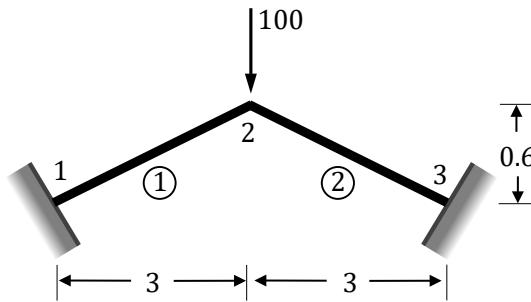
When it comes to the input file layout, the user does not have to follow a strict sequence for parameter definition, however, it is highly recommended that a structured approach similar to

```

main.m x +
1 clear;clc;clear numLinSolves;clear numFunEvals;clear numStepReductions;
2
3 if (~exist('output','dir'))
4     mkdir output
5 else
6     delete('output/*.dat');
7 end
8
9 utilpath = strcat(pwd,'/utilities'); addpath(genpath(utilpath));
10 examplespath = strcat(pwd,'/examples'); addpath(examplespath);
11 %% Input
12
13 % doc00
14 [snodes, element, anlsData] = ex4();
15 % ===== END OF INPUT =====
16
17 %% Pre-proc
18 [element, snodes, anlsData] = processInput(element, snodes, anlsData);
19
20 %% Solve
21 [anlsData,element] = solveJob(anlsData,element);
22

```

**Figure G.36.** The **main.m** file of the hybrid code.



**Figure G.37.** The geometry of the frame used as input.

the GUI input file is followed. In order to facilitate the exposition, a simple two-beam structure shown in Fig. G.37 will be used as input case.

All sections in the matlab input file are shown in the figure sequence G.38, G.39, G.40, G.41.

The parts pertinent to material and geometry definitions are similar in structure to the GUI input file. It is required that the user specifies a flange and web width even if a rectangular section is to be used. The values will be ignored in this case. The same holds for the fiber discretization. An integer value has to be assigned to variable *nlayer* (line 18) even if an elastic analysis is scheduled. In that case, the value is, again, ignored.

As far as node and element definition, the user must assign the pertinent data directly to

```

1 %% Two-beam Frame
2 function [snodes, element, anlsData] = ex3()
3
4 % Material properties
5 E = 200e9; % Young's modulus
6 nu = 0.3; % Poisson's ratio
7 h_iso = 0; % Isotropic hardening modulus
8 h_kin = 1/100*E; % Kinematic hardening modulus
9 fy = 2e9; % Yield Stress
10
11 % Section Geometry
12 h=0.3; % Cross-section height
13 b=0.3; % Cross-section width
14 tw = 0.002; % web width if wide-flange
15 tf = 0.002; % flange width if wide flange
16
17 % Cross-section layers (for plasticity)
18 nlayer = 31;
19
20 % Shear modulus
21 G = E/(2*(1+nu));
22
23 % NODES
24 snodes(1).coor = [0.0 0.0];
25 snodes(2).coor = [3.0 0.6];
26 snodes(3).coor = [6.0 0.0];
27
28 % ELEMENTS
29 element(1).nodes = [1 2];
30 element(2).nodes = [2 3];
31
32 % Number of quadrature points (shared by all elements)
33 numip = 5;
34
35 % Loading
36 Fext =100;
37
38 % Apply loading to nodes [nodenum dof1 dof2 dof3]
39 Pc = []; % Constant loads during incremental analysis
40 P = [2 0 -Fext 0]; % Loads to be incremented during analysis

```

**Figure G.38.** First portion of Matlab input file.

```

38 % Apply loading to nodes [nodenum dof1 dof2 dof3]
39 Pc = []; % Constant loads during incremental analysis
40 P = [2 0 -Fext 0]; % Loads to be incremented during analysis
41
42 % Boundary conditions [nodenum dof1 dof2 dof3]
43 bc = [1 1 1 1;3 1 1 1]; % 0 = free, 1 = fixed
44
45 % ~~~~~ Analysis settings ~~~~~
46 % Formulation
47 f = 2; % 1: full hessian, 2: hessian decomposition
48
49 % Kinematic assumption
50 nlgeomflag = 1; % 0: small displacements, 1: geometrically exact
51
52 % Solution Scheme Flag
53 solflag = 4; % 1: Newton, 2: Arc-Length, 3: Homotopy, 4: Displacement control
54
55 % Element cross-section flag
56 crossflag = 2; % 1: symmetric wide-flange, 2: rectangular
57
58 % plasticity flag
59 fibflag = 0; % 0: off, 1: on
60
61 % Numerical integration flag
62 numflag = 2; % 0: Gauss, 1: Lobatto
63
64 % Displacement control parameters
65 uc_DOF = [2 0 1 0]; % component 2 of node 2 controlled
66 u_target = 0.9; % Target displacement for controlled DOF
67
68 % Incremental procedure settings
69 steps = 50; % in how many steps should P be applied
70 tol = 1.e-8; % Newton iteration tolerance
71 iters = 8; % Newton maximum iterations per increment

```

**Figure G.39.** Second portion of Matlab input file.

```

main.m ex3.m +
68 % Incremental procedure settings
69 steps = 50; % in how many steps should P be applied
70 tol = 1.e-8; % Newton iteration tolerance
71 iters = 8; % Newton maximum iterations per increment
72
73 % These are only for arc-length/Homotopy scheme
74 max_steps = 200; % max steps for arc-length
75 DS = 0.02; % Initial step-length
76 minDS = DS/4; % min allowed step-length
77 maxDS = 2*DS; % max allowed step-length
78 a = 3.9; % For elastic, keep a between 3 and 4.
79 force_stop= 0; % force analysis to stop at lambda = 1; ON if 1, OFF if 0;
80
81 % This is only for the homotopy scheme
82 NumN = 4; % Perform NumN load control steps at first
83 COR = 1; % Determines the corrector scheme. If COR = 1, Normal flow, ...
84 % ...COR = 2, Secant-length, COR = 3, Normal Plane
85 s_con = 1; % Whether to perform arc-length type step-length..
86 % ..controls. 0 = OFF, 1 = ON.
87 PRE = 2; % Determines the predictor scheme...
88 %... If PRE = 1, Tangent predictor, PRE = 2, WLS
89
90 % Parameters for WLS predictor (used only if PRE = 2)
91 m = 2; % Order of polynomial degree
92 No = 3; % Number of points passed (SHOULD BE NumN >= No-1)
93 beta = 0.2; % Controls the weighting function
94 WLS_par = 4; % parameter that determines which WLS predictor...
95 % ...is used (1: WLSE, 2: WLST, 3: WLSIT)
96
97
98 % Element material and geometric data
99 element(1).data = [E G h b tf tw crossflag h_iso h_kin fy nlayer nu];
100 element(1).nquads = numip;
101
102 element(2).data = [E G h b tf tw crossflag h_iso h_kin fy nlayer nu];
103 element(2).nquads = numip;
104
105 % Assemble
106 element(1).nodI = snode(element(1).nodes(1)).coor;
107 element(1).nodJ = snode(element(1).nodes(2)).coor;
108 element(2).nodI = snode(element(2).nodes(1)).coor;
109 element(2).nodJ = snode(element(2).nodes(2)).coor;
110

```

**Figure G.40.** Third portion of Matlab input file.

```

105 % Assemble
106 element(1).nodI = snode(element(1).nodes(1)).coor;
107 element(1).nodJ = snode(element(1).nodes(2)).coor;
108 element(2).nodI = snode(element(2).nodes(1)).coor;
109 element(2).nodJ = snode(element(2).nodes(2)).coor;
110
111 % Process element cross-sections
112 element = getSectionData(element);
113
114 % AnalysisData
115 anlsData.incrs = steps;
116 anlsData.force = P;
117 anlsData.bcond = bc;
118 anlsData.nquads = numip;
119 anlsData.tol = tol;
120 anlsData.maxiter = iters;
121 anlsData.formulation = f;
122 anlsData.fflag = fibflag;
123 anlsData.solscheme = [solflag max_steps DS a minDS maxDS NumN COR s_con PRE...
124 m No WLS_par force_stop beta];
125 anlsData.geomflag = nlgeomflag;
126 anlsData.numintflag = numflag;
127 anlsData.DisplControl = [uc_DOF u_target];
128 anlsData.ConstantForce = Pc;
129
130 printAnalSettings(anlsData)

```

**Figure G.41.** Fourth and final portion of Matlab input file.

the corresponding structure. For the definition of node  $n$ , a  $1 \times 2$  vector containing the  $x$  and  $y$  coordinates is assigned directly to field *coor* of the  $n$ -th component of structure **snode** as follows:

$$\mathbf{snode}(n).\mathbf{coor} = [1.25 \ 0.5];$$

See also lines 24-26 in Fig. G.38. In a similar fashion, for the definition of element  $m$ , we assign the vector  $[node_I \ node_J]$  directly to field *nodes* of the  $m$ -th component of structure **element**, as shown in lines 29-30 of Fig. G.38. The general format is:

$$\mathbf{element}(m).\mathbf{nodes} = [2 \ 3];$$

where element  $m$  connects  $node_I = 2$  and  $node_J = 3$ .

For each parameter that needs to be defined by the user there are brief but detailed tooltips in the form of comments on the same line. Two features that are available in this form of input but are not in the GUI version are 1) the option to define “gravity” loads which remain constant throughout the analysis and 2) the option to perform a displacement control incremental

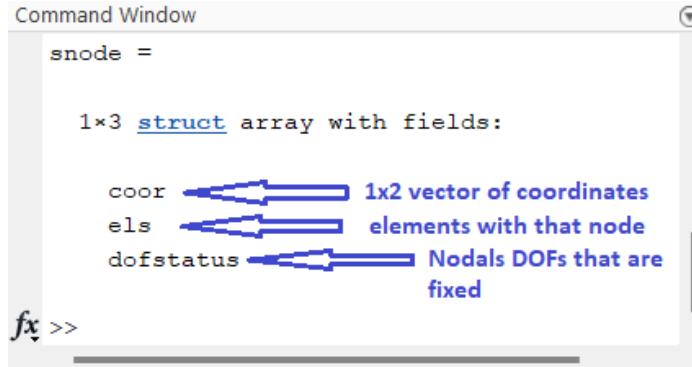
analysis. If this option is chosen (see line 53 in Fig. G.39), then the user needs to also specify the nodal DOF that is controlled and the target displacement up to which the said DOF is to be incremented to. The relevant lines for these definitions are 65-66 in Fig. G.39.

The user can choose any name for the variables they define, however they should ensure that all variables are stored in a consistent fashion and the correspondence with the basic data structures **snode**, **element** and **anlsData** is not spoiled. As can be seen in the third and fourth portions of the input file in Figs. G.40, G.41, if a different name is chosen for a parameter (left-hand side during definition), then the same name has to be assigned to the relevant structure field (right-hand side during assignment). In addition, using matlab functions to do analysis with the hybrid code, while more versatile, it is also more error-prone. The numerous error checks implemented for the GUI version are not carried out when setting up a problem using matlab input files, so users should be more cautious in the latter case.

### *The main data structures*

As mentioned earlier, the three main data structures are 1) **snode**, 2) **element** and 3) **anlsData**. For the structure in Fig. G.37, the **snode** data structure with its fields are shown Fig. G.42. For a particular node m, **snode(m)**, the *els* field is a vector with components the labels (integers) of those elements that include node m. The field *dofstatus* returns a vector that contains the nodal DOFs for node m that are restrained. For example, since node 1 in the frame above is fixed on the ground, then the field returns **snode(1).dofstatus = [1 2 3]** since all 3 DOFs are restrained. If no DOFs are restrained, *dofstatus* returns an empty vector.

Next, we have the **element** structure, which is used to store data pertaining to individual elements that will be used and updated during analysis and, then, stored for post-processing purposes. The relevant fields are shown in Fig. G.43 for the 2-beam frame. We can see that **element** is of dimension 2, since there are only two elements. For most fields, the description added in the figure is adequate. The field *data* is a  $1 \times 11$  vector that contains the following parameters:



**Figure G.42.** The **snode** data structure.

$$\text{element}(m).\text{data} = [E \ G \ A \ I \ H_{iso} \ H_{kin} \ n_l \ k_s \ u_p]$$

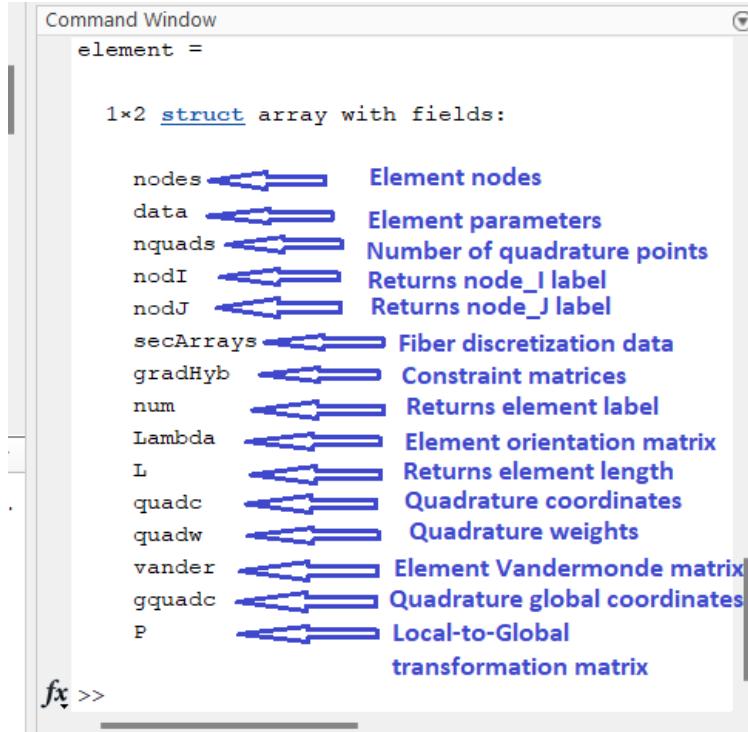
where these parameters are: Young's modulus, shear modulus, cross-section area, moment of inertia, isotropic hardening modulus, kinematic hardening modulus,  $n_l$  is the number of layers, shear coefficient and, latsly,  $u_p$  is a parameter given by the following integral over a cross-section:

$$u_p = \int_A \varphi(X_2)^2 dA$$

and is used to determine the shear coefficient  $k_q$  for quadratic shear strain distribution (see Eqs. 3.31, 3.32). The *secArrays* field returns a  $2 \times n_l$  vector where the first column contains the area of fibers and the second contains the heights of these fibers on the cross-section. The field *gradHyb* stores matrix  $\nabla_{y_i} \vec{h}_g^B$ , given in Appendix B, at each element quadrature point.

Lastly, we use the **anlsData** structure to store all parameters pertaining to the analysis settings. Its dimension is one and the relevant fields it encapsulates are shown in Fig. G.44. Again, short descriptions are provided in the figure and below we discuss certain fields whose use might not be apparent.

The *incls* field returns the number of incremental steps for the Load-Controlled Newton method. It is overridden by parameter *max\_steps* (see Line 74 in Fig. G.40), which is stored in the second place of field *solscheme*. Both *force* and *bcond* fields, which pertain to external loading and support specification respectively, are defined to in the same way as with the GUI



**Figure G.43.** The `element` data structure.

input. Field *formulation* determines which solver is called to handle the linearized system of the Lagrangian gradient. It should always be set to 2, which leads to a solution by block elimination of the Hessian. The *ConstantForce* field returns the vector of “gravity” loads that remain constant during the incremental analysis. While the proper way to handle the presence of such loads is to perform a separate initial, usually load-controlled analysis and then proceed with the nonlinear incremental analysis by using the solution of the former one as starting state, presently, no such approach is followed. Instead, all loads are assumed to be present from the start and only the loads in the *force* field are incremented. For that reason, caution is advised when setting values for the gravity loads, since large values might cause early termination due to the solution not being close to the initial state. The *Hessian* and *Gradient* fields return the updated Hessian and Gradient of the Lagrangian, respectively, for the whole structure for the current step. When the analysis is concluded, both of these objects pertain to the last step. The field *totdofs* returns the total number of DOFs for the problem, including Lagrange multipliers. The *dhis* stores nodal displacements and quadrature point strain measures for each step, which

are then used for post-processing purposes. Field  $\lambda$  is the homotopy parameter. When the load is parameterized in the context of the NPNLP framework, this field returns a vector with the values of the load proportionality parameter at each step. The field *CurveLength* stores the length of the homotopy/equilibrium curve up to each step. It returns a  $1 \times steps$  vector. Field *ResultantHis* stores the three stress resultants at all nodes and all element quadrature points at each step. If Displacement Control is selected, then the additional necessary data are stored in the *DisplControl*. The *numintflag* field determines the quadrature rule. If *numintflag*=1 then the Gauss-Lobatto quadrature will be used, while for Gauss-Legendre, the option is *numintflag*=2. Notice that this is different compared to the GUI input, where Lobatto is still 1 but Gauss-Legendre is 0. Lastly, the *solscheme* field stores the following analysis parameters:

```
anlsData.solscheme = [solflag max_steps ΔS minΔS maxΔS NumN COR s_con PRE m N_o  
WLS_par force_stop beta]
```

- **solflag** is the flag that determines which solver is used (see Line 53, Fig. G.39).
- **max\_steps** determines the maximum number of incremental steps.
- $\Delta S$  is the step-length when the arc-length or the homotopy solvers are used.
- $\min\Delta S$  is the minimum allowed step-length.
- $\max\Delta S$  is the maximum allowed step-length.
- **NumN** is the number of load-controlled steps to perform before switching to WLS predictor.
- **COR** specifies which corrector is to be used. If **COR=1**, the Normal Flow corrector is used. If **COR=2**, the arc-length/secant-length correctors are used. if **COR=3**, the Normal Plane corrector is used.
- **s\_con** determines whether step-length controls are to be performed. If **s\_con =1**, then  $\Delta S$  is adjusted at the end of each step.

- **PRE** determines which predictor will be used. If **PRE=1** then the tangent predictor is used. If **PRE=2** then the WLS predictor is used.
- **m** determines the order of polynomial basis for the WLS approximation.
- **N\_o** specifies the number of previously converged points passed to the WLS predictor.
- **WLS\_par** determines which WLS predictor variant is to be used. For **WLS\_par=1**, then the predictor is WLSE, while for **WLS\_par=2** and **WLS\_par=3**, the predictors are the WLST and WLSIT respectively.
- **force\_stop** is a flag that determines if analysis should stop when the load proportionality parameter becomes larger than 1.
- **beta** is the weighting function parameter used in the WLS predictor.

```

Command Window
>> anlsData

anlsData =
struct with fields:

    incrs: 50                                         Number of increments
    force: [2 0 -100 0]                                External load table - (node, dof1,dof2,dof3)
    bcond: [2×4 double]                               Support table - (node, dof1,dof2,dof3)
    nquads: 5                                         Number of quadrature points per element
    tol: 1.0000e-08                                    Tolerance used to check convergence
    maxiter: 8                                         Maximum number of Newton iterations
    formulation: 2                                     System solver - always set it to 2
    fflag: 0                                           Plasticity flag - 1 = on
    solscheme: [4 200 0.0200 3.9000 0.0050 0.0400 4 1 1 2 2 3 4 0 0.2000] Geometric flag : 0 = small displacement analysis
    geomflag: 1                                         Quadrature flag : 1 = Lobatto , 2 = Gauss-Legendre
    numintflag: 2
    DisplControl: [2 0 1 0 0.9000 2]                  Displacement control table
    ConstantForce: [9×1 double]                         Constant,'gravity, force vector
        Hessian: [65×65 double]                          Hessian of the Lagrangian
        Gradient: [65×1 double]                          Gradient of the Lagrangian
        totdofs: 65                                      Total number of DOFs
        dhis: [51×65 double]                            Analysis history table - used for plotting
        lambda: [51×1 double]                            Load-proportionality factor
        numel: 2                                         Total number of elements
        numnod: 3                                         Total number of nodes
        rdfs: [1 2 3 7 8 9]                            Fixed DOFs in the global system
        forceVector: [9×1 double]                          External global force vector
        axisRange: [-1.8000 7.8000 -4.5000 5.1000]     Axes range for plotting
    AnalysisOutput: [50×4 double]                        Misc. output results , including norms, iterations...
        freedofs: 59                                     Number of free DOFs
        CurveLength: [51×1 double]                       Length of equilibrium curve at solution points
        ResultantHis: [51×42 double]                      Stress resultant table at each step

fx >

```

**Figure G.44.** The `anlsData` data structure.

## BIBLIOGRAPHY

- [1] K.-J. Bathe and S. Bolourchi, “Large displacement analysis of three-dimensional beam structures,” *International Journal for Numerical Methods in Engineering*, vol. 14, no. 7, pp. 961–986, 1979.
- [2] T. Belytschko and B. Hsieh, “Non-linear transient finite element analysis with convected coordinates,” *International Journal for Numerical Methods in Engineering*, vol. 7, no. 3, pp. 255–271, 1973.
- [3] T. Belytschko, L. Schwer, and M. Klein, “Large displacement, transient analysis of space frames,” *International Journal for Numerical Methods in Engineering*, vol. 11, no. 1, pp. 65–84, 1977.
- [4] C. Rankin and F. Brogan, “An element independent corotational procedure for the treatment of large rotations,” *Journal of Pressure Vessel Technology*, vol. 108, no. 2, pp. 165–174, 1986.
- [5] M. Crisfield, “A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 2, pp. 131–150, 1990.
- [6] J. Argyris, S. Kelsey, and H. Kamel, “Matrix methods of structural analysis: A precis of recent developments,” *Matrix Methods of Structural Analysis*, vol. Pergamon Press, New York, 1964.
- [7] G. Wempner, “Finite elements, finite rotations and small strains of flexible shells,” *International Journal of Solids and Structures*, vol. 5, no. 2, pp. 117–153, 1969.
- [8] C. Felippa and B. Haugen, “A unified formulation of small-strain corotational finite elements: I. theory,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 21-24, pp. 2285–2335, 2005.
- [9] E. Reissner, “On one-dimensional finite-strain beam theory: The plane problem,” *Zeitschrift für Angewandte Mathematik und Physik ZAMP*, vol. 23, no. 5, pp. 795–804, 1972.
- [10] J. Simo, “A finite strain beam formulation. The three-dimensional dynamic problem. Part I,” *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 1, pp. 55–70, 1985.
- [11] J. Simo and L. Vu-Quoc, “A three-dimensional finite-strain rod model. Part II: Computational aspects,” *Computer Methods in Applied Mechanics and Engineering*, vol. 58, no. 1, pp. 79–116, 1986.
- [12] J. Simo and L. Vu-Quoc, “On the dynamics of flexible beams under large overall motions—the plane case: Part I,” *Journal of Applied Mechanics*, vol. 4, no. 53, pp. 849–854, 1986.
- [13] A. Cardona and M. Geradin, “A beam finite element non-linear theory with finite rotations,” *International Journal for Numerical Methods in Engineering*, vol. 26, no. 11, pp. 2403–2438, 1988.
- [14] A. Ibrahimbegovic, “On finite element implementation of geometrically nonlinear reissner’s beam theory: Three-dimensional curved beam elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 122, no. 1–2, pp. 11–27, 1995.
- [15] I. Romero and F. Armero, “An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energy-momentum conserving scheme in dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 54, no. 12, pp. 1683–1716, 2002.
- [16] J. Argyris, “An excursion into large rotations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 32, no. 1–3, pp. 85–155, 1982.

- [17] I. Romero, “The interpolation of rotations and its application to finite element models of geometrically exact rods,” *Computational Mechanics*, vol. 34, no. 2, pp. 121–133, 2004.
- [18] I. Romero, “A comparison of finite elements for nonlinear beams: The absolute nodal coordinate and geometrically exact formulations,” *Multibody System Dynamics*, vol. 20, no. 1, pp. 51–68, 2008.
- [19] A. A. Shabana and R. Y. Yakoub, “Three dimensional absolute nodal coordinate formulation for beam elements: Theory,” *Journal of Mechanical Design*, vol. 123, no. 4, pp. 606–613, 2001.
- [20] K. M. Mathisen *et al.*, “A comparative study of beam element formulations for nonlinear analysis: Corotational vs. geometrically exact formulations,” *MekIT’ 17-Ninth national conference on Computational Mechanics*, 2017.
- [21] N. Oliveto and M. Sivaselvan, “3D finite-deformation beam model with viscous damping: Computational aspects and applications,” *Journal of Engineering Mechanics*, vol. 141, no. 1, p. 04014103, 2014.
- [22] A. N. Panteli and K. V. Spiliopoulos, “Comparative study of two geometrically nonlinear beam models with different strain measures,” *International Journal of Non-Linear Mechanics*, p. 103480, 2020.
- [23] H. Santos, P. Pimenta, and J. Almeida, “A hybrid-mixed finite element formulation for the geometrically exact analysis of three-dimensional framed structures,” *Computational Mechanics*, vol. 48, no. 5, p. 591, 2011.
- [24] C. Meier, M. J. Grill, W. A. Wall, and A. Popp, “Geometrically exact beam elements and smooth contact schemes for the modeling of fiber-based materials and structures,” *International Journal of Solids and Structures*, vol. 154, pp. 124–146, 2018.
- [25] E. Marino, “Isogeometric collocation for three-dimensional geometrically exact shear-deformable beams,” *Computer Methods in Applied Mechanics and Engineering*, vol. 307, pp. 383–410, 2016.
- [26] A. Tasora, S. Benatti, D. Mangoni, and R. Garziera, “A geometrically exact isogeometric beam for large displacements and contacts,” *Computer Methods in Applied Mechanics and Engineering*, vol. 358, p. 112635, 2020.
- [27] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39-41, pp. 4135–4195, 2005.
- [28] M. Salehi and P. Sideris, “A finite-strain gradient-inelastic beam theory and a corresponding force-based frame element formulation,” *International Journal for Numerical Methods in Engineering*, vol. 116, no. 6, pp. 380–411, 2018.
- [29] I. Planinc, M. Saje, and B. Cas, “On the local stability condition in the planar beam finite element,” *Structural Engineering and Mechanics*, vol. 12, no. 5, pp. 507–526, 2001.
- [30] M. Gams, M. Saje, S. Srpcic, and I. Planinc, “Finite element dynamic analysis of geometrically exact planar beams,” *Computers & Structures*, vol. 85, no. 17-18, pp. 1409–1419, 2007.
- [31] D. Zupan and M. Saje, “Finite-element formulation of geometrically exact three-dimensional beam theories based on interpolation of strain measures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 49-50, pp. 5209–5248, 2003.
- [32] M. Saje, “Finite element formulation of finite planar deformation of curved elastic beams,” *Computers & Structures*, vol. 39, no. 3-4, pp. 327–337, 1991.
- [33] E. Bayo, J. G. de Jalón, A. Avello, and J. Cuadrado, “An efficient computational method for real time multibody dynamic simulation in fully cartesian coordinates,” *Computer Methods in Applied Mechanics and Engineering*, vol. 92, no. 3, pp. 377–395, 1991.

- [34] S. A. Kaba and S. A. Mahin, *Refined modelling of reinforced concrete columns for seismic analysis*. University of California, Earthquake Engineering Research Center, 1984.
- [35] C. A. Zeris and S. A. Mahin, “Analysis of reinforced concrete beam-columns under uniaxial excitation,” *Journal of structural engineering*, vol. 114, no. 4, pp. 804–820, 1988.
- [36] F. Taucer, E. Spacone, and F. C. Filippou, *A fiber beam-column element for seismic response analysis of reinforced concrete structures*. Earthquake Engineering Research Center, College of Engineering, University . . ., 1991, vol. 91.
- [37] E. Spacone, V. Ciampi, and F. C. Filippou, *A beam element for seismic damage analysis*. Earthquake Engineering Research Center Berkeley, CA, 1992.
- [38] A. C. Scordelis, “Computer models for nonlinear analysis of reinforced and prestressed concrete structures,” *PCI JOURNAL*, vol. 29, no. 6, pp. 116–135, 1984.
- [39] D. C. Drucker, “The effect of shear on the plastic bending of beams,” *Journal of Applied Mechanics*, vol. 23, no. 4, pp. 509–514, 1956.
- [40] F. J. Vecchio and M. P. Collins, “The modified compression-field theory for reinforced concrete elements subjected to shear,” *ACI J.*, vol. 83, no. 2, pp. 219–231, 1986.
- [41] F. J. Vecchio and M. P. Collins, “Predicting the response of reinforced concrete beams subjected to shear using modified compression field theory,” *ACI Structural Journal*, vol. 85, no. 3, pp. 258–268, 1988.
- [42] J. Bairan Garcia and A. Mari Bernat, “Shear-bending-torsion interaction in structural concrete members: A nonlinear coupled sectional approach,” *Archives of Computational Methods in Engineering*, vol. 14, no. 3, pp. 249–278, 2007.
- [43] G. Ranzo and M. Petrangeli, “A fibre finite beam element with section shear modelling for seismic analysis of rc structures,” *Journal of Earthquake Engineering*, vol. 2, no. 03, pp. 443–473, 1998.
- [44] R. Martino, E. Spacone, and G. Kingsley, “Nonlinear pushover analysis of rc structures,” in *Advanced Technology in Structural Engineering*, American Society of Civil Engineers, 2000, pp. 1–8.
- [45] A. Marini and E. Spacone, “Analysis of reinforced concrete elements including shear effects,” *ACI Structural Journal*, vol. 103, no. 5, p. 645, 2006.
- [46] R. Taylor, F. C. Filippou, A. Saritas, and F. Auricchio, “A mixed finite element method for beam and frame problems,” *Computational mechanics*, vol. 31, no. 1, pp. 192–203, 2003.
- [47] C. M. Lyritsakis, C. P. Andriotis, and K. G. Papakonstantinou, “Geometrically exact hybrid beam element based on nonlinear programming,” *International Journal for Numerical Methods in Engineering*, vol. 122, no. 13, pp. 3273–3299, 2021.
- [48] A. Papachristidis, M. Fragiadakis, and M. Papadrakakis, “A 3d fibre beam-column element with shear modelling for the inelastic analysis of steel structures,” *Computational Mechanics*, vol. 45, no. 6, pp. 553–572, 2010.
- [49] A. Saritas and F. C. Filippou, “Frame element for metallic shear-yielding members under cyclic loading,” *Journal of Structural Engineering*, vol. 135, no. 9, pp. 1115–1123, 2009.
- [50] P. Ceresa, L. Petrini, R. Pinho, and R. Sousa, “A fibre flexure–shear model for seismic analysis of rc-framed structures,” *Earthquake Engineering & Structural Dynamics*, vol. 38, no. 5, pp. 565–586, 2009.
- [51] J. N. Gregori, P. M. Sosa, M. F. Prada, and F. C. Filippou, “A 3d numerical model for reinforced and prestressed concrete elements subjected to combined axial, bending, shear and torsion loading,” *Engineering Structures*, vol. 29, no. 12, pp. 3404–3419, 2007.

- [52] A. Kagermanov and P. Ceresa, “Fiber-section model with an exact shear strain profile for two-dimensional rc frame structures,” *Journal of Structural Engineering*, vol. 143, no. 10, p. 04017132, 2017.
- [53] S. Klinkel and S. Govindjee, “Using finite strain 3d-material models in beam and shell elements,” *Engineering Computations*, vol. 19, no. 3, pp. 254–271, 2002.
- [54] R. H. Dodds Jr, “Numerical techniques for plasticity computations in finite element analysis,” *Computers & Structures*, vol. 26, no. 5, pp. 767–779, 1987.
- [55] E. A. de Souza Neto, D. Peric, and D. R. Owen, *Computational methods for plasticity: theory and applications*. John Wiley & Sons, 2011.
- [56] T. J. Ypma, “Historical development of the newton–raphson method,” *SIAM review*, vol. 37, no. 4, pp. 531–551, 1995.
- [57] J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.
- [58] C. G. Broyden, “A class of methods for solving nonlinear simultaneous equations,” *Mathematics of computation*, vol. 19, no. 92, pp. 577–593, 1965.
- [59] W. C. Davidon, “Variable metric method for minimization,” *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 1–17, 1991.
- [60] E. R. Berndt, B. H. Hall, R. E. Hall, and J. A. Hausman, “Estimation and inference in nonlinear structural models,” in *Annals of Economic and Social Measurement, Volume 3, number 4*, NBER, 1974, pp. 653–665.
- [61] H. Matthies and G. Strang, “The solution of nonlinear finite element equations,” *International journal for numerical methods in engineering*, vol. 14, no. 11, pp. 1613–1626, 1979.
- [62] T. J. Hughes, I. Levit, and J. Winget, “An element-by-element solution algorithm for problems of structural and solid mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 36, no. 2, pp. 241–254, 1983.
- [63] D. C. Liu and J. Nocedal, “On the limited memory bfgs method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [64] E. L. Allgower and K. Georg, *Introduction to numerical continuation methods*. SIAM, 2003.
- [65] W. C. Rheinboldt, “Numerical continuation methods: A perspective,” *Journal of computational and applied mathematics*, vol. 124, no. 1-2, pp. 229–244, 2000.
- [66] H. B. Keller, “Global homotopies and newton methods,” in *Recent advances in numerical analysis*, Elsevier, 1978, pp. 73–94.
- [67] T.-Y. Li and J. A. Yorke, “A simple reliable numerical algorithm for following homotopy paths,” *Analysis and computation of fixed points*, pp. 73–91, 1980.
- [68] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, “Finding zeroes of maps: Homotopy methods that are constructive with probability one,” *Mathematics of Computation*, vol. 32, no. 143, pp. 887–899, 1978.
- [69] S.-N. Chow, J. Mallet-Paret, and J. A. Yorke, “A homotopy method for locating all zeros of a system of polynomials,” in *Functional differential equations and approximation of fixed points*, Springer, 1979, pp. 77–88.
- [70] L. T. Watson, “Globally convergent homotopy methods: A tutorial,” *Applied Mathematics and Computation*, vol. 31, pp. 369–396, 1989.
- [71] L. T. Watson, “Globally convergent homotopy algorithms for nonlinear systems of equations,” *Nonlinear Dynamics*, vol. 1, no. 2, pp. 143–191, 1990.

- [72] E. L. Allgower, “A survey of homotopy methods for smooth mappings,” in *Numerical solution of nonlinear equations*, Springer, 1981, pp. 1–29.
- [73] W. C. Rheinboldt, “Solution fields of nonlinear equations and continuation methods,” *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 221–237, 1980.
- [74] W. C. Rheinboldt and J. V. Burkardt, “A locally parameterized continuation process,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 9, no. 2, pp. 215–235, 1983.
- [75] T. Wayburn and J. Seader, “Homotopy continuation methods for computer-aided process design,” *Computers & Chemical Engineering*, vol. 11, no. 1, pp. 7–25, 1987.
- [76] D. Davidenko, “On a new method of numerical solution of systems of nonlinear equations,” in *Dokl. Akad. Nauk SSSR*, vol. 88, 1953, pp. 601–602.
- [77] D. Davidenko, “On the approximate solution of systems of nonlinear equations,” *Ukr. Mat. Zh.*, vol. 5, no. 2, pp. 196–206, 1953.
- [78] R. Klopfenstein, “Zeros of nonlinear functions,” *Journal of the ACM (JACM)*, vol. 8, no. 3, pp. 366–373, 1961.
- [79] C. Haselgrove, “The solution of non-linear equations and of differential equations with two-point boundary conditions,” *The Computer Journal*, vol. 4, no. 3, pp. 255–259, 1961.
- [80] M. Yakovlev, “The solution of systems of non-linear equations by a method of differentiation with respect to a parameter,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 1, pp. 198–203, 1964.
- [81] G. A. Wempner, “Discrete approximations related to nonlinear theories of solids,” *International Journal of Solids and Structures*, vol. 7, no. 11, pp. 1581–1599, 1971.
- [82] P. Bergan, G Horrigmoe, B Bråkeland, and T. Søreide, “Solution techniques for nonlinear finite element problems,” *International Journal for Numerical Methods in Engineering*, vol. 12, no. 11, pp. 1677–1696, 1978.
- [83] P. Bergan and T. Soreide, “Solution of large displacement and instability problems using the current stiffness parameter,” *Finite elements in nonlinear mechanics*, vol. 2, pp. 647–669, 1978.
- [84] J.-L. Batoz and G. Dhatt, “Incremental displacement algorithms for nonlinear problems,” *International Journal for Numerical Methods in Engineering*, vol. 14, no. 8, pp. 1262–1267, 1979.
- [85] E Riks, “An incremental approach to the solution of snapping and buckling problems,” *International journal of solids and structures*, vol. 15, no. 7, pp. 529–551, 1979.
- [86] E. Ramm, “Strategies for tracing the nonlinear response near limit points,” in *Nonlinear finite element analysis in structural mechanics*, Springer, 1981, pp. 63–89.
- [87] M. Crisfield, “A fast incremental/iterative solution procedure that handles ”snap-through”,” *Computers & Structures*, vol. 13, no. 1–3, pp. 55–62, 1981.
- [88] A. Cassell and W Novozhilov, “Shells of revolution under arbitrary loading and the use of fictitious densities in dynamic relaxation.,” *Proceedings of the Institution of Civil Engineers*, vol. 45, no. 1, pp. 65–78, 1970.
- [89] J. Brew and D. Brotton, “Non-linear structural analysis by dynamic relaxation,” *International Journal for Numerical Methods in Engineering*, vol. 3, no. 4, pp. 463–483, 1971.
- [90] J. A. STKICKLIN, W. E. HAISLER, and W. A. VON RIESEMANN, “Self-correcting initial value formulations in nonlinear structural mechanics,” *AIAA Journal*, vol. 9, no. 10, pp. 2066–2067, 1971.

- [91] M Papadrakakis, “A method for the automatic evaluation of the dynamic relaxation parameters,” *Computer methods in applied mechanics and engineering*, vol. 25, no. 1, pp. 35–48, 1981.
- [92] K. Park, “A family of solution algorithms for nonlinear structural analysis based on relaxation equations,” *International Journal for Numerical Methods in Engineering*, vol. 18, no. 9, pp. 1337–1347, 1982.
- [93] J. Oden and J. Key, “Analysis of static non-linear response by explicit time integration,” *International journal for numerical methods in Engineering*, vol. 7, no. 2, pp. 225–227, 1973.
- [94] L. T. Watson, S. M. Holzer, and M. C. Hansen, “Tracking nonlinear equilibrium paths by a homotopy method,” Virginia Polytechnic Institute and State University, Tech. Rep., 1981.
- [95] L. T. Watson, M. P. Kamat, and M. H. Reaser, “A robust hybrid algorithm for computing multiple equilibrium solutions,” *Engineering Computations*, 1985.
- [96] W. C. Rheinboldt, “Numerical analysis of continuation methods for nonlinear structural problems,” in *Computational Methods in Nonlinear Structural and Solid Mechanics*, Elsevier, 1981, pp. 103–113.
- [97] A. B. Poore and Q Al-Hassan, “The expanded lagrangian system for constrained optimization problems,” *SIAM journal on control and optimization*, vol. 26, no. 2, pp. 417–427, 1988.
- [98] L. Zhenghua, L. Yong, and Y. Bo, “A combined homotopy interior point method for general nonlinear programming problems,” *Applied mathematics and computation*, vol. 80, no. 2-3, pp. 209–224, 1996.
- [99] C. Hillermeier, “Generalized homotopy approach to multiobjective optimization,” *Journal of Optimization Theory and Applications*, vol. 110, no. 3, pp. 557–583, 2001.
- [100] W Song and G. Yao, “Homotopy method for a general multiobjective programming problem,” *Journal of Optimization Theory and Applications*, vol. 138, no. 1, p. 139, 2008.
- [101] G. D. Byrne and L. A. Baird, “Distillation calculations using a locally parameterized continuation method,” *Computers & chemical engineering*, vol. 9, no. 6, pp. 593–599, 1985.
- [102] W.-J. Lin, J. Seader, and T. Wayburn, “Computing multiple solutions to systems of interlinked separation columns,” *AICHE Journal*, vol. 33, no. 6, pp. 886–897, 1987.
- [103] J. W. Kovach III, “Heterogenous azeotropic distillation-homotopy-continuation methods,” *Computers & chemical engineering*, vol. 11, no. 6, pp. 593–605, 1987.
- [104] A Ushida and L. Chua, “Tracing solution curves of non-linear equations with sharp turning points,” *International journal of circuit theory and applications*, vol. 12, no. 1, pp. 1–21, 1984.
- [105] C. Hughes, D. Mehta, and J.-I. Skallerud, “Enumerating gribov copies on the lattice,” *Annals of Physics*, vol. 331, pp. 188–215, 2013.
- [106] D. Mehta, T. Chen, J. W. Morgan, and D. J. Wales, “Exploring the potential energy landscape of the thomson problem via newton homotopies,” *The Journal of chemical physics*, vol. 142, no. 19, p. 194113, 2015.
- [107] H. Mobahi and J. W. Fisher, “On the link between gaussian homotopy continuation and convex envelopes,” in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2015, pp. 43–56.
- [108] M. Bilasse, I. Charpentier, Y. Koutsawa, *et al.*, “A generic approach for the solution of nonlinear residual equations. part ii: Homotopy and complex nonlinear eigenvalue method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 49–52, pp. 3999–4004, 2009.

- [109] B. Banerjee, T. F. Walsh, W. Aquino, and M. Bonnet, “Large scale parameter estimation problems in frequency-domain elastodynamics using an error in constitutive equation functional,” *Computer methods in applied mechanics and engineering*, vol. 253, pp. 60–72, 2013.
- [110] T. Ligurský and Y. Renard, “A continuation problem for computing solutions of discretised evolution problems with application to plane quasi-static contact problems with friction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 280, pp. 222–262, 2014.
- [111] P. Sideris, M. Nikoukalam, and M Salehi, “A generalized normal flow method with online step controls for pushover analysis of nonlinear softening structures,” *Engineering Structures*, vol. 143, pp. 232–244, 2017.
- [112] R. Groh, D Avitabile, and A Pirrera, “Generalised path-following for well-behaved nonlinear structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 394–426, 2018.
- [113] S. Chakraborty, S. Goswami, and T. Rabczuk, “A surrogate assisted adaptive framework for robust topology optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 346, pp. 63–84, 2019.
- [114] H. Tari and M. J. Dapino, “Globally convergent nonlinear 3d inverse model for smart materials with hessian-based optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 318, pp. 864–881, 2017.
- [115] M. Yu and Z. Wang, “Homotopy continuation of the high-order flux reconstruction/correction procedure via reconstruction (fr/cpr) method for steady flow simulation,” *Computers & Fluids*, vol. 131, pp. 16–28, 2016.
- [116] D. A. Brown and D. W. Zingg, “A monolithic homotopy continuation algorithm with application to computational fluid dynamics,” *Journal of Computational Physics*, vol. 321, pp. 55–75, 2016.
- [117] R. N. Borkovsky, U. Doraszelski, and Y. Kryukov, “A user’s guide to solving dynamic stochastic games using the homotopy method,” *Operations Research*, vol. 58, no. 4-part-2, pp. 1116–1132, 2010.
- [118] D. Besanko, U. Doraszelski, Y. Kryukov, and M. Satterthwaite, “Learning-by-doing, organizational forgetting, and industry dynamics,” *Econometrica*, vol. 78, no. 2, pp. 453–508, 2010.
- [119] P. J.-J. Herings and R. Peeters, “Homotopy methods to compute equilibria in game theory,” *Economic Theory*, vol. 42, no. 1, pp. 119–156, 2010.
- [120] M. Bartoň and V. M. Calo, “Optimal quadrature rules for odd-degree spline spaces and their application to tensor-product-based isogeometric analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 305, pp. 217–240, 2016.
- [121] P. J. Barendrecht, M. Bartoň, and J. Kosinka, “Efficient quadrature rules for subdivision surfaces in isogeometric analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 340, pp. 1–23, 2018.
- [122] R. Seydel and V. Hlavacek, “Role of continuation in engineering analysis,” *Chemical engineering science*, vol. 42, no. 6, pp. 1281–1295, 1987.
- [123] A. Neuenhofer and F. C. Filippou, “Geometrically nonlinear flexibility-based frame finite element,” *Journal of Structural Engineering*, vol. 124, no. 6, pp. 704–711, 1998.
- [124] M. Schulz and F. C. Filippou, “Non-linear spatial timoshenko beam element with curvature interpolation,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 4, pp. 761–785, 2001.
- [125] A. Cardona and M. Geradin, *Flexible Multibody Dynamics: A Finite Element Approach*, English. UK: John Wiley and Sons, 2001.

- [126] D. Hodges, “Proper definition of curvature in nonlinear beam kinematics,” *American Institute of Aeronautics and Astronautics Journal*, vol. 22, no. 12, pp. 1825–1827, 1984.
- [127] D. Luenberger and Y. Yu, *Linear and Nonlinear Programming*, English. New York,NY: Springer, 2008.
- [128] K. Washizu, *Variational Methods in Elasticity and Plasticity*, English. Pergamon Press, 1975.
- [129] J. Simo and T. Hughes, *Computational Inelasticity*, English. New York,NY: Springer, 1998.
- [130] C. P. Andriotis, K. G. Papakonstantinou, and V. K. Koumousis, “Nonlinear programming hybrid beam-column element formulation for large-displacement elastic and inelastic analysis,” *Journal of Engineering Mechanics*, vol. 144, no. 10, p. 04018096, 2018.
- [131] G. Cowper, “The shear coefficient in timoshenko’s beam theory,” *Journal of Applied Mechanics*, vol. 33, no. 2, pp. 335–340, 1966.
- [132] S. Mazzoni, F. McKenna, M. Scott, and G. Fenves, “Opensees command language manual,” *Pacific Earthquake Engineering Research (PEER) Center*, vol. 264, 2006.
- [133] A. Neuenhofer and F. C. Filippou, “Evaluation of nonlinear frame finite-element models,” *Journal of Structural Engineering*, vol. 123, no. 7, pp. 958–966, 1997.
- [134] *ABAQUS/Standard User’s Manual*, English. United States: Dassault Systèmes Simulia Corp, 2019.
- [135] K. Bishopp and D. Drucker, “Large deflection of cantilever beams,” *Quarterly of Applied Mathematics*, vol. 3, no. 3, pp. 272–275, 1945.
- [136] K. Mattiasson, “Numerical results from large deflection beam and frame problems analysed by means of elliptic integrals,” *International Journal for Numerical Methods in Engineering*, vol. 17, no. 1, pp. 145–153, 1981.
- [137] M. Batista, “A closed-form solution for Reissner planar finite-strain beam using Jacobi elliptic functions,” *International Journal of Solids and Structures*, vol. 87, pp. 153–166, 2016.
- [138] J. Reddy, “On locking-free shear deformable beam finite elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 149, no. 1-4, pp. 113–132, 1997.
- [139] G. Prathap and G. Bhashyam, “Reduced integration and the shear-flexible beam element,” *International Journal for Numerical Methods in Engineering*, vol. 18, no. 2, pp. 195–210, 1982.
- [140] O. Zienkiewicz, R. Taylor, and J. Too, “Reduced integration technique in general analysis of plates and shells,” *International Journal for Numerical Methods in Engineering*, vol. 3, no. 2, pp. 275–290, 1971.
- [141] S.-L. Lee, F. S. Manuel, and E. C. Rossow, “Large deflections and stability of elastic frames,” *Journal of Engineering Mechanics Division*, vol. 94, no. 2, pp. 521–548, 1968.
- [142] M. R. Horne, “Elastic-plastic failure loads of plane frames,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 274, no. 1358, pp. 343–364, 1963.
- [143] H. Santos, “Complementary-energy methods for geometrically non-linear structural models: An overview and recent developments in the analysis of frames,” *Archives of Computational Methods in Engineering*, vol. 18, no. 4, p. 405, 2011.
- [144] J. Simo and R. Taylor, “A return mapping algorithm for plane stress elastoplasticity,” *International Journal for Numerical Methods in Engineering*, vol. 22, no. 3, pp. 649–670, 1986.

- [145] J. C. Simo and T. J. Hughes, *Computational inelasticity*. Springer Science & Business Media, 2006, vol. 7.
- [146] M. L. Wilkins, “Calculation of elastic-plastic flow,” California Univ Livermore Radiation Lab, Tech. Rep., 1963.
- [147] F. De Angelis and R. L. Taylor, “An efficient return mapping algorithm for elastoplasticity with exact closed form solution of the local constitutive problem,” *Engineering Computations*, vol. 32, no. 8, 2015.
- [148] J. Clausen, L. Damkilde, and L. Andersen, “Efficient return algorithms for associated plasticity with multiple yield planes,” *International Journal for Numerical Methods in Engineering*, vol. 66, no. 6, pp. 1036–1059, 2006.
- [149] O. Hopperstad and S Remseth, “A return mapping algorithm for a class of cyclic plasticity models,” *International Journal for Numerical Methods in Engineering*, vol. 38, no. 4, pp. 549–564, 1995.
- [150] W. M. Scherzinger, “A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 526–553, 2017.
- [151] A. R. Hartloper, A. de Castro e Sousa, and D. G. Lignos, “Constitutive modeling of structural steels: Nonlinear isotropic/kinematic hardening material model and its calibration,” *Journal of Structural Engineering*, vol. 147, no. 4, p. 04021031, 2021.
- [152] J. C. Simo and R. L. Taylor, “Consistent tangent operators for rate-independent elastoplasticity,” *Computer methods in applied mechanics and engineering*, vol. 48, no. 1, pp. 101–118, 1985.
- [153] M Ortiz and E. P. Popov, “Accuracy and stability of integration algorithms for elastoplastic constitutive relations,” *International journal for numerical methods in engineering*, vol. 21, no. 9, pp. 1561–1576, 1985.
- [154] K. D. Hjelmstad and E. P. Popov, “Cyclic behavior and design of link beams,” *Journal of Structural Engineering*, vol. 109, no. 10, pp. 2387–2403, 1983.
- [155] J. Argyris, B Boni, U Hindenlang, and M Kleiber, “Finite element analysis of two-and three-dimensional elasto-plastic frames—the natural approach,” *Computer Methods in Applied Mechanics and Engineering*, vol. 35, no. 2, pp. 221–248, 1982.
- [156] R. A. Ridha and L. H. Lee, “Inelastic finite deformation of planar frames,” *Journal of the Engineering Mechanics Division*, vol. 97, no. 3, pp. 773–789, 1971.
- [157] M Amir, K. Papakonstantinou, and G. Warn, “Scaled spherical simplex filter and state-space damage-plasticity finite-element model for computationally efficient system identification,” *Journal of Engineering Mechanics*, vol. 148, no. 2, p. 04021138, 2022.
- [158] M. Kojima and R. Hirabayashi, “Continuous deformation of nonlinear programs,” in *Sensitivity, Stability and Parametric Analysis*, Springer, 1984, pp. 150–198.
- [159] H Gfrerer, H. Wacker, W Zulehner, and J Guddat, “Path-following methods for kuhn-tucker curves by an active index set strategy,” in *Systems and Optimization*, Springer, 1985, pp. 111–131.
- [160] J. Guddat, F. G. Vazquez, and H. T. Jongen, *Parametric optimization: singularities, pathfollowing and jumps*. Springer, 1990.
- [161] H. Gfrerer, J Guddat, *et al.*, “A globally convergent algorithm based on imbedding and parametric optimization,” *Computing*, vol. 30, no. 3, pp. 225–252, 1983.
- [162] W. I. Zangwill and C. Garcia, “Equilibrium programming: The path following approach and dynamics,” *Mathematical Programming*, vol. 21, no. 1, pp. 262–289, 1981.
- [163] A. B. Poore, “Bifurcations in parametric nonlinear programming,” *Annals of Operations Research*, vol. 27, no. 1, pp. 343–369, 1990.

- [164] H. T. Jongen and G.-W. Weber, “On parametric nonlinear programming,” *Annals of Operations Research*, vol. 27, no. 1, pp. 253–283, 1990.
- [165] C. Tiahrt and A. Poore, “Bifurcation and persistance of minima in nonlinear parametric programming,” in *Computation and Control*, Springer, 1989, pp. 343–353.
- [166] L. T. Watson, “Probability-one homotopies in computational science,” *Journal of Computational and Applied Mathematics*, vol. 140, no. 1-2, pp. 785–807, 2002.
- [167] S. Smale, “A convergent process of price adjustment and global newton methods,” *Journal of Mathematical Economics*, vol. 3, no. 2, pp. 107–120, 1976.
- [168] C. Garcia and F. Gould, “Relations between several path following algorithms and local and global newton methods,” *Siam Review*, vol. 22, no. 3, pp. 263–274, 1980.
- [169] A. C. Sun and W. D. Seider, “Homotopy-continuation method for stability analysis in the global minimization of the gibbs free energy,” *Fluid Phase Equilibria*, vol. 103, no. 2, pp. 213–249, 1995.
- [170] R. Menzel and H. Schwetlick, “Parametrization via secant length and application to path following,” *Numerische Mathematik*, vol. 47, no. 3, pp. 401–412, 1985.
- [171] M. Kuno and J. Seader, “Computing all real solutions to systems of nonlinear equations with a global fixed-point homotopy,” *Industrial & engineering chemistry research*, vol. 27, no. 7, pp. 1320–1329, 1988.
- [172] G. Powell and J. Simons, “Improved iteration strategy for nonlinear structures,” *International Journal for Numerical Methods in Engineering*, vol. 17, no. 10, pp. 1455–1467, 1981.
- [173] K.-J. Bathe and E. N. Dvorkin, “On the automatic solution of nonlinear finite element equations,” *Computers & Structures*, vol. 17, no. 5-6, pp. 871–879, 1983.
- [174] A. Klarbring, “On discrete and discretized non-linear elastic structures in unilateral contact (stability, uniqueness and variational principles),” *International journal of solids and structures*, vol. 24, no. 5, pp. 459–479, 1988.
- [175] G. Björkman, “Path following and critical points for contact problems,” *Computational Mechanics*, vol. 10, no. 3, pp. 231–246, 1992.
- [176] M. J. Clarke and G. J. Hancock, “A study of incremental-iterative strategies for non-linear analyses,” *International Journal for Numerical Methods in Engineering*, vol. 29, no. 7, pp. 1365–1391, 1990.
- [177] W. C. Rheinboldt, *An adaptive continuation process for solving systems of nonlinear equations*. SIAM, 1975.
- [178] L. T. Watson, “Numerical linear algebra aspects of globally convergent homotopy methods,” *SIAM review*, vol. 28, no. 4, pp. 529–545, 1986.
- [179] P. Deuflhard, B. Fiedler, and P. Kunkel, “Efficient numerical pathfollowing beyond critical points,” *SIAM journal on numerical analysis*, vol. 24, no. 4, pp. 912–927, 1987.
- [180] R. Seydel and V. Hlavacek, “Role of continuation in engineering analysis,” *Chemical engineering science*, vol. 42, no. 6, pp. 1281–1295, 1987.
- [181] A Šalgovič, V Hlaváček, and J Ilavský, “Global simulation of countercurrent separation processes via one-parameter imbedding techniques,” *Chemical Engineering Science*, vol. 36, no. 10, pp. 1599–1604, 1981.
- [182] H. Schwetlick and J. Cleve, “Higher order predictors and adaptive steplength control in path following algorithms,” *SIAM journal on numerical analysis*, vol. 24, no. 6, pp. 1382–1393, 1987.

- [183] J.-J. Gervais and H. Sadiky, “A continuation method based on a high order predictor and an adaptive steplength control,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, vol. 84, no. 8, pp. 551–563, 2004.
- [184] M. Syam, “Interpolation predictors over implicitly defined curves,” *Computers & Mathematics with Applications*, vol. 44, no. 8-9, pp. 1067–1076, 2002.
- [185] M. I. Syam, “Cubic spline interpolation predictors over implicitly defined curves,” *Journal of computational and applied mathematics*, vol. 157, no. 2, pp. 283–295, 2003.
- [186] L. T. Watson, S. C. Billups, and A. P. Morgan, “Algorithm 652: Hompack: A suite of codes for globally convergent homotopy algorithms,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 13, no. 3, pp. 281–310, 1987.
- [187] B. N. Lundberg and A. B. Poore, “Numerical continuation and singularity detection methods for parametric nonlinear programming,” *SIAM Journal on Optimization*, vol. 3, no. 1, pp. 134–154, 1993.
- [188] E. Allgower, C.-S. Chien, K Georg, and C.-F. Wang, “Conjugate gradient methods for continuation problems,” *Journal of Computational and Applied Mathematics*, vol. 38, no. 1-3, pp. 1–16, 1991.
- [189] H. B. Keller, “Lectures on numerical methods in bifurcation problems,” *Applied Mathematics*, vol. 217, p. 50, 1987.
- [190] B. N. Lundberg and A. B. Poore, “Variable order adams-bashforth predictors with an error-stepsize control for continuation methods,” *SIAM journal on scientific and statistical computing*, vol. 12, no. 3, pp. 695–723, 1991.
- [191] W. Mackens, “Numerical differentiation of implicitly defined space curves,” *Computing*, vol. 41, no. 3, pp. 237–260, 1989.
- [192] M. Kubíček, “Algorithm 502: Dependence of solution of nonlinear systems on a parameter [c5],” *ACM Transactions on Mathematical Software (TOMS)*, vol. 2, no. 1, pp. 98–107, 1976.
- [193] L. T. Watson, “A globally convergent algorithm for computing fixed points of c2 maps,” *Applied Mathematics and Computation*, vol. 5, no. 4, pp. 297–311, 1979.
- [194] K. Georg, “On tracing an implicitly defined curve by quasi-newton steps and calculating bifurcation by local perturbations,” *SIAM Journal on Scientific and Statistical Computing*, vol. 2, no. 1, pp. 35–50, 1981.
- [195] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, “Finding zeroes of maps: Homotopy methods that are constructive with probability one,” *Mathematics of Computation*, vol. 32, no. 143, pp. 887–899, 1978.
- [196] S. Ragon, Z. Gürdal, and L. T. Watson, “A comparison of three algorithms for tracing nonlinear equilibrium paths of structural systems,” *International journal of solids and structures*, vol. 39, no. 3, pp. 689–698, 2002.
- [197] W. C. Rheinboldt and J. V. Burkardt, “Algorithm 596: A program for a locally parameterized continuation process,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 9, no. 2, pp. 236–241, 1983.
- [198] E. Riks, “The application of newton’s method to the problem of elastic stability,” *ASME Journal of Applied Mechanics*, vol. 39, no. 4, pp. 1060–1065, 1972.
- [199] H. B. Keller, “The bordering algorithm and path following near singular points of higher nullity,” *SIAM journal on scientific and statistical computing*, vol. 4, no. 4, pp. 573–582, 1983.
- [200] T. F. Chan and H. Keller, “Arc-length continuation and multigrid techniques for nonlinear elliptic eigenvalue problems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 2, pp. 173–194, 1982.

- [201] A. Griewank, “On solving nonlinear equations with simple singularities or nearly singular solutions,” *SIAM review*, vol. 27, no. 4, pp. 537–563, 1985.

## VITA

### **Charilaos Lyritsakis**

Charilaos Lyritsakis was born in Heraklion of Crete, Greece, on March 16, 1990. He obtained his Engineer's Degree in Civil Engineering from National Technical University of Athens (N.T.U.A.), Greece, in 2015 with specialization in Structural and Earthquake Engineering. From 2015 to 2017 he enrolled in the interdisciplinary postgraduate programme "Analysis and Design of Earthquake Resistant Structures" in N.T.U.A and earned his Master of Science Degree in May of 2017. Since September of 2017, Charilaos Lyritsakis has been a Ph.D. student at the Pennsylvania State University in the Department of Civil and Environmental Engineering. His academic advisor was Dr. Kostas Papakonstantinou. His research work was supported by the U.S. National Science Foundation under grant No. 1634575 and focused on developing a hybrid beam finite element for general nonlinear analysis and incorporating it in a nonlinear optimization framework. Other topics explored during this collaboration and incorporated in the project were return mapping algorithms for rate-independent elastoplasticity, numerical continuation algorithms, global and parametric nonlinear programming frameworks.