Algorithms for Branch-Following and Critical Point Identification in the Presence of Symmetry

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Vincent Jusuf

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Ryan Scott Elliott, Adviser

April 2010

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my adviser, Professor Ryan S. Elliott. It is quite difficult to put into words the impact you have made on my academic interest and research ambitions. I deeply appreciate your insight, guidance and patience. It has truly been my honor to be able to work with you over the past years.

I thank my dissertation committee, Professor Ellad Tadmor and Professor Traian Dumitrica, for their insight and advice.

I thank my family for their love, encouragement, and support without which I would not be the person I am today.

I thank my beloved Catherine whose love and support have been invaluable to me and continue to bring me joy. You are my sunshine.

During these years at the University of Minnesota, I have been fortunate to work alongside talented colleagues whom I now call friends. I especially would like to thank my office mates Venkata Guthikonda and Nikhil Admal for sharing with me an immeasurable amount of knowledge. You have made long hours at the office more enjoyable. I would also like to acknowledge Amartya Banerjee, Shankar Krishnan, Amit Singh, Narina Jung, and Harris Ahmed Mohammed Ismail for their influence in shaping my research.

Last but not least, I thank my puppies Cosmo and Chili for giving me unconditional love and untold hours of joy and amusement.

To my loving family, Catherine, Cosmo and Chilli.

Abstract

The objective of this work is to present techniques that can be used to study steady state solutions to systems generically described by the equation $\mathbf{F}(\mathbf{x}, \lambda) = \frac{d\mathbf{x}}{dt} = \mathbf{0}$, where $\mathbf{x} \in \mathbb{R}^N$ is the set of degrees of freedom (DOFs) of the system and $\lambda \in \mathbb{R}$ is a loading parameter. In particular, the implications of the presence of discrete and continuous symmetry to systems described by $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$ will be explored. Additionally, numerical techniques that can be used to "path-follow" the evolution of the DOFs as the loading parameter λ is varied will be presented along with schemes to increase the computationally efficiency of the path-following algorithms. These techniques can be used to obtain solutions to $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$ around regular and singular points. Additionally, methods to detect, and compute singular points of $\mathbf{F}$ will be presented. Finally, the benefits of these techniques will be illustrated through the computation of solutions to two typical problems.

Contents

List of figures 1

Chapter 1 Introduction and outline **3**

1.1 Background of nonlinear differential equation problems 3

1.2 Symmetry 4

1.3 Branch-following methods 5

1.4 Critical points 6

1.5 Applications 7

Chapter 2 Invariances due to symmetry **9**

2.1 Discrete invariances 10

2.1.1 Restrictions due to symmetry 10

2.1.2 Example: Square symmetric spring system 12

2.2 Continuous invariances 15

2.2.1 Projection method 18

2.2.2 Phantom Energy method 19

2.2.3 Comments on the Projection and Phantom Energy methods 20

Chapter 3 Branch-following methods **22**

3.1 Predictor-corrector branch-following algorithms 23

3.2 Symmetry-breaking critical points 25
3.3 Symmetry-preserving critical points 28
3.3.1 Pseudo-Arclength Constraint Approach 30
3.3.2 Pseudo-Arclength Tangent Approach 31
3.4 Jacobian updating and steplength adaptation 35
3.4.1 Jacobian and QR updating 35
3.4.2 Steplength adaptation 39

Chapter 4 Critical point identification and calculation 44

4.1 Critical point detection and computation 45
4.1.1 Detecting zero eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ along the equilibrium curve $\mathbf{c}(s)$ 45
4.1.2 Preconditioning of the gradient 46
4.1.3 Computation of critical points 47
4.2 Classification of critical points 51

Chapter 5 Applications 55

5.1 A short rectangular column of FCC nickel under compression 55
5.1.1 Problem description 55
5.1.2 Principal equilibrium path 57
5.1.3 Performance of the Jacobian updating schemes 57
5.1.3.1 Comparison of results using the Projection method 61
5.1.3.2 Comparison of results using the Phantom Energy method 63
5.1.4 Benefit of using symmetry-reduction techniques 65
5.1.5 Summary of short rectangular column of FCC nickel under compression study 66

5.2 Equilibrium path-following of a one-dimensional crystal under uniaxial load 67

5.2.1 Problem description 67

5.2.2 Simulation results 68

5.2.3 A benefit of the Projection and Phantom Energy methods 69

5.3 Summary 70

Bibliography 72

Appendix A Symmetry of the square **76**

Appendix B One dimensional projection operator algorithm **81**

Appendix C Implicit function theorem **83**

List of Figures

1.1 Predictor step. 6

1.2 Corrector process. 6

1.3 Examples of (a) a bifurcation point and (b) a turning point along an equilibrium path $\mathbf{c}(\alpha)$. 8

2.1 Symmetry of the square. 10

2.2 A problem with square symmetry. All springs have stiffness "K". 13

2.3 A system with translational invariance of energy. 16

2.4 Partition of $\mathbf{R}^2$ into equivalence classes, and subspaces. 17

3.1 The existence of an equilibrium path of $\mathbf{F}$ by the IFT. 23

3.2 Starting from $\mathbf{c}(\alpha_0)$, a predictor step is taken onto $(\mathbf{x}^*, \lambda^*)$. 24

3.3 Starting from $(\mathbf{x}^*, \lambda^*)$, the corrector process iteratively solves for $\mathbf{c}(\alpha_1)$. . . 25

3.4 Branch following of $\mathbf{c}(\alpha), 0 \leq \alpha \leq \alpha_1$ with the path parameterized by λ. . . 26

3.5 Extension of curve beyond symmetry-breaking bifurcation points. 27

3.6 Extending curve beyond turning points. 29

3.7 General schematic of the Pseudo-Arclength Constraint Approach. 31

3.8 General schematic of Pseudo-Arclength Tangent Approach. 33

3.9 Steplength adaptation. 36

4.1 Preconditioning of the gradient. 46

4.2 Branch-following with PACA. 48

4.3 Branch-following with PATA. 48

4.4 A symmetric bifurcation point at $(\mathbf{x}_c, \lambda_c)$. 53

4.5 An asymmetric bifurcation point at $(\mathbf{x}_c, \lambda_c)$. 53

4.6 A turning point at $(\mathbf{x}_c, \lambda_c)$. 54

5.1 Rectangular column of FCC nickel under compression. 56

5.2 Principal equilibrium solution path of rectangular column of FCC nickel under compression. 58

5.3 Select configuration of rectangular column of FCC nickel along equilibrium path. 59

5.4 Select configuration of rectangular column of FCC nickel along equilibrium path. 59

5.5 Select configuration of rectangular column of FCC nickel along equilibrium path. 60

5.6 Select configuration of rectangular column of FCC nickel along equilibrium path. 60

5.7 One-dimensional bi-atomic crystal under uniaxial tension. 67

5.8 Principal path along with select bifurcating paths (*Dobson et al.* (2007)). . . 69

5.9 Eigenvalues of the stiffness matrix using conventional method. 70

5.10 Eigenvalues of the stiffness matrix using Phantom Energy method. 71

A.1 Elements of Square symmetry group. 76

Chapter 1

Introduction and outline

1.1 Background of nonlinear differential equation problems

Nonlinear differential equations are of special importance to physicists and mathematicians since most physical systems are nonlinear in nature. However, nonlinear systems are extremely difficult to solve and many nonlinear differential equations do not have explicit closed-form solutions. In fact, even seemingly simple systems can exhibit very complicated behavior. At this stage of history, only a few classes of solutions may be obtained analytically, such as those that exhibit either an equilibrium, periodic or quasi-periodic (combinations of several periodic terms with incommensurate frequencies) behavior (*Robinson* (2004)). One approach for exploring the behavior of nonlinear systems is to numerically calculate the evolution of the physical system in question, subject to a given initial condition. For example, E. Lorenz made an important contribution when he used a computer to study nonlinear equations motivated by the turbulence of motion of the atmosphere (*Lorenz* (1963)). Another example of numerical studies of nonlinear equations using computers is the Fermi-Pasta-Ulam (FPU) problem (*Fermi et al.* (1955)). The FPU problem is historically one of the first problems to be tackled using "numerical experimentation" and as a result stimulated the use of computers to study the behavior of complex systems that are analytically unsolvable. The study found that complicated, coupled, nonlinear systems of differential equations could exhibit almost exactly quasi-periodic behavior instead of the expected ergodic behavior. The seemingly paradoxical results obtained by Enrico Fermi, John R. Pasta and Stanislaw Marcin Ulam (also Mary Tsingou, the original coder of the problem) triggered a renewed interest in classical physics that eventually lead to the development of the field of dynamical systems and chaos theory. For more on the FPU problem,

refer to *Fermi et al.* (1955) and *Berman and Izrailev* (2005).

Due to the extreme complexities of most nonlinear differential systems, numerical computation methods have proven to be especially useful for providing insight into their behaviour. As such, it is important to develop analytical and numerical techniques that can efficiently and accurately explore any physical system of interest. The goal of this work is to study the equilibrium solutions of systems of differential equations of the following form:

$$\frac{d\mathbf{x}}{dt} = -\mathbf{F}(\mathbf{x}, \lambda), \tag{1.1}$$

or

$$\frac{d^2\mathbf{x}}{dt^2} = -\mathbf{F}(\mathbf{x}, \lambda), \tag{1.2}$$

where $\lambda \in \mathbb{R}$ is a scalar parameter and, $\mathbf{F}(\mathbf{x}, \lambda)$ is assumed to be obtained from a first derivative of a potential function $\mathcal{E}(\mathbf{x}, \lambda)$ such that $\mathbf{F}(\mathbf{x}, \lambda) \equiv \nabla_{\mathbf{x}}(\mathcal{E}(\mathbf{x}, \lambda))$. The equilibrium solutions to the above differential equations for a fixed $\lambda = \overset{o}{\lambda}$ are $\mathbf{x}(t; \overset{o}{\lambda}) = \overset{o}{\mathbf{x}}$, a constant, and satisfy

$$\mathbf{F}(\overset{o}{\mathbf{x}}, \overset{o}{\lambda}) = \mathbf{0}. \tag{1.3}$$

Continuous parameterized sets of solutions $\mathbf{c}(\alpha) \equiv (\overset{o}{\mathbf{x}}(\alpha), \overset{o}{\lambda}(\alpha))$ are called "equilibrium paths" or "equilibrium branches".

1.2 Symmetry

Often, the differential systems of interest possess certain invariances of the energy function that arise due to the presence of symmetry. The particular type of invariances are dependent on the type of symmetry present in the problem. In this work, two categories of symmetry will be considered: (1) "continuous symmetry" that gives rise to invariances of the form $\mathcal{E}(\mathbf{x} + \mathbf{T}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda)$, where $\mathbf{T} = \sum_{i=1}^{m} t_i \hat{\mathbf{T}}_i$, with $m < N$, $t_i \in \mathbb{R}$ and $\hat{\mathbf{T}}_i \in \mathbb{R}^N$ are unit vectors, and (2) "discrete symmetry" that gives rise to invariances of the form $\mathcal{E}(\mathbf{T}_g\mathbf{x}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda)$, $\forall g \in G$, where g is an element of the abstract "symmetry group" G and $\mathbf{T}_g \in \mathbb{R}^N \times \mathbb{R}^N$ is the linear operator on the "carrier space" $\mathbb{R}^N$ belonging to an appropriate representation of G. In cases such as these, it is possible to take advantage of the presence of symmetry by using techniques to eliminate a number of trivially satisfied equilibrium equations and thus obtain a restricted set of equilibrium equations, whose solutions are also

exact solutions to the full set of equilibrium equations. It is desirable to use the restricted set of equations to enable the use of the numerical techniques presented in this work as well as to lower the computational cost of performing branch following calculations. These symmetry techniques will be discussed further in Chapter 2.

1.3 Branch-following methods

Branch-following (or continuation) methods have long served as useful tools in modern mathematics. Their use can be traced back at least to such venerated works as those of Poincaré (*Poincaré* (1881-1886)), Klein (*Klein* (1883)) and Bernstein (*Bernstein* (1910)) (see *Allgower* (1980) for a more complete bibliography). Since the mid-twentieth century, researchers have performed numerical experiments with computers to examine multitudes of problems in a broad range of disciplines such as quantum mechanical atomistic simulations, continuum level simulations of machine parts, and problems in pure mathematics, finance, sociology, biology, and chemistry.

The objective of branch-following methods is to calculate the response or evolution of equilibrium solutions of systems such as Eq. (1.1) or Eq. (1.2) as a function of the parameter $\lambda \in \mathbb{R}$. The numerical branch-following techniques used in this work are based on the works of *Keller* (1987) and use a predictor-corrector method that takes advantage of the equations' smoothness. The main tool used to develop techniques for exploring equilibrium states of systems such as the one described above is the "implicit function theorem" (*Keller* (1987)). It states that the equilibrium equations implicitly define a locally unique continuous curve $\mathbf{c}(\alpha) \equiv (\mathbf{x}(\alpha), \lambda(\alpha))$ such that $\mathbf{F}(\mathbf{c}(\alpha)) \equiv \mathbf{F}(\mathbf{x}(\alpha), \lambda(\alpha)) = \mathbf{0}$, where α is analogous to an arclength or distance along the curve. The idea behind the continuation method is that given an initial solution point $(\mathbf{x}(\alpha_0), \lambda(\alpha_0)) \in \mathbf{c}(\alpha)$, a "predictor" step is taken to construct an approximation to the solution at $\alpha_1 = (\alpha_0 + d\alpha)$ (Fig. (1.1)). After obtaining the initial approximation, an iterative scheme is used to obtain a more accurate approximation to the solution; this is the "corrector" process (Fig. (1.2)). By keeping the predictor stepsize $d\alpha$ small, the iterative corrector process will generally converge since the predicted approximation point will be "close" to the solution point at α_1. Finally, once the corrector process has converged, the predictor-corrector process is repeated. In this way, a sequence of points $\{(\mathbf{x}(\alpha_i), \lambda(\alpha_i))\} \subset \mathbf{c}(\alpha)$ is generated that numerically traces the curve $\mathbf{c}(\alpha)$. A detailed description of the predictor-corrector branch-following algorithm will be presented in Chapter 3.

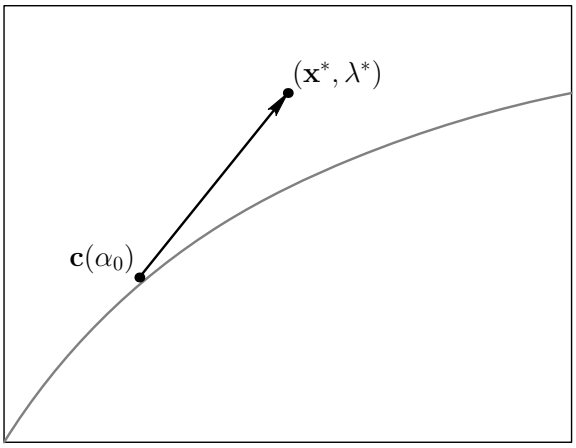

Figure 1.1: Predictor step.

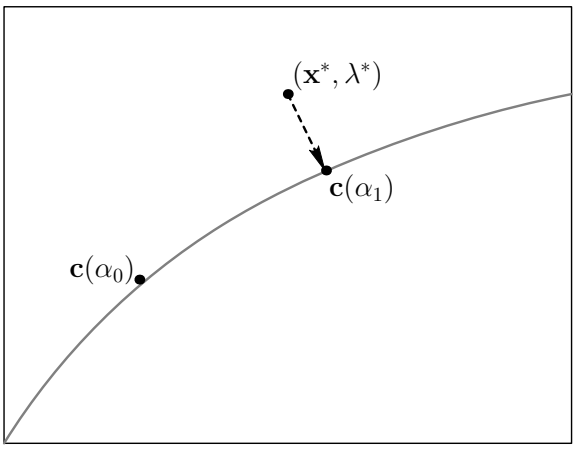

Figure 1.2: Corrector process.

1.4 Critical points

Generally, there will be certain points $(\mathbf{x}(\alpha_c), \lambda(\alpha_c)) \in \mathbf{c}(\alpha)$ with the property that the stiffness matrix $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(\alpha_c)) \equiv \mathbf{grad}_{\mathbf{x}}(\mathbf{F})\Big|_{\mathbf{c}(\alpha_c)}$ is singular (i.e, contains at least one zero eigenvalue). These points are called critical points of $\mathbf{F}$ and are, in this work, divided into two types: "symmetry breaking" and "symmetry preserving".

Symmetry breaking critical points are also called symmetry breaking bifurcation points and are equilibrium solution points where the singularity of the stiffness matrix occurs for the stiffness matrix of the full problem $\nabla_{\mathbf{x}}\mathbf{F}$ but the stiffness matrix of the reduced equations (introduced in Section 1.2) $\nabla_{\mathbf{x}_G}\mathbf{F}_G$ is non-singular. Moreover, bifurcation points are also characterized by the presence of multiple solution paths crossing through a single solution

point (see Fig. (1.3(a))). Because the stiffness matrix of the reduced set of equations does not become singular at these critical points, the branch-following algorithm can be coupled with a method like bisection (on α) in order to accurately calculate the critical point $(\mathbf{x}(\alpha_c), \lambda(\alpha_c))$. Once the critical point is accurately known, an asymptotic analysis can identify all bifurcating equilibrium paths (refer to *Triantafyllidis and Peek* (1992)) and *Elliott et al.* (2002). Furthermore, since multiple paths branching from a symmetry breaking bifurcation point will, by definition, have different symmetry characteristics, the continuation method can be restricted to generate solutions to each path of distinct symmetry by using the symmetry-reduced set of equations associated with the symmetry of each path. In this way, a method is available to selectively choose a path to follow based on its symmetry property. The techniques used to take advantage of the presence of symmetry will be presented in Chapter 2.

Symmetry preserving critical points are solution points where the stiffness matrix of both the full problem, $\nabla_{\mathbf{x}}\mathbf{F}$ and the symmetry-reduced problem (discussed previously in Section 1.2), $\nabla_{\mathbf{x}_G}\mathbf{F}_G$ are singular. Symmetry preserving critical points are further divided into "turning points" and "symmetry preserving bifurcation points". However, it can be shown that symmetry preserving bifurcation points are degenerate, or structurally unstable, in the sense that such a bifurcation point will be transformed into a turning point under any "generic" perturbation of the equilibrium equations (*Budiansky* (1974)). For this reason, symmetry-preserving bifurcation points will not be considered further in this work. Turning points, unlike bifurcation points, are critical points with a single unique equilibrium path that passes through the critical point (see Fig. (1.3(b))) and correspond to points of local extrema with respect to λ of the solution path $(\mathbf{x}(\alpha), \lambda(\alpha))$. Due to the singularity in $\nabla_{\mathbf{x}}\mathbf{F}$ and $\nabla_{\mathbf{x}_G}\mathbf{F}$ present at turning points, the equilibrium equations must be augmented by a constraint that regularizes the stiffness matrix so that a numerical continuation method can be used to accurately calculate the critical point $(\mathbf{x}(\alpha_c), \lambda(\alpha_c))$ and to explore the curve $\mathbf{c}(\alpha)$ past the critical point. Predictor-corrector continuation methods of this type will be discussed in Chapter 3, while a detailed discussion of the identification and calculation of critical points will be presented in Chapter 4.

1.5 Applications

After the details of the use of symmetry methods (Chapter 2), branch-following algorithms (Chapter 3), and critical point identification and calculation techniques (Chapter 4) have been presented, the use of these techniques to investigate problems of lattice-statics will be demonstrated in Chapter 5. In particular, the various methods will be used to calculate the

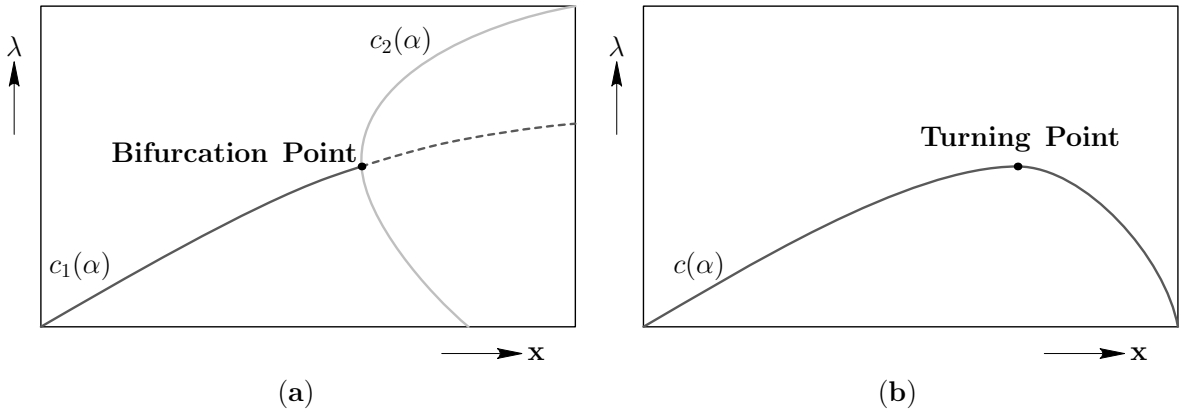

Figure 1.3: Examples of (a) a bifurcation point and (b) a turning point along an equilibrium path $\mathbf{c}(\alpha)$.

principal compressive deformation path of a plane-strain perfect crystal of a short column of nickel and the methods will then be compared again based on their performance and other strengths and weaknesses. Additionally, the methods will be used to investigate the behaviour of a one-dimensional crystal model of a phase-transforming material in tension. This problem will also be explored to study the benefits of symmetry techniques presented in Chapter 2.

Chapter 2

Invariances due to symmetry

The problems of interest in this work primarily concern systems that can be described using a potential function of the degrees of freedom and loading parameter $\mathcal{E}(\mathbf{x}, \lambda), (\mathbf{x}, \lambda) \in \mathbb{R}^N \times \mathbb{R}$. Specifically, the equation of interest is given by

$$\mathbf{F}(\mathbf{x}, \lambda) \equiv \nabla_{\mathbf{x}} \mathcal{E}(\mathbf{x}, \lambda) = \mathbf{0}. \tag{2.1}$$

In general, the set of solutions to Eq. (2.1) defines a one-dimensional equilibrium manifold (or equilibrium path) for the energy function. However, when continuous symmetries of the energy exist the equilibrium paths become equilibrium hyper-surfaces. Fortunately, these surfaces may be generated from a one-dimensional path by the continuous symmetry group's action, and thus, only a single representative path of distinct (in an appropriate sense) solutions is required. In this Chapter, methods will be presented for generating, directly from the full system of equations given by Eq. (2.1), a system of equations whose solution set is one such representative one-dimensional manifold.

Additionally, in the presence of discrete symmetry, the equilibrium equations can be reduced so that certain solutions can be obtained by solving a lower-dimensional set of equations. The idea behind this approach is that the equilibrium equations can be solved so that only solutions that preserve the symmetry of the original problem are found. In this way, symmetry related degrees of freedom that trivially satisfy the equilibrium equations can be identified, from which a lower-dimensional set of equations can then be obtained.

In this chapter, the consequences of symmetry for systems that can be described by a potential function are discussed.

2.1 Discrete invariances

In this work, discrete invariances are those invariances resulting from finite symmetry groups. These symmetries satisfy the following invariance equation:

$$\mathcal{E}(\mathbf{T}_g\mathbf{x}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda), \quad \forall g \in G, \tag{2.2}$$

where $\mathcal{E}$ is the energy of the system and $\mathbf{T}_g$ is the appropriate orthogonal matrix representation on the carrier space $\mathbb{R}^N$ of the element g belonging to the abstract finite symmetry group G of the system. For example, if G is the symmetry group of the square, then $G = \{E, c_4, c_2, c_4^{-1}, \sigma_{13}, \sigma_{24}, \sigma_{x_2}, \sigma_{x_1}\}$ where E is the identity element. Refer to Fig. (2.1) for the remaining symmetry operations of a square.

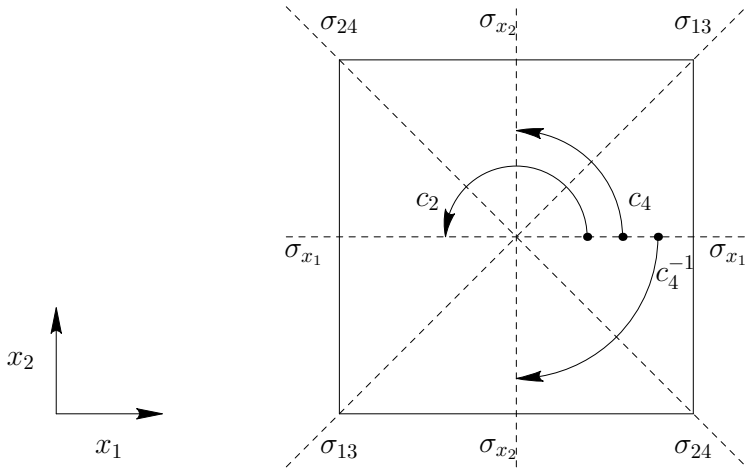

Figure 2.1: Symmetry of the square.

2.1.1 Restrictions due to symmetry

Here, we follow the general theory of linear group representations (*McWeeny* (2002)). Let $\mathbb{R}^N$ be a vector space, $\mathbf{x} \in \mathbb{R}^N$, and $\mathbf{T}_g$ a matrix representation of the element $g \in G$, where G is the finite symmetry group of order $|G|$ corresponding to the symmetry of the system. Observe that a consequence of Eq. (2.2) is the following condition obtained by taking the gradient with respond to $\mathbf{x}$ of Eq. (2.2):

$$\mathbf{F}(\mathbf{T}_g\mathbf{x}, \lambda) = \mathbf{T}_g\mathbf{F}(\mathbf{x}, \lambda), \quad \forall g \in G. \tag{2.3}$$

Equations that satisfy the above condition are said to be "equivariant" under the action of the symmetry group G.

Define S_G to be a subset of vectors in $\mathbb{R}^N$ as follows:

$$S_G \equiv \{\mathbf{x} \in \mathbb{R}^N \mid \mathbf{T}_g\mathbf{x} = \mathbf{x}, \forall g \in G\}. \tag{2.4}$$

It can be shown that S_G is a subspace of $\mathbb{R}^N$ and is referred to as the "G-symmetric subspace" of $\mathbb{R}^N$. Since S_G is a subspace of $\mathbb{R}^N$, one may find a set of $m < N$ vectors that form an orthonormal basis for S_G such that

$$S_G = \text{span}\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_m\}, \tag{2.5}$$

where m is given by

$$m = \frac{1}{|G|}\sum_{g\in G} \text{tr } \mathbf{T}_g. \tag{2.6}$$

See *McWeeny* (2002) for a derivation and proof of Eq. (2.6).

A projection operator $\mathbf{P}$ that projects any vector in $\mathbb{R}^N$ into S_G is defined in the following way:

$$\mathbf{P} = \frac{1}{|G|}\sum_{g\in G} \mathbf{T}_g. \tag{2.7}$$

Again, refer to *McWeeny* (2002) for derivation and proof of Eq. (2.7). By definition, for any vector $\boldsymbol{\psi} \in S_G$, it must be that

$$\mathbf{P}\boldsymbol{\psi} = \boldsymbol{\psi}. \tag{2.8}$$

Thus, the vectors $\boldsymbol{\psi}_i$ in Eq. (2.5) may be obtained by solving the eigenvalue problem in Eq. (2.8) for m eigenvectors with eigenvalue equal to one (*Healey* (1988)).

Equation (2.4) implies that for $\mathbf{x} \in S_G$ and for all $g \in G$

$$\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{F}(\mathbf{T}_g\mathbf{x}, \lambda) = \mathbf{T}_g\mathbf{F}(\mathbf{x}, \lambda). \tag{2.9}$$

This shows that $\mathbf{F}(\mathbf{x}, \lambda) \in S_G$ when $\mathbf{x} \in S_G$.

Equation (2.9) allows for the construction of the symmetry-reduced set of equations $\mathbf{F}_G(\mathbf{v}, \lambda)$

(the reduction of $\mathbf{F}(\mathbf{x}, \lambda)$ to $\mathbb{R}^m$ on the domain $\mathbb{R}^m \times \mathbb{R}$) in the following way:

$$\mathbf{F}_G(\mathbf{v}, \lambda) \equiv (\mathbf{\Psi}_G)^\top \mathbf{F}(\mathbf{\Psi}_G \mathbf{v}, \lambda) = \mathbf{0}, \tag{2.10}$$

where $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{\Psi}_G \equiv [\psi_1, \psi_2, ..., \psi_m]$, is an $N \times m$ matrix that maps $\mathbf{v}$ into S_G (and $(\mathbf{\Psi}_G)^\top$, the inverse map of $\mathbf{\Psi}_G$, maps vectors in S_G into $\mathbb{R}^m$).

An important result (*Healey* (1988)) is that $(\mathbf{v}, \lambda)$ satisfies $\mathbf{F}_G(\mathbf{v}, \lambda) = \mathbf{0}$ if and only if $\mathbf{F}(\mathbf{\Psi}_G \mathbf{v}, \lambda) = \mathbf{0}$ is satisfied. Thus, there is a one-to-one relationship between solutions $(\mathbf{v}, \lambda)$ of the reduced equations and the G-symmetric solutions of the full equations, $(\mathbf{x}, \lambda) = (\mathbf{\Psi}_G \mathbf{v}, \lambda)$.

Note that not all solutions to the full problem may be obtained from the reduced problem. Rather, the reduced problem captures only the G-symmetric solution points of the full problem; that is, only those solutions that preserve the symmetry of the original problem. For solutions that break the symmetry of the system, subgroups $\mathcal{H}$ of the full group can be used in order to construct the appropriate projection operator (and corresponding $\mathbf{\Psi}_\mathcal{H}$) associated with the symmetry of the desired solution.

2.1.2 Example: Square symmetric spring system

In this subsection, the benefits of exploiting discrete symmetry in problems is illustrated via an example. Consider the two-dimensional problem with square symmetry as shown (in the reference unstretched configuration) in Fig. (2.2). Let the six springs connecting nodes 1, 2, 3, and 4 have stiffness K. Furthermore, the structure is loaded at each node with in-plane loads of equal magnitude pointing towards the center of the square. That is, $\mathbf{\Lambda}_1 = [\lambda, \lambda]^\top$; $\mathbf{\Lambda}_2 = [\lambda, -\lambda]^\top$; $\mathbf{\Lambda}_3 = [-\lambda, -\lambda]^\top$; and $\mathbf{\Lambda}_4 = [-\lambda, \lambda]^\top$.

The state of the system can be described by the degree of freedom vector $\mathbf{u} \in \mathbb{R}^8$ where

$$\mathbf{u} = \left[u_{11}, u_{12}, u_{21}, u_{22}, u_{31}, u_{32}, u_{41}, u_{42}\right]^\top ,$$

and u_{ij} is the displacement of the i-th node in the x_j direction.

The total potential energy of the system is given by

$$\mathcal{E}(\mathbf{u}, \lambda) = \sum_{i=1}^{6} U_i + \sum_{j=1}^{4} \Omega_j, \tag{2.11}$$

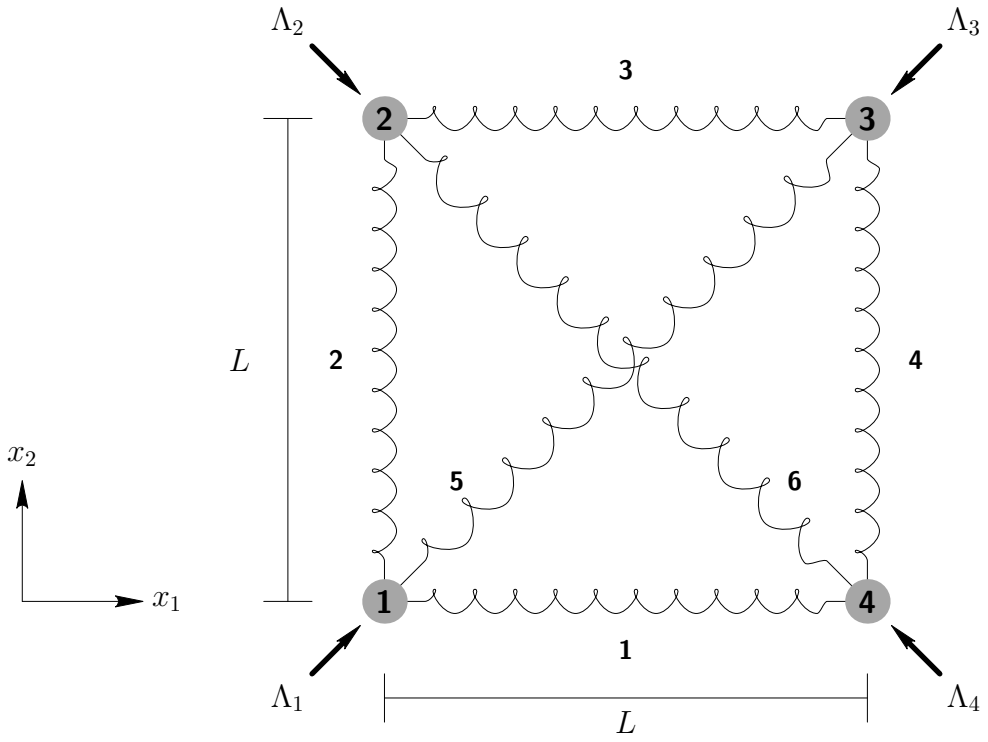

Figure 2.2: A problem with square symmetry. All springs have stiffness "K".

where U_i and Ω_j are the potential energy of the springs and the potential energy of the applied dead loads, respectively.

The potential energy of spring i is given by $U_i = \frac{1}{2}K(\ell_i - L_i)^2$ where ℓ_i and L_i are the deformed and original length of spring i, respectively. Therefore, the potential energy of each spring is given by

$$U_1 = \frac{1}{2}K\left(\sqrt{(L + u_{41} - u_{11})^2 + (u_{42} - u_{12})^2} - L\right)^2, \tag{2.12}$$

$$U_2 = \frac{1}{2}K\left(\sqrt{(u_{21} - u_{11})^2 + (L + u_{22} - u_{12})^2} - L\right)^2, \tag{2.13}$$

$$U_3 = \frac{1}{2}K\left(\sqrt{(L + u_{31} - u_{21})^2 + (u_{32} - u_{22})^2} - L\right)^2, \tag{2.14}$$

$$U_4 = \frac{1}{2}K\left(\sqrt{(u_{41} - u_{31})^2 + (u_{42} - u_{32} - L)^2} - L\right)^2, \tag{2.15}$$

$$U_5 = \frac{1}{2}K\left(\sqrt{(L + u_{31} - u_{11})^2 + (L + u_{32} - u_{12})^2} - \sqrt{2}L\right)^2, \tag{2.16}$$

$$U_6 = \frac{1}{2}K\left(\sqrt{(L + u_{41} - u_{21})^2 + (u_{42} - u_{22} - L)^2} - \sqrt{2}L\right)^2. \tag{2.17}$$

The potential energy of the applied load at node j is given by $\Omega_j = -\mathbf{\Lambda}_j \cdot \mathbf{u}_j$ and so the Ω_j

are given by

$$\Omega_1 = -\lambda(u_{11} + u_{12}), \tag{2.18}$$

$$\Omega_2 = -\lambda(u_{21} - u_{22}), \tag{2.19}$$

$$\Omega_3 = +\lambda(u_{31} + u_{32}), \tag{2.20}$$

$$\Omega_4 = -\lambda(-u_{41} + u_{42}). \tag{2.21}$$

The equilibrium equations are

$$\mathbf{F}(\mathbf{u}, \lambda) \equiv \nabla_{\mathbf{u}}(\mathcal{E}(\mathbf{u}, \lambda)) = \begin{bmatrix} \frac{\partial \mathcal{E}}{\partial u_{11}} \\ \frac{\partial \mathcal{E}}{\partial u_{12}} \\ \vdots \\ \frac{\partial \mathcal{E}}{\partial u_{41}} \\ \frac{\partial \mathcal{E}}{\partial u_{42}} \end{bmatrix} = \mathbf{0}. \tag{2.22}$$

Since the problem exhibits square symmetry, the above equation can be solved for solutions that preserve the symmetry of the system (G-symmetric solutions) by solving the reduced set of equations

$$\mathbf{F}_G(\mathbf{v}, \lambda) \equiv {\mathbf{\Psi}_G}^{\top} \mathbf{F}(\mathbf{\Psi}_G \mathbf{v}, \lambda) = \mathbf{0}. \tag{2.23}$$

Using Eq. (2.6), it can be shown that S_G is one-dimensional. Thus, $\mathbf{\Psi}_G$ consists of a single vector in $\mathbb{R}^8$ and is given by $\mathbf{\Psi}_G = \frac{1}{\sqrt{8}}[1, 1, 1, -1, -1, -1, -1, 1]^{\top}$ (refer to Appendix A for details). Accordingly, the equilibrium state vector $\mathbf{u}$ in the G-symmetric subspace S_G is given by

$$\mathbf{u} = \begin{bmatrix} u_{11} \\ u_{12} \\ u_{21} \\ u_{22} \\ u_{31} \\ u_{32} \\ u_{41} \\ u_{42} \end{bmatrix} = \frac{v}{\sqrt{8}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}. \tag{2.24}$$

Substituting Eq. (2.24) into Eq. (2.22) gives

$$\mathbf{F}(\mathbf{\Psi}_G v, \lambda) = \frac{\sqrt{2}Kv - \lambda}{\sqrt{8}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}. \tag{2.25}$$

The following reduced equation is then obtained:

$$F_G(v, \lambda) = \lambda - \sqrt{2}Kv = 0. \tag{2.26}$$

Solving the above reduced equation, which is a linear equation in one unknown, we obtain the solution $v = \frac{\lambda}{\sqrt{2}K}$ which relates the applied load to the nodal displacements (that preserve the square symmetry of the system). Individual nodal displacements can then be obtained by using the relation given in Eq. (2.24).

The above example illustrates the benefits of exploiting the presence of symmetry. While the full equilibrium equations can be used to obtain the nodal displacements as a function of the applied load, the number of equations to be solved is greatly reduced by taking advantage of symmetry. In this example, an eight-dimensional vector equation is reduced to a single equation that can be solved analytically.

2.2 Continuous invariances

In this work, continuous invariances are invariances resulting from the action of a continuous transformation group. Here, we consider continuous symmetry groups that are generated by a finite number of "infinitesimal transformations". An example of such a continuous transformation group is the isometry group of translations and rotations of three-dimensional Euclidean space. In this work, the implications of the invariance of a potential under the action of a continuous transformation group of translations will be explored. These continuous symmetries satisfy the following invariance equation:

$$\mathcal{E}(\mathbf{x} + \mathbf{T}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda). \tag{2.27}$$

where $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{T} = \sum_{i=1}^{m} t_i \hat{\mathbf{T}}_i$ with $t_i \in \mathbb{R}$, while $\hat{\mathbf{T}}_i \in \mathbb{R}^N$ are unit vectors and $m < N$. This energy invariance is schematically shown in Fig. (2.3).

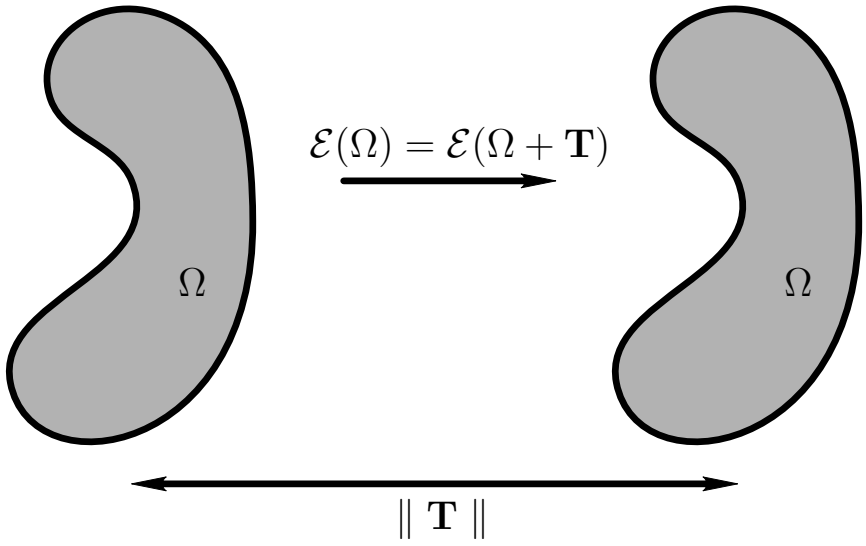

Figure 2.3: A system with translational invariance of energy.

Taking the gradient of Eq. (2.27) with respect to $\mathbf{x}$ gives

$$\mathbf{F}(\mathbf{x} + \mathbf{T}, \lambda) = \mathbf{F}(\mathbf{x}, \lambda). \tag{2.28}$$

Thus if $\overset{o}{\mathbf{x}}$ is an equilibrium configuration, the energy invariance of Eq. (2.27) leads to "trivial" equilibrium solutions of the form $\overset{o}{\mathbf{x}} + \mathbf{T}$. In fact, Eq. (2.27) suggests an equivalence relation between vectors in $\mathbb{R}^N$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, then $\mathbf{y}$ is equivalent to $\mathbf{x}$, denoted $\mathbf{y} \sim \mathbf{x}$, if $\mathbf{y} = \mathbf{x} + \mathbf{T}$ for some $\mathbf{T}$ in the "translation subspace" $\mathcal{T} \equiv \text{span}\{\widehat{\mathbf{T}}_1, \widehat{\mathbf{T}}_2, \ldots, \widehat{\mathbf{T}}_m\}$. Equivalently, $\mathbf{y} \sim \mathbf{x}$ if $(\mathbf{y} - \mathbf{x}) \in \mathcal{T}$. This equivalence relation partitions $\mathbb{R}^N$ into disjoint "equivalence classes" containing elements which are all equivalent. In Fig. (2.4), a schematic description of the partition of $\mathbb{R}^2$ into disjoint equivalence classes has been presented. In the figure, the translation vector $\widehat{\mathbf{T}}$ is shown with the solid gray line indicating the translation subspace $\mathcal{T}$ and the dashed gray lines indicating equivalence classes. The orthogonal complement $\mathcal{T}^\perp$ is a subspace of $\mathbb{R}^2$ perpendicular to $\mathbf{T}$ and is shown in the figure as the solid black line. Note that this particular partition is specific to the vector $\mathbf{T}$ shown in the figure.

Proposition 1. *A subspace of $\mathbb{R}^N$ can be identified such that it contains one and only one element from each equivalence class.*

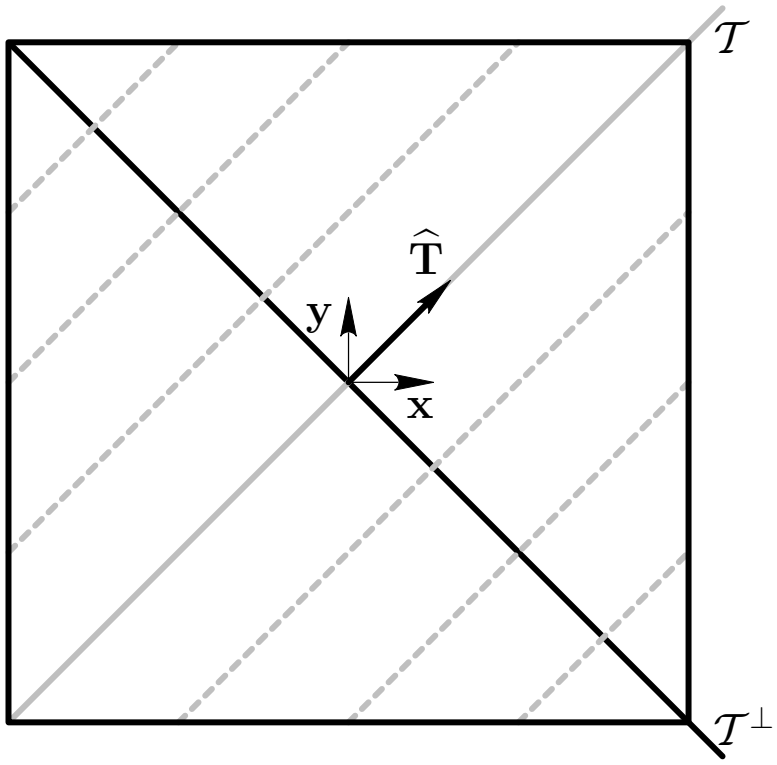

Figure 2.4: Partition of $\mathbb{R}^2$ into equivalence classes (dashed gray lines), translation subspace $\mathcal{T}$ (solid gray line) and translation orthogonal complement subspace $\mathcal{T}^{\perp}$ (solid black line).

Proof. Let $\mathcal{T}^{\perp}$ be the orthogonal complement of $\mathcal{T}$. That is, define a vector subspace of $\mathbb{R}^N$ as follows:

$$\mathcal{T}^{\perp} \equiv \{\mathbf{x} \in \mathbf{R}^N \mid \mathbf{x} \cdot \widehat{\mathbf{T}}_i = \mathbf{0}, i = 1, 2, \ldots, m\}. \tag{2.29}$$

Every vector $\mathbf{x} \in \mathbb{R}^N$ can be uniquely written as $\mathbf{x} = \mathbf{y} + \mathbf{T}$ where $\mathbf{y} \in \mathcal{T}^{\perp}$ and $\mathbf{T} \in \mathcal{T}$. Thus, it is clear that each vector of $\mathbb{R}^N$ is either in $\mathcal{T}^{\perp}$ (i.e, $\mathbf{T} = \mathbf{0}$) or equivalent to some vector in $\mathcal{T}^{\perp}$. Furthermore, no two elements of $\mathcal{T}^{\perp}$ are equivalent, since for $\mathbf{x}, \mathbf{y} \in \mathcal{T}^{\perp}$, $(\mathbf{x} - \mathbf{y}) \in \mathcal{T}^{\perp} \notin \mathcal{T}$. □

Only one configuration from each equivalence class needs to be considered in order to determine all equilibrium solutions for the system. This is because if $\mathbf{x}$ and $\mathbf{y}$ belong to the same equivalence class, then $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$ if and only if $\mathbf{F}(\mathbf{y}, \lambda) = \mathbf{0}$. Based on this fact, it is possible to construct a modified energy function whose set of equilibrium solutions is in $\mathcal{T}^{\perp}$. In this work, two techniques are developed that can be used to construct this modified energy function.

2.2.1 Projection method

This approach constructs the restriction of $\mathcal{E}$ to $\mathcal{T}^\perp$ in a manner similar to that used in Section 2.1.1. Let $\boldsymbol{\psi}_i$, $i = 1, 2, \ldots, N-m$, be an orthonormal basis for $\mathcal{T}^\perp$.
Then $\boldsymbol{\Psi}_{\mathcal{T}^\perp} \equiv [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_{N-m}]$ maps vectors in $\mathbb{R}^{N-m}$ to $\mathcal{T}^\perp$. The restricted energy function is defined as

$$\widetilde{\mathcal{E}}(\mathbf{v}, \lambda) \equiv \mathcal{E}(\boldsymbol{\Psi}_{\mathcal{T}^\perp}\mathbf{v}, \lambda), \tag{2.30}$$

where $\mathbf{v} \in \mathbb{R}^{N-m}$ and $\lambda \in \mathbb{R}$. The associated equilibrium equation is given by

$$\widetilde{\mathbf{F}}(\mathbf{v}, \lambda) = \boldsymbol{\Psi}_{\mathcal{T}^\perp}^\top \mathbf{F}(\boldsymbol{\Psi}_{\mathcal{T}^\perp}\mathbf{v}, \lambda) = \mathbf{0}. \tag{2.31}$$

An algorithm to generate a highly sparse (and therefore computationally efficient) $\boldsymbol{\Psi}_{\mathcal{T}^\perp}$ for the case of a one-dimensional translational subspace is given in Appendix B. This algorithm may be easily extended for use with higher-dimensional translation subspaces.

Claim 1. *Let $(\mathbf{v}, \lambda)$ satisfy Eq. (2.31). Then, $(\mathbf{x}, \lambda) = (\boldsymbol{\Psi}_{\mathcal{T}^\perp}\mathbf{v}, \lambda)$ is an equilibrium solution of the original problem. That is, $\widetilde{\mathbf{F}}(\mathbf{v}, \lambda) = \mathbf{0} \Rightarrow \mathbf{F}(\boldsymbol{\Psi}_{\mathcal{T}^\perp}\mathbf{v}, \lambda) = \mathbf{0}$. Furthermore, by definition, $\boldsymbol{\Psi}_{\mathcal{T}^\perp}\mathbf{v} \in \mathcal{T}^\perp$ and therefore has zero translation part.*

Proof. An expansion of the potential function is given by

$$\mathcal{E}(\mathbf{x} + d\mathbf{x}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda) + \mathbf{F}(\mathbf{x}, \lambda) \cdot d\mathbf{x} + \frac{1}{2}(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x}, \lambda)d\mathbf{x}) \cdot d\mathbf{x} + \ldots. \tag{2.32}$$

Let $d\mathbf{x} = \mathbf{t} \in \mathcal{T}$. Then by the invariance given in Eq. (2.27), Eq. (2.32) becomes

$$\mathcal{E}(\mathbf{x} + \mathbf{t}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda) + \mathbf{F}(\mathbf{x}, \lambda) \cdot \mathbf{t} + \cdots = \mathcal{E}(\mathbf{x}, \lambda), \tag{2.33}$$

which gives

$$\mathbf{F}(\mathbf{x}, \lambda) \cdot \mathbf{t} + \cdots = \mathbf{0}. \tag{2.34}$$

Since this must be satisfied for all $\mathbf{t} \in \mathcal{T}$, each term in Eq. (2.34) is identically zero. Therefore, it must be that $\mathbf{F}(\mathbf{x}, \lambda) \cdot \mathbf{t} = \mathbf{0}, \forall \mathbf{x} \in \mathbb{R}^N, \forall \mathbf{t} \in \mathcal{T}$, and consequently, $\mathbf{F}(\mathbf{x}, \lambda)$ has no component in $\mathcal{T}$. This implies that (using an obvious notation)

$$\boldsymbol{\Psi}_{\mathcal{T}}^\top \mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}, \forall \mathbf{x} \in \mathbb{R}^N. \tag{2.35}$$

Next, note that

$$\mathbf{I} = \mathbf{\Psi}_{\mathcal{T}}\mathbf{\Psi}_{\mathcal{T}}^{\top} + \mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{\Psi}_{\mathcal{T}^{\perp}}^{\top}, \tag{2.36}$$

then one immediately obtains

$$\mathbf{F}(\mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{v}, \lambda) = \mathbf{\Psi}_{\mathcal{T}}\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{F}(\mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{v}, \lambda) + \mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{\Psi}_{\mathcal{T}^{\perp}}^{\top}\mathbf{F}(\mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{v}, \lambda). \tag{2.37}$$

From Eq. (2.35), the first term is zero and from Eq. (2.31), the second term is $\mathbf{\Psi}_{\mathcal{T}^{\perp}}\widetilde{\mathbf{F}}(\mathbf{v}, \lambda)$ which is zero by assumption. Therefore, $\mathbf{F}(\mathbf{\Psi}_{\mathcal{T}^{\perp}}\mathbf{v}, \lambda) = \mathbf{0}$. □

2.2.2 Phantom Energy method

This method is inspired by the so-called penalty methods. The idea is to modify the original energy function in such a way as to break the continuous symmetry. Similar to the projection method discussed in the preceeding section, the resulting modified energy function should then have one solution for each equivalence class of the original problem. The method defines the following modified energy function:

$$\widetilde{\mathcal{E}}(\mathbf{x}, \lambda) \equiv \mathcal{E}(\mathbf{x}, \lambda) + \frac{1}{2}(\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x}) \cdot (\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x}), \tag{2.38}$$

where $\mathbf{\Psi}_{\mathcal{T}} : \mathbb{R}^m \mapsto \mathcal{T}$, is the projection operator that maps any vector in $\mathbb{R}^m$ to a vector in $\mathcal{T} \subset \mathbb{R}^N$. The second term in Eq. (2.38) is called the "Phantom Energy" and draws its name from the fact that its value is identically zero at all equilibrium solutions of the modified system. The equilibrium equations for this system are

$$\widetilde{\mathbf{F}}(\mathbf{x}, \lambda) = \mathbf{F}(\mathbf{x}, \lambda) + \mathbf{\Psi}_{\mathcal{T}}\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x} = \mathbf{0}. \tag{2.39}$$

Claim 2. $\widetilde{\mathbf{F}}(\mathbf{y}, \lambda) = \mathbf{0}$ *if and only if* $\mathbf{F}(\mathbf{y}, \lambda) = \mathbf{0}$ *and* $\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y} = \mathbf{0}$.

Proof. First, assume that $\widetilde{\mathbf{F}}(\mathbf{y}, \lambda) = \mathbf{0}$. Then, from Eq. (2.39)

$$\mathbf{F}(\mathbf{y}, \lambda) = -\mathbf{\Psi}_{\mathcal{T}}\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y}. \tag{2.40}$$

Multiplying both sides of Eq. (2.40) by $\mathbf{\Psi}_{\mathcal{T}}^{\top}$ gives (using Eq. (2.35))

$$\mathbf{0} = -\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{\Psi}_{\mathcal{T}}\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y}. \tag{2.41}$$

But $\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{\Psi}_{\mathcal{T}} = \mathbf{I}_{m\times m}$, the $m \times m$ identity. This gives

$$\mathbf{0} = \mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y}. \tag{2.42}$$

Thus, $\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y} = \mathbf{0}$, and by Eq. (2.40), $\mathbf{F}(\mathbf{y}, \lambda) = \mathbf{0}$. Therefore, $\widetilde{\mathbf{F}}(\mathbf{y}, \lambda) = \mathbf{0} \Rightarrow \mathbf{F}(\mathbf{y}, \lambda) = \mathbf{0}$ and $\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{y} = \mathbf{0}$. The converse is immediate. □

Thus, equilibrium solutions $(\mathbf{x}, \lambda)$ to Eq. (2.39) have zero translation part $(\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x} = \mathbf{0})$ and are therefore in $\mathcal{T}^{\perp}$. A useful property of the Phantom Energy function is that at equilibrium it has the same numerical value as the original energy. That is, for $(\mathbf{x}, \lambda)$ satisfying Eq. (2.39)

$$\widetilde{\mathcal{E}}(\mathbf{x}, \lambda) = \mathcal{E}(\mathbf{x}, \lambda) + \frac{1}{2}(\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x}) \cdot (\mathbf{\Psi}_{\mathcal{T}}^{\top}\mathbf{x}) = \mathcal{E}(\mathbf{x}, \lambda). \tag{2.43}$$

2.2.3 Comments on the Projection and Phantom Energy methods

Two methods for removing continuous translation invariance from problems have been discussed in this section. Both methods lead to the same set of equilibrium solutions, however there are significant differences between them. From the perspective of ease of computer implementation, the Phantom Energy method is clearly superior. The only additional tasks required are the construction and storage of $\mathbf{\Psi}_{\mathcal{T}}$, the addition of one simple term to the equilibrium equations, and a similarly simple additional term to $\nabla_{\mathbf{x}}\mathbf{F}$ when techniques such as Newton-Rhapson methods are used. In contrast, the Projection method requires a matrix multiplication of the equilibrium equations ($\widetilde{\mathbf{F}} = \mathbf{\Psi}_{\mathcal{T}^{\perp}}^{\top}\mathbf{F}$) and two such multiplications for $\nabla_{\mathbf{v}}\widetilde{\mathbf{F}} = \mathbf{\Psi}_{\mathcal{T}^{\perp}}^{\top}(\nabla_{\mathbf{x}}\mathbf{F})\mathbf{\Psi}_{\mathcal{T}^{\perp}}$.

These characteristics also have an effect on the computational efficiency of each method, with two competing factors to consider. First is the time required to construct the linear equations to be solved by the Newton-Rhapson procedure. The Phantom Energy method, requiring only a vector and matrix addition, significantly outperforms the Projection method in this task. The second factor is the time required to solve the resulting linear equations. In this case, while the Phantom Energy method results in a system of N equations, the Projection method leads to a system of $N - m$ equations. Thus generally, the projection method will have the best performance for this task.

Ultimately, the fastest method depends on the number N/m. When N/m is small (for example, $N/m < 10$), the Projection method is found to be more efficient. However when the dimension of the translation subspace m is small compared to N (i.e, $m \ll N$), then the

Phantom Energy method significantly outperforms the Projection method. See Chapter 5 for some examples of the use of both methods and a comparison of their efficiency.

Chapter 3

Branch-following methods

The development of large-memory, high-speed computers in the last quarter century has exerted a profound influence on various fields of science. Whereas classical analysis is overwhelmingly biased in favor of linear problems, computers are particularly suitable for solving systems of linear and by extension nonlinear differential equations. Of particular interest are the equilibrium solutions associated with these equations. This chapter introduces the concept of branch-following or continuation (*Ficken* (1951)) methods for computing the evolution of these equilibrium solutions as a parameter is varied, and presents a few different numerical branch-following schemes. The basic concept behind the numerical continuation methods used here deals with curves that can be parameterized with respect to a general parameter λ. These curves are generally described as solutions to nonlinear systems of the form

$$\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}. \tag{3.1}$$

Under sufficient smoothness conditions, and the absence of continuous invariances, the equation above implicitly defines a locally continuous curve or one-manifold of solution points $\mathbf{x}$ as a function of the parameter λ. Continuation methods obtain solutions along the curve by using an iterative solution method to generate a sequence of points $\{(\mathbf{x}_i, \lambda_i)\}$ that solve Eq. (3.1). That is, given an initial solution point $(\mathbf{x}_0, \lambda_0)$ such that $\mathbf{F}(\mathbf{x}_0, \lambda_0) = \mathbf{0}$ and for some small perturbation $d\lambda$, a point $(\mathbf{x}(\lambda_0 + d\lambda), \lambda_0 + d\lambda)$ such that $\mathbf{F}(\mathbf{x}(\lambda_0 + d\lambda), \lambda_0 + d\lambda) = \mathbf{0}$ is sought. Once this solution is obtained, the process is then repeated.

3.1 Predictor-corrector branch-following algorithms

Once continuous invariances have been eliminated from the equations of interest (as described in Chapter 2), it becomes possible to discuss the concept of solution paths. Suppose a solution point $(\mathbf{x}_0, \lambda_0)$ to Eq. (3.1) is known. In general, the matrix $\nabla_{\mathbf{x}}\mathbf{F}$ will be regular for the point $(\mathbf{x}_0, \lambda_0)$. In cases such as these, the implicit function theorem (IFT) states that there exists a locally continuous curve of solutions $\mathbf{c}(\alpha) \equiv (\mathbf{x}(\alpha), \lambda(\alpha))$ parameterized by $\alpha \in (\alpha_0, \alpha_1) \subset \mathbb{R}$ containing the origin such that $\mathbf{c}(0) \equiv (\mathbf{x}_0, \lambda_0)$ (*Pugh* (2000)). This continuous curve is schematically shown in Fig. (3.1), with the solid black line indicating the region of the equilibrium path where IFT is valid.

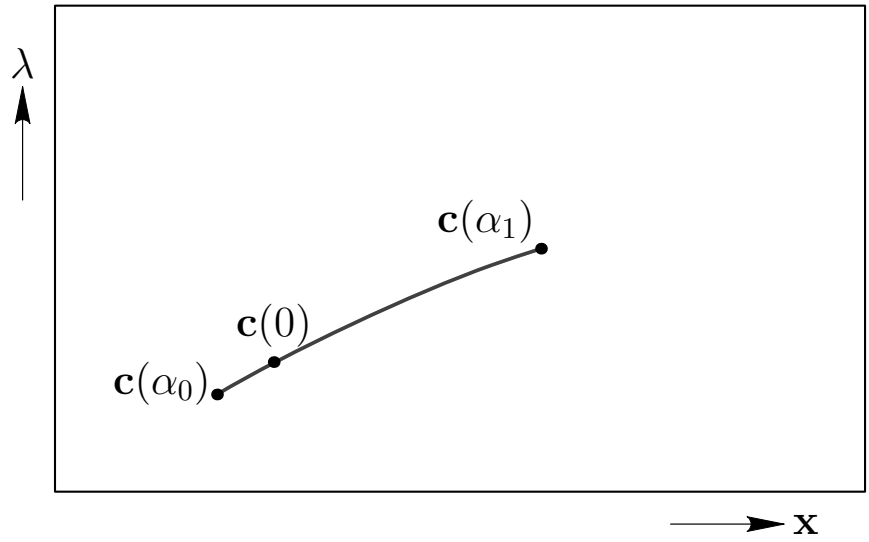

Figure 3.1: Solid line indicates the curve defined by $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$. Starting from the point $\mathbf{c}(0)$, the line indicates the region where the IFT is applicable and is therefore accessible using the continuation methods in this section.

Numerically, this path can be determined by identifying a sequence of N points along the path $\mathbf{c}(\alpha)$ using the algorithm given in Algorithm 1. Here the "Predictor" function obtains an approximation to the solution point $(\mathbf{x}_i, \lambda_i)$ given a previous point $(\mathbf{x}_{i-1}, \lambda_{i-1})$ (Fig. (3.2)) and the "Corrector" function iteratively updates the predicted point until $\| \mathbf{F}(\mathbf{y}, \xi) \| \leq \epsilon$, where ϵ is a given tolerance criterion (Fig. (3.3)).

The simplest approach is where the path is parameterized by the load λ. Figure (3.4) shows solid dots along the path indicating solution points computed by parameterizing the load. In this case, a commonly used predictor is the "secant predictor" where the solution $\mathbf{x}_{i+1}$ at λ_{i+1} is predicted by linearly interpolating from the solution at λ_{i-1} and λ_i. That is, the

Algorithm 1 General branch-following algorithm

Input: N, $d\alpha$, $(\mathbf{x}_0, \lambda_0)$, ϵ
for $i := 0$ to $N - 1$ **do**
  $(\mathbf{y}, \xi)$:= Predictor$(\mathbf{x}_i, \lambda_i; d\alpha)$; % Predict solution point $\mathbf{c}((i+1)d\alpha)$ from $\mathbf{c}(i\Delta\alpha) \equiv (\mathbf{x}_i, \lambda_i)$
  while $\| \mathbf{F}(\mathbf{y}, \xi) \| > \epsilon$ **do**
    $(\mathbf{y}, \xi) := (\mathbf{y}, \xi)+$ Corrector$(\mathbf{y}, \xi)$; % Improve (correct) the solution
  end while
  $(\mathbf{x}_{i+1}, \lambda_{i+1}) := (\mathbf{y}, \xi)$;
end for
Return $\{(\mathbf{x}_i, \lambda_i)\}$;

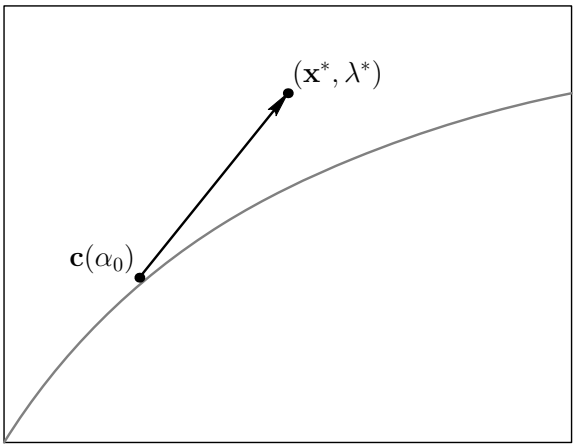

Figure 3.2: Starting from $\mathbf{c}(\alpha_0)$, a predictor step is taken onto $(\mathbf{x}^*, \lambda^*)$.

predicted point $\mathbf{y}(\lambda_{i+1})$ is given by

$$\mathbf{y} = \mathbf{x}_i + (\lambda_{i+1} - \lambda_i)\frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\lambda_i - \lambda_{i-1}}. \tag{3.2}$$

A common corrector method is the Newton-Rhapson (N-R) algorithm. An algorithm using these components is given in Algorithm 2.

Assuming an appropriately small value of $d\lambda$ is used, this algorithm will successfully be able to follow $\mathbf{c}(\alpha)$ to the point $\mathbf{c}(\alpha_1)$ starting from $\mathbf{c}(0)$ at which point the assumptions of the IFT first fail ($\nabla_{\mathbf{x}}\mathbf{F}$ is singular). Such points are called critical points and were categorized, in Chapter 1, as symmetry-breaking or symmetry-preserving. It turns out that in both cases, the curve $\mathbf{c}(\alpha)$ may be uniquely extended beyond $\mathbf{c}(\alpha_1)$. However, depending on the type of critical point, different techniques must be employed.

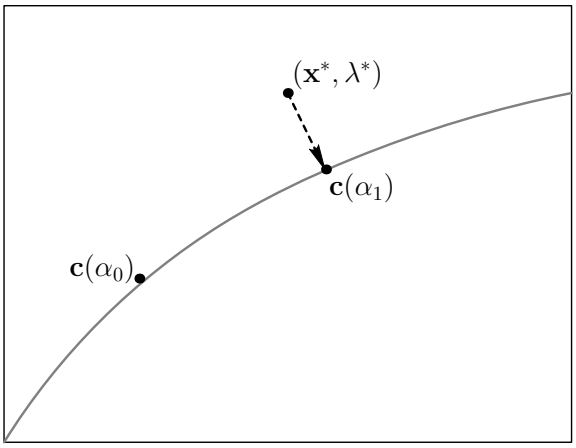

Figure 3.3: Starting from $(\mathbf{x}^*, \lambda^*)$, the corrector process iteratively solves for $\mathbf{c}(\alpha_1)$.

Algorithm 2 Path-following algorithm using Secant Predictor and N-R Corrector

Input: $N, d\lambda, (\mathbf{x}_0, \lambda_0), \epsilon$
for $i := 0$ to $N - 1$ **do**
  if $i = 0$ **then**
    $(\mathbf{y}, \xi) := (\mathbf{x}_0, \lambda_0)$; % Trivial Predictor
  else
    $(\mathbf{y}, \xi) := (\mathbf{x}_i + d\lambda \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\lambda_i - \lambda_{i-1}}, \lambda_i + d\lambda)$; % Secant Predictor
  end if
  while $\| \mathbf{F}(\mathbf{y}, \xi) \| > \epsilon$ **do**
    $(d\mathbf{y}, d\xi) := -[\nabla_{\mathbf{x}} \mathbf{F}(\mathbf{y}, \xi)]^{-1} \mathbf{F}(\mathbf{y}, \xi)$; % Newton-Rhapson Corrector
    $(\mathbf{y}, \xi) := (\mathbf{y}, \xi) + (d\mathbf{y}, d\xi)$;
  end while
  $(\mathbf{x}_{i+1}, \lambda_{i+1}) := (\mathbf{y}, \xi)$;
end for
Return $\{(\mathbf{x}_i, \lambda_i)\}$;

3.2 Symmetry-breaking critical points

The symmetry group of the critical point (also called the isotropy subgroup) at $\mathbf{c}^c \equiv (\mathbf{x}^c, \lambda^c) = (\mathbf{x}(\alpha_1), \lambda(\alpha_1))$ is defined as

$$G_{\mathbf{x}^c} = \{g \in G \mid \mathbf{T}_g \mathbf{x}^c = \mathbf{x}^c\} \subseteq G. \tag{3.3}$$

The symmetry reduced equilibrium equations near $\mathbf{c}^c$ are then $\mathbf{F}_{G_{\mathbf{x}^c}}(\mathbf{v}, \lambda) \equiv \mathbf{\Psi}_{G_{\mathbf{x}^c}}^{\top} \mathbf{F}(\mathbf{\Psi}_{G_{\mathbf{x}^c}} \mathbf{v}, \lambda)$. With this definition, the critical point is called a symmetry-breaking bifurcation point if

$$\det(\nabla_{\mathbf{v}} \mathbf{F}_{G_{\mathbf{x}^c}}(\mathbf{c}^c_{G_{\mathbf{x}^c}})) \neq \det(\nabla_{\mathbf{x}} \mathbf{F}(\mathbf{c}^c)) = 0, \tag{3.4}$$

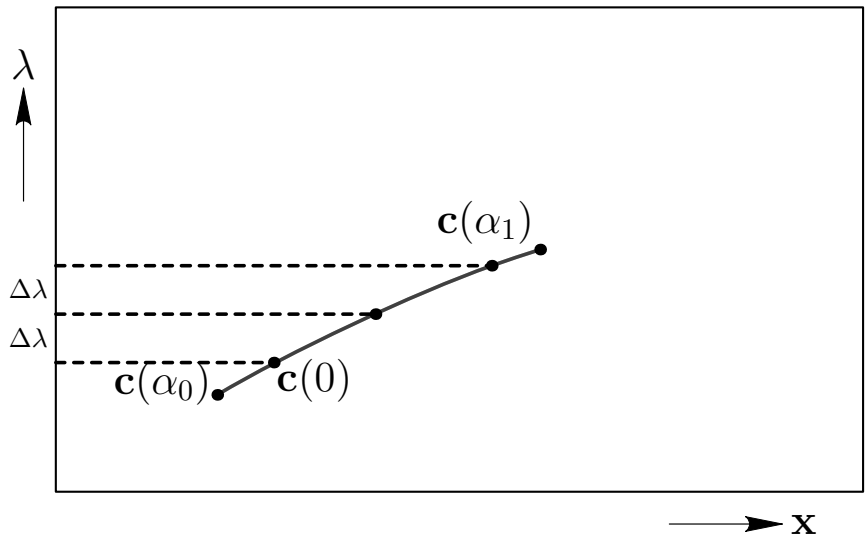

Figure 3.4: Branch following of $\mathbf{c}(\alpha), 0 \leq \alpha \leq \alpha_1$ with the path parameterized by λ.

where $\mathbf{c}^c_{G_{\mathbf{x}^c}} \equiv (\mathbf{\Psi}^\top_{G_{\mathbf{x}^c}} \mathbf{x}^c, \lambda^c) \in \mathbb{R}^m \times \mathbb{R}$ is the $G_{\mathbf{x}^c}$-symmetric solution point of $\mathbf{F}_{G_{\mathbf{x}^c}}$ associated with the critical point $\mathbf{c}^c$ of $\mathbf{F}$. A bifurcation point is a critical point where two or more solution paths intersect (i.e., one or more solution paths "bifurcates" from the "principal" solution path). A more detailed discussion of the different types of critical points including bifurcation points will be presented in Chapter 4.

There are two cases to be considered. First, suppose the initial solution $\mathbf{c}(0) \equiv (\mathbf{x}_0, \lambda_0)$ is such that $G_{\mathbf{x}_0} = G_{\mathbf{x}^c} \subseteq G$. That is, $\mathbf{x}_0$ has the same symmetry as the critical point. Then, the path $\mathbf{c}(\alpha)$ has the symmetry $G_{\mathbf{x}_0}$ and there is a one-to-one relationship between the zeros of $\mathbf{F}_{G_{\mathbf{x}_0}}$ and the $G_{\mathbf{x}_0}$-symmetric zeros of $\mathbf{F}$. In this case, the IFT can be applied to the set of equations $\mathbf{F}_{G_{\mathbf{x}_0}}$ to obtain a solution path $\mathbf{c}_{G_{\mathbf{x}_0}}(\alpha) \equiv (\mathbf{v}(\alpha), \lambda(\alpha)) \in \mathbb{R}^m \times \mathbb{R}$ for $\alpha \in (\alpha'_0, \alpha'_1)$. This path can then be mapped back to the $G_{\mathbf{x}_0}$-symmetric solution path of $\mathbf{F}$ as $\mathbf{c}(\alpha) \equiv (\mathbf{\Psi}_{G_{\mathbf{x}_0}} \mathbf{v}(\alpha), \lambda(\alpha))$ for $\alpha \in (\alpha'_0, \alpha'_1)$. By the definition of a symmetry-breaking bifurcation point, it is observed that α_1 must lie within (α'_0, α'_1). This shows that the solution path has been uniquely extended beyond the symmetry-breaking bifurcation point of $\mathbf{F}$. The solid black line in Fig. (3.5) shows the region of the path able to be computed by this method. Thus, to numerically follow solution paths through symmetry-breaking bifurcation points, it suffices to construct the symmetry reduced equations (as described in Chapter 2) $\mathbf{F}_{G_{\mathbf{x}_0}}(\mathbf{v}, \lambda)$ and use a general path-following algorithm, such as Algorithm 2, with initial condition $(\mathbf{v}_0, \lambda_0) = (\mathbf{\Psi}^\top_{G_{\mathbf{x}_0}} \mathbf{x}_0, \lambda_0)$ to obtain $\mathbf{c}_{G_{\mathbf{x}_0}}(\alpha)$. Once $\mathbf{c}_{G_{\mathbf{x}_0}}(\alpha)$ has been obtained, it can be mapped back to the desired solution path $\mathbf{c}(\alpha)$ in the usual way. A significant numerical advantage of this approach is that the symmetry-breaking bifurcation

point $(\mathbf{x}^c, \lambda^c)$ may be computed to a high numerical precision (see Chapter 4 for more details).

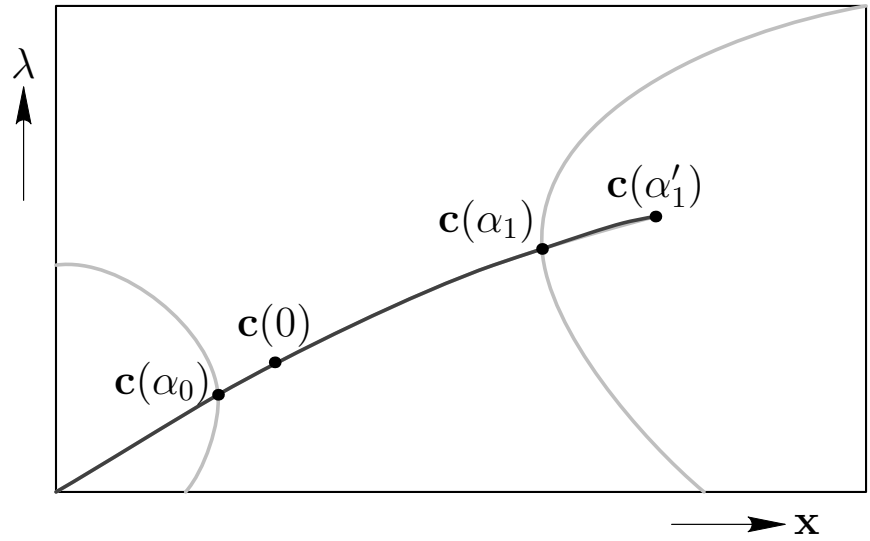

Figure 3.5: Extension of curve beyond symmetry-breaking bifurcation points.

It is useful to note that the above procedure will be able to follow the path to the point $\mathbf{c}(\alpha_1')$ at which stage, the assumptions of the IFT must fail for both $\mathbf{F}$ and $\mathbf{F}_{G_{\mathbf{x}_0}}$ (namely both $\nabla_{\mathbf{x}}\mathbf{F}$ and $\nabla_{\mathbf{v}}\mathbf{F}_{G_{\mathbf{x}_0}}$ are singular). Thus, by definition the point $\mathbf{c}(\alpha_1')$ must correspond to a symmetry-preserving critical point, which will be discussed in the next subsection, or a symmetry-breaking bifurcation point with $G_{\mathbf{x}_0} \subset G_{\mathbf{x}^c}$. The latter case will be discussed next.

In this case, the critical point has higher symmetry than the path along which it was discovered. It is assumed that $\mathbf{c}^c$ corresponds to a regular point of the $G_{\mathbf{x}^c}$-reduced equilibrium equations $\mathbf{F}_{G_{\mathbf{x}^c}}(\mathbf{v}, \lambda)$, so that $\mathbf{c}^c$ is a symmetry-breaking bifurcation point. Thus, the path being followed has symmetry group $\mathbf{G}_{\mathbf{x}_0}$ and corresponds to (one of) the path(s) bifurcating from the $G_{\mathbf{x}^c}$-symmetric solution path which crosses $\mathbf{c}^c$. Here, the situation is more delicate. In practice, Algorithm 2, detailed above (or the path-following methods to be introduced in later sections), is able to simply "step over" the bifurcation point and continue to follow the $\mathbf{c}_{\mathbf{G}_{\mathbf{x}_0}}(\alpha)$ path. However, sometimes these methods will "switch paths" at $\mathbf{c}^c$. In other words, the algorithm will follow $\mathbf{c}_{\mathbf{G}_{\mathbf{x}_0}}(\alpha)$ as it approaches $\mathbf{c}^c$ but near $\mathbf{c}^c$, the algorithm may converge to a point on $\mathbf{c}_{G_{\mathbf{x}^c}}(\alpha)$ after which it will path-follow $\mathbf{c}_{G_{\mathbf{x}^c}}(\alpha)$. Some techniques are available to ensure that this does not occur (see *Gatermann and Hohmann* (1991)) however, these techniques will not be discussed further in this work. In any case, if the symmetry-breaking bifurcation point $\mathbf{c}^c \equiv (\mathbf{x}^c, \lambda^c)$ is required to be known to high nu-

merical precision, it must be calculated using the $G_{\mathbf{x}^c}$ reduced equations $\mathbf{F}_{G_{\mathbf{x}^c}}$ as described in Chapter 2 and further in Chapter 4.

3.3 Symmetry-preserving critical points

A critical point is called "symmetry-preserving" if

$$\det(\nabla_{\mathbf{v}}\mathbf{F}_{G_{\mathbf{x}^c}}(\mathbf{c}^c_{G_{\mathbf{x}^c}})) = \det(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}^c)) = 0, \tag{3.5}$$

where, as in the previous section, $\mathbf{c}^c_{G_{\mathbf{x}^c}} \equiv (\mathbf{\Psi}^\top_{G_{\mathbf{x}^c}}\mathbf{x}^c, \lambda^c) \in \mathbb{R}^m \times \mathbb{R}$ is the $G_{\mathbf{x}^c}$-symmetric solution point of $\mathbf{F}_{G_{\mathbf{x}^c}}$ associated with the critical point $\mathbf{c}^c$ of $\mathbf{F}$. There are two types of symmetry-preserving critical points: "turning points" (also called "limit loads") and "symmetry-preserving bifurcation points". The latter, as discussed in Chapter 1, are "structurally unstable" in the sense that they are transformed into turning points by any "generic" perturbation of the equilibrium equations. Thus, the focus of this section will be on the development of a continuation technique capable of traversing turning-points encountered along a solution curve $\mathbf{c}(\alpha)$. Since a turning point singularity is exhibited by both the full equations $\mathbf{F} = \mathbf{0}$ and the reduced equations $\mathbf{F}_{G_{\mathbf{x}^c}} = \mathbf{0}$, the methods discussed below may be applied to either set of equations. For simplicity of notation and generality, the following discussion will use the full equations.

Turning points are symmetry-preserving critical points $\mathbf{c}(\alpha'_1)$ that, in addition to satisfying Eq. (3.5), satisfy

$$\nabla_{\mathbf{x}}\mathbf{F}\Big|_{\mathbf{c}(\alpha'_1)} \frac{d\mathbf{x}(\alpha'_1)}{d\alpha} = \mathbf{0}, \qquad \frac{\partial\mathbf{F}}{\partial\lambda}\Big|_{\mathbf{c}(\alpha'_1)} \cdot \frac{d\mathbf{x}(\alpha'_1)}{d\alpha} \neq 0, \quad \text{and} \quad \frac{d\lambda(\alpha'_1)}{d\alpha} = 0. \tag{3.6}$$

See Chapter 4 for a derivation of the above conditions. The motivation for the designation "turning point" is clear from the condition $\frac{d\lambda(\alpha'_1)}{d\alpha} = 0$. That is, the graph of $\mathbf{c}(\alpha)$ will have a local extremum in the path parameter λ. At such a point the path will "turn around" and reverse direction with respect to λ.

From the above, it is clear that the path $\mathbf{c}(\alpha)$ may not be parameterized by λ near $\mathbf{c}(\alpha'_1)$. However, any component of $\mathbf{x}$, say x_i, such that $\frac{dx_i(\alpha'_i)}{d\alpha} \neq 0$ can be used to parameterize the path. Choosing any such new parameter, it is straightforward to show that the reparameterized equilibrium equations satisfy the assumptions of the IFT at the turning point. Therefore, the curve may be extended uniquely beyond the turning point. In this way, by switching the local parameterization as needed and eliminating symmetry-breaking bifurcation points as described in Section 3.2, it has now been shown that the path may be ex-

tended indefinitely (schematically shown in Fig. (3.6)) as long as no symmetry-preserving bifurcation points exist.

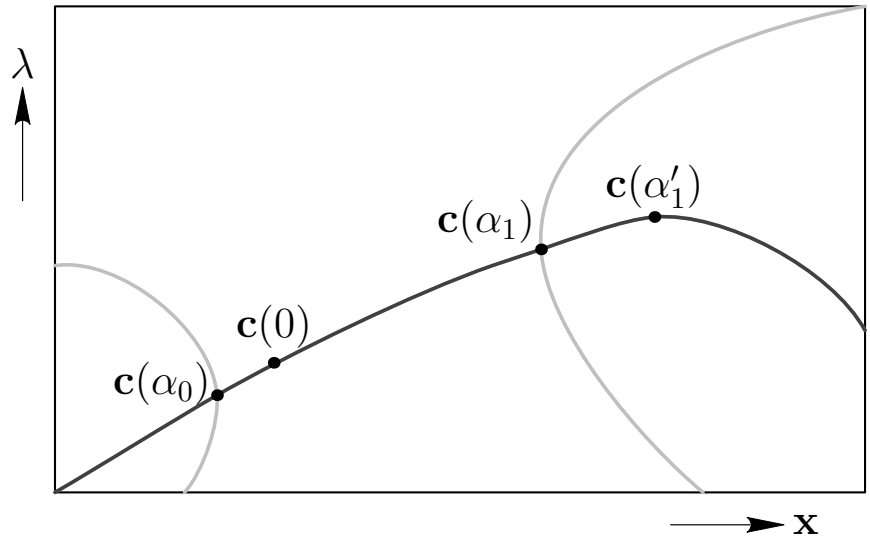

Figure 3.6: Extending curve beyond turning points.

It is advantageous at this point to reparameterize the constructed curve with the arclength parameter such that

$$ds = \left[\sum_{j=1}^{N}\left(\frac{dc_j(\alpha)}{d\alpha}\right)\right] d\alpha, \tag{3.7}$$

where s is now the arclength parameter and α represents the (local) parameter used to construct the curve. With this reparameterization and the above assumptions, it is possible to show (*Allgower* (1980)) that $\mathbf{c}(s)$ falls into one of only two categories:

1. $\mathbf{c}(s)$ is a closed curve such that there exists a number T with the property that $\mathbf{c}(s+T) = \mathbf{c}(s), \quad \forall s \in \mathbb{R}$, or
2. $\mathbf{c}(s)$ is injective (one-to-one) and has no accumulation points for $s \to \pm\infty$.

In other words, $\mathbf{c}(s)$ is either a bounded closed one-dimensional curve in $\mathbb{R}^{N+1}$ or it is an open unbounded one-dimensional curve in $\mathbb{R}^{N+1}$.

Now that $\mathbf{c}(s)$ is known to extend beyond $\mathbf{c}(s_1')$ it is possible to develop a path following algorithm that is capable of adaptively switching parameterizations as needed in order to avoid the difficulties associated with turning points. However, the implementation of such

an algorithm is a formidable task. Fortunately, better methods exist. Just as parameterizing the curve with the arclength is advantageous for mathematical proof, the idea has also led to a class of path following algorithms known as "pseudo-arclength" methods. These methods are able to numerically map out the path $\mathbf{c}(s)$ and to traverse turning points without complication or additional special consideration. In this thesis, two such methods will be discussed.

3.3.1 Pseudo-Arclength Constraint Approach

The Pseudo-Arclength Constraint Approach (PACA) approximates the increment of arclength along the path by the $(N+1)$-dimensional distance between two points on the curve as follows:

$$ds \approx \| \mathbf{c}(s_i + ds) - \mathbf{c}(s_i) \| . \tag{3.8}$$

Thus, given a solution point $\mathbf{c}(s_i)$ the algorithm finds a solution point $\mathbf{c}(s_{i+1})$ such that $\| \Delta\mathbf{c} \| = \| \mathbf{c}(s_{i+1}) - \mathbf{c}(s_i) \| = ds$, where ds is a given parameter, by augmenting the equilibrium equations $\mathbf{F} = \mathbf{0}$, with the following constraint equation:

$$\frac{1}{2}(\| \Delta\mathbf{c} \|^2 - ds^2) = 0. \tag{3.9}$$

The method also uses $\Delta\mathbf{c}$ as an approximation to the curve's tangent $\left(\frac{d\mathbf{c}}{ds}\right)$ for the purpose of the predictor step. For this reason, the method requires two solution points as initial input.

Based on these ideas, the general path-following algorithm of Section 3.1 is specialized as shown in Algorithm 3, where the notation $\mathbf{F}' \equiv \left[\nabla_{\mathbf{x}}\mathbf{F}, \frac{\partial \mathbf{F}}{\partial \lambda}\right]$ for the Jacobian of $\mathbf{F}$ is useful. See Fig. (3.7) for a pictorial description of the PACA.

In practice, this method is robust and reasonably efficient. Its biggest advantages are its conceptual simplicity and its ease of implementation. Disadvantages include the requirement of two solutions as input, the possibility (uncommon in practice) of converging to the solution point $\mathbf{c}_{i-1}$ for the $(i+1)$-th solution, and the extra computational expense and algorithmic complexity of performing the non-symmetric $(n+1) \times (n+1)$ linear solve. To a large extent, all of these disadvantages are eliminated by the Pseudo-Arclength Tangent Approach, which is discussed next.

Algorithm 3 Pseudo-Arclength Constraint Approach (PACA) path-following algorithm

Input: $N, ds, \epsilon, \mathbf{c}_0 = (\mathbf{x}_0, \lambda_0), \mathbf{c}_1 = (\mathbf{x}_1, \lambda_1)$
for $i := 1$ to $N - 2$ **do**
  $\Delta\mathbf{c} := \mathbf{c}_i - \mathbf{c}_{i-1}$;
  $\mathbf{d} := \mathbf{c}_i + \Delta\mathbf{c}$; % Prediction for % solution point $\mathbf{c}_{i+1}$
  while $\sqrt{\| \mathbf{F}(\mathbf{d}) \|^2 + \frac{1}{2}(\| \mathbf{d} - \mathbf{c}_i \|^2 - ds^2)} > \epsilon$ **do**
    $\mathbf{d} := \mathbf{d} - \begin{bmatrix} \mathbf{F}'(\mathbf{d}) \\ (\mathbf{d} - \mathbf{c}_i)^\top \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F}(\mathbf{d}) \\ \frac{1}{2}(\| \mathbf{d} - \mathbf{c}_i \|^2 - ds^2) \end{bmatrix}$; % Augmented N-R % Corrector
  end while
  $\mathbf{c}_{i+1} := \mathbf{d}$;
end for
Return $\{\mathbf{c}_i\}$;

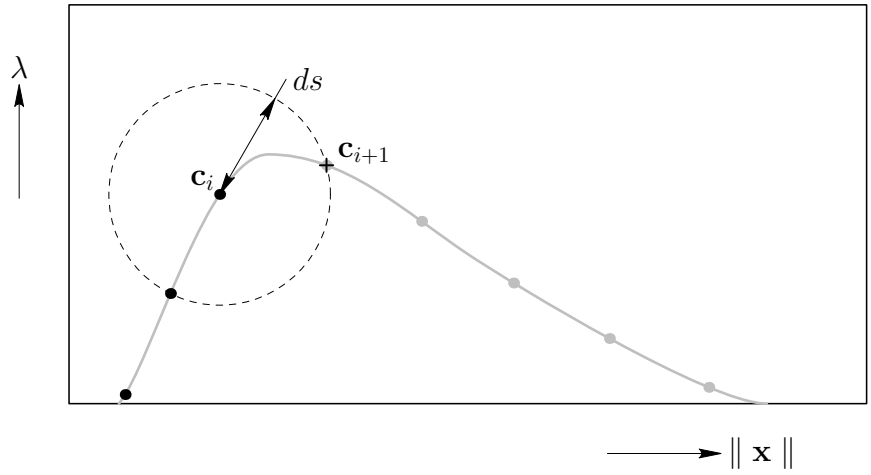

Figure 3.7: General schematic of the Pseudo-Arclength Constraint Approach. Starting from a known solution point $\mathbf{c}_i$, a solution point $\mathbf{c}_{i+1}$ is obtained such that $||\mathbf{c}_{i+1} - \mathbf{c}_i|| = ds$.

3.3.2 Pseudo-Arclength Tangent Approach

The Pseudo-Arclength Tangent Approach (PATA) approximates the increment of arclength along the path by the projection along the local path tangent $\frac{d\mathbf{c}(s_i)}{ds}$ as follows:

$$ds \approx \frac{d\mathbf{c}(s_i)}{ds} \cdot \Delta\mathbf{c}. \tag{3.10}$$

The path tangent can be conveniently obtained from the Jacobian of the equilibrium equations $\mathbf{F}' \in \mathbb{R}^N \times \mathbb{R}^{N+1}$. This may be shown by considering the total derivative of the

equilibrium equations $\mathbf{F} = \mathbf{0}$ along the path:

$$\frac{d}{ds}\mathbf{F}(\mathbf{c}(s)) = \mathbf{F}'(\mathbf{c}(s))\frac{d\mathbf{c}(s)}{ds} = \mathbf{0}. \tag{3.11}$$

Thus, one observes that the path tangent is a null-vector of the Jacobian. Except at bifurcation points, the Jacobian will be of maximal rank N which shows that the tangent is generally the unique (up to a sign) null-vector of the Jacobian. The sign convention of differential geometry for the positive direction along a curve is such that

$$\det \begin{bmatrix} \mathbf{F}' \\ \frac{d\mathbf{c}}{ds}^\top \end{bmatrix} > 0. \tag{3.12}$$

This convention will also be adopted in this work. It is convenient to define the following notation $\mathbf{t}(\mathbf{A}) \in \mathbb{R}^{N+1}$ for the tangent of a maximal rank operator $\mathbf{A} \in \mathbb{R}^N \times \mathbb{R}^{N+1}$. Thus, $\mathbf{t}(\mathbf{A})$ satisfies the following three properties:

1. $\mathbf{A}\mathbf{t}(\mathbf{A}) = \mathbf{0}$,
2. $\| \mathbf{t}(\mathbf{A}) \| = 1$,
3. $\det \begin{bmatrix} \mathbf{A} \\ \mathbf{t}(\mathbf{A})^\top \end{bmatrix} > 0.$

Given a solution point $\mathbf{c}(s_i)$, the PATA uses the tangent $\mathbf{t}(\mathbf{F}'(\mathbf{c}(s_i)))$ to generate a prediction $\mathbf{d}$ for the solution point $\mathbf{c}(s_i + ds)$. The method then aims to find the unique point $\mathbf{c}(\tilde{s})$ which minimizes the Euclidean distance from the point $\mathbf{d}$ to the solution curve. That is, $\mathbf{c}(\tilde{s})$ is taken as the point that satisfies:

$$\mathbf{c}(\widetilde{s}) = \min_{\mathbf{c}(s)} \{ \| \mathbf{c}(s) - \mathbf{d} \| \}. \tag{3.13}$$

When $\mathbf{F}$ is an affine function, the Moore-Penrose Inverse (*Campbell and Jr.* (1991)) of the Jacobian $\mathbf{F}'^{+}$ can be used to solve the minimization problem exactly giving:

$$\mathbf{c}(\tilde{s}) = \mathbf{d} - \mathbf{F}'^{+}\mathbf{F}(\mathbf{d}). \tag{3.14}$$

This is analogous to a Newton-Rhapson step for a square affine system. In the nonlinear case, an iterative approach is required. *Allgower* (1980) shows that under appropriate conditions this "under-determined Newton-Rhapson" iteration converges to a solution point $\mathbf{c}(s_{i+1})$ and that the distance between this point $\mathbf{c}(s_{i+1})$ and the exact solution $\mathbf{c}(\tilde{s})$ is of

order $\| \mathbf{d} - \mathbf{c}(\tilde{s}) \|^2$. Figure (3.8) is a schematic of the PATA. Based on these ideas, the general path-following algorithm of Section 3.1 is specialized as shown in Algorithm 4.

Algorithm 4 Pseudo-Arclength Tangent Approach (PATA) path-following algorithm

Input: $N, ds, \epsilon, \mathbf{c}_0 = (\mathbf{x}_0, \lambda_0)$
for $i := 0$ to $N - 1$ **do**
  $\mathbf{d} := \mathbf{c}_i + \mathbf{t}(\mathbf{F}'(\mathbf{c}_i))ds$; % Prediction for solution point $\mathbf{c}_{i+1}$
  while $\| \mathbf{F}(\mathbf{d}) \| > \epsilon$ **do**
    $\mathbf{d} := \mathbf{d} - \mathbf{F}'^{+}(\mathbf{d})\mathbf{F}(\mathbf{d})$; % under-determined N-R Corrector
  end while
  $\mathbf{c}_{i+1} := \mathbf{d}$;
end for
Return $\{\mathbf{c}_i\}$;

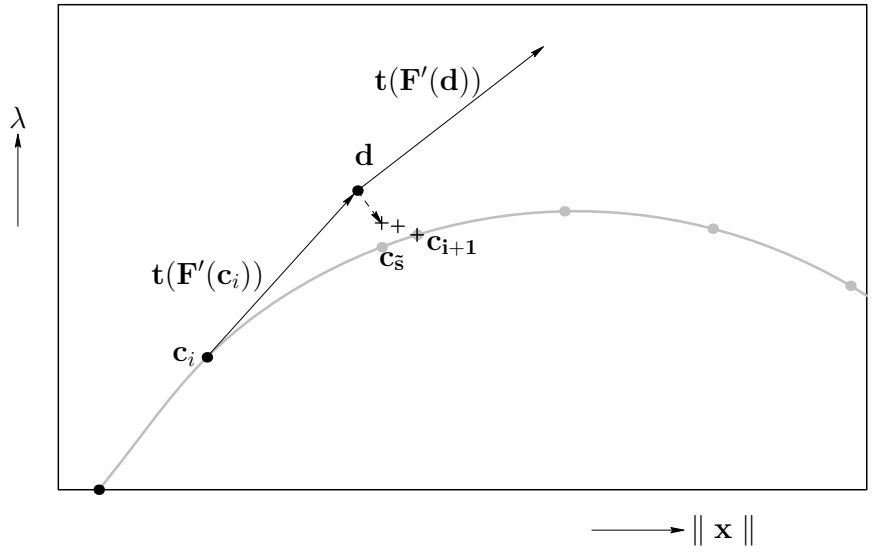

Figure 3.8: General schematic of Pseudo-Arclength Tangent Approach. Starting from a known solution point $\mathbf{c}_i$, the predicted point $\mathbf{d}$ is obtained by $\mathbf{d} = \mathbf{c}_i + \mathbf{t}(\mathbf{F}'(\mathbf{c}_i))ds$, where $\mathbf{t}(\mathbf{F}'(\mathbf{c}_i))$ is the tangent at $\mathbf{c}_i$. The corrector process then seeks the solution point $\mathbf{c}_{i+1}$ that is "closest" to $\mathbf{d}$.

A convenient and numerically stable method for obtaining the tangent vector and for calculating the Moore-Penrose (Pseudo) Inverse is to use the QR decomposition of the transpose

of the Jacobian, that is, $\mathbf{QR} = (\mathbf{F}')^\top$. Consider the definition of the tangent vector $\mathbf{t}$

$$\begin{aligned}
\mathbf{0} &= \mathbf{F}'\mathbf{t} \\
&= (\mathbf{QR})^\top \mathbf{t} \\
&= \left(\left[\widetilde{\mathbf{Q}}, \widetilde{\mathbf{q}} \right] \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}^\top \end{bmatrix} \right)^\top \mathbf{t} \\
&= \left([\widetilde{\mathbf{R}}^\top, \mathbf{0}] \begin{bmatrix} \widetilde{\mathbf{Q}}^\top \\ \widetilde{\mathbf{q}}^\top \end{bmatrix} \right) \mathbf{t},
\end{aligned}$$

where $\mathbf{Q} \in \mathbb{R}^{N+1} \times \mathbb{R}^{N+1}$ is an orthogonal matrix and $\widetilde{\mathbf{R}} \in \mathbb{R}^N \times \mathbb{R}^N$ is an upper triangular matrix. In general $\widetilde{\mathbf{R}}$ is of full rank, so by examining the above equation and recalling that the columns of $\mathbf{Q}$ are by definition mutually orthogonal, it is revealed that $\mathbf{t}$ must be identical (up to a sign) to $\widetilde{\mathbf{q}}$, the last column of $\mathbf{Q}$.
In order to determine the sign of $\mathbf{t}$, recall that the convention for a positive $\mathbf{t}$ is:

$$\det \begin{bmatrix} \mathbf{F}' \\ \mathbf{t}^\top \end{bmatrix} > 0. \tag{3.15}$$

Allgower (1980) shows that this implies that $\det(\mathbf{Q}) \det(\widetilde{\mathbf{R}}) > 0$.
The Pseudo-Inverse of $\mathbf{F}'$ may also be easily obtained from the QR decomposition $(\mathbf{F}')^\top$. By definition, the Pseudo-Inverse is given as:

$$\mathbf{F}'^{+} = \mathbf{F}'^{\top}(\mathbf{F}'\mathbf{F}'^{\top})^{-1}. \tag{3.16}$$

Substituting $\mathbf{F}'^{\top} = \mathbf{QR}$, into Eq. (3.16) gives:

$$\begin{aligned}
\mathbf{F}'^{+} &= \mathbf{QR}(\mathbf{R}^\top \mathbf{Q}^\top \mathbf{QR})^{-1} = \mathbf{QR}(\mathbf{R}^\top \mathbf{R})^{-1} \\
&= \mathbf{Q} \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}^\top \end{bmatrix} \left([\widetilde{\mathbf{R}}^\top, \mathbf{t}] \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}^\top \end{bmatrix} \right)^{-1} \\
&= \mathbf{Q} \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}^\top \end{bmatrix} \left(\widetilde{\mathbf{R}}^\top \widetilde{\mathbf{R}} \right)^{-1} \\
&= \mathbf{Q} \begin{bmatrix} \widetilde{\mathbf{R}} \\ \mathbf{0}^\top \end{bmatrix} \widetilde{\mathbf{R}}^{-1} \widetilde{\mathbf{R}}^{-\top} \\
&= \mathbf{Q} \begin{bmatrix} \widetilde{\mathbf{R}}^{-\top} \\ \mathbf{0}^\top \end{bmatrix}.
\end{aligned}$$

Due to the upper triangular form of $\widetilde{\mathbf{R}}$ the evaluation of $\Delta\mathbf{d} = -\mathbf{F}'^{+}\mathbf{F}(\mathbf{d})$ may be efficiently obtained by a two step process of (1) solving $\widetilde{\mathbf{R}}^{\top}\mathbf{y} = \mathbf{F}(\mathbf{d})$ for $\mathbf{y}$ by forward substitution, followed by (2) a straight forward matrix multiplication $\Delta\mathbf{d} = -\mathbf{Q}\begin{bmatrix}\mathbf{y}\\ \mathbf{0}^{\top}\end{bmatrix}$.

In cases where the computation of the Jacobian and/or the construction of its QR decomposition is regarded as computationally expensive, updating procedures to obtain good approximations of these quantities may be employed. These modifications to the PATA are discussed in the next section.

In practice, the PATA is robust and very efficient. Its advantages include its numerical stability and its computational efficiency. Few disadvantages exist and this fact makes the PATA the path following method of choice (for this author at least).

3.4 Jacobian updating and steplength adaptation

In general, the computation of $\mathbf{F}'$ is significantly more expensive than the computation of $\mathbf{F}$. Additionally, the quadratic convergence rate of the Newton-Rhapson method allows for good (at least linear) convergence when an approximation to $\mathbf{F}'$ is used instead of the exact value. For these reasons, it is advantageous to use an "updating scheme" for $\mathbf{F}'$ and/or its QR decomposition. Four such schemes are discussed in subsection 3.4.1.

Another computational efficiency issue is the choice of stepsize ds along the path $\mathbf{c}(s)$. If ds is too big, then fine scale features of the curve $\mathbf{c}(s)$ may be completely missed and the Newton-Rhapson method may have convergence difficulties. If ds is too small, a lot of computational effort is wasted by computing many solution points in regions of $\mathbf{c}(s)$ where the curve is well approximated by a straight line (and thus linear or cubic spline interpolation between two moderately separated solution points is sufficient). The solution to this problem is to adaptively change the stepsize ds based on certain local properties of the curve. This is schematically presented in Fig. (3.9), where the solid black dots indicate points along the solution path (gray line) that would be computed using a steplength adaptation technique. In subsection 3.4.2, three criteria are presented that can be used to set up an adaptive steplength mechanism.

3.4.1 Jacobian and QR updating

The PATA algorithm requires the Jacobian in order to obtain the tangent that is used in its prediction stage as well as the computation of the Jacobian for each iteration of the

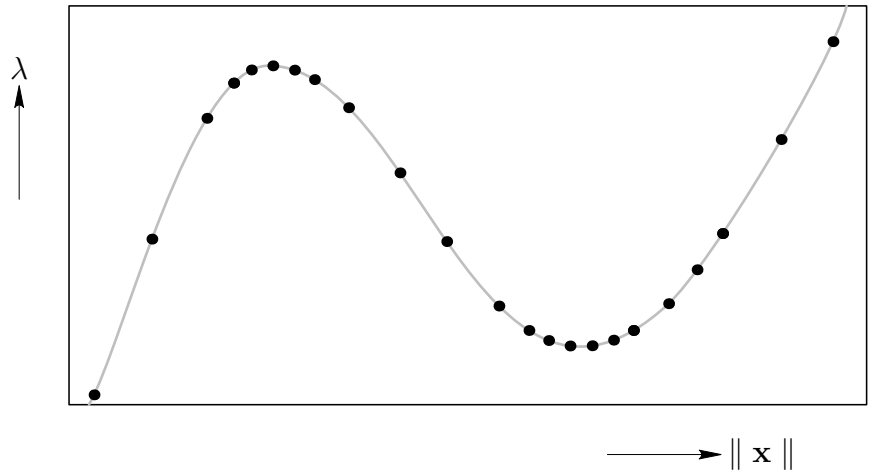

Figure 3.9: Schematic description of steplength adaptation. Solid black dots are points along the solution path (gray line) that are computed using a steplength adaptation technique.

Newton-Rhapson correction stage. Here, four Jacobian updating options are discussed (in order of increasing computational expense).

1. *No updating*

   This option computes the Jacobian once at the initial prediction point and repeatedly uses the same Jacobian value during the entire correction stage. In principle, due to the high computation cost of computing the exact Jacobian, this method would yield the most efficient scheme in all the updating routines. However, slower convergence might be encountered if the stepsize taken in the predictor step is too large.

2. *Broyden's good formula update of* $\mathbf{F}'$

   Broyden's good formula (see *Allgower* (1980) for a more detailed discussion) provides an update formula for the Jacobian based on the idea of least change of the Jacobian approximation and a first-order accurate approximation of the Jacobian action on the most current solution update $\Delta\mathbf{d}$.

   This update scheme produces a series of approximate Jacobians $\mathbf{J}_i$. The initial Jacobian $\mathbf{J}_0 = \mathbf{F}'(\mathbf{d}_0)$ is computed exactly in the prediction stage based on the predicted solution $\mathbf{d}_0$ and is used for the first iteration of the corrector stage. Once this iteration is complete, the following information is available: the residual $\mathbf{F}(\mathbf{d}_1)$ and the solution increment $\Delta\mathbf{d}_0 = -\mathbf{F}'^{+}(\mathbf{d}_0)\mathbf{F}(\mathbf{d}_0)$. Broyden's good formula then gives the

updated Jacobian approximation $\mathbf{J}_1$ as:

$$\mathbf{J}_1 = \mathbf{J}_0 + \frac{\mathbf{F}(\mathbf{d}_1) \otimes \Delta\mathbf{d}_0}{\| \Delta\mathbf{d}_0 \|^2}. \tag{3.17}$$

This formula is readily generalized to the i-th update as follows:

$$\mathbf{J}_i = \mathbf{J}_{i-1} + \frac{\mathbf{F}(\mathbf{d}_i) \otimes \Delta\mathbf{d}_{i-1}}{\| \Delta\mathbf{d}_{i-1} \|^2}. \tag{3.18}$$

Although the process to update the Jacobian matrix is more time consuming than the No Update method, Broyden's good updating routine often improves the convergence rate of the corrector method.

3. *Broyden's good formula update for* $\mathbf{Q}$ *and* $\mathbf{R}$

   The previous update scheme saves significant computation by using Broyden updates to approximate the Jacobian $\mathbf{F}'$. However, after each update, a QR decomposition must be performed in order to solve the Newton iteration using the Pseudo-Inverse. As such, another option to increase the performance of the predictor-corrector method is to update a QR factorization of the Jacobian. This update scheme performs the Broyden update directly on the QR decomposition and thus bypasses the more expensive two step procedure of first updating the Jacobian and then decomposing.

   Given the QR decomposition of the i-th Jacobian approximation $\mathbf{J}_i^\top = \mathbf{Q}_i\mathbf{R}_i$ and the required update information $\mathbf{F}(\mathbf{d}_{i+1})$ and $\Delta\mathbf{d}_i$, Algorithm 5 computes directly $\mathbf{Q}_{i+1}$ and $\mathbf{R}_{i+1}$ such that $\mathbf{Q}_{i+1}\mathbf{R}_{i+1} = \mathbf{J}_{i+1}^\top$. As described in detail by *Allgower* (1980), Algorithm 5 uses Givens Rotations in order to obtain $\mathbf{Q}_{i+1}$ and $\mathbf{R}_{i+1}$ where the function "QR" is given in Algorithm 6.

   The QR updating scheme detailed in Algorithm 5 has a computational cost of $O(N^3)$, where "N" is the number of equilibrium equations. However, by using the QR updating scheme, the computation of the Jacobian matrix during the corrector process is completely bypassed and is often found to be the most computationally efficient method studied in this work.

4. *Exact updating*

   Here, no approximation is used and the full exact Jacobian is computed each time it is required. This is obviously the most computationally expensive scheme of all the updating options but guarantees quadratic convergence of the corrector process.

Algorithm 5 QR updating

Input: $\mathbf{Q} \in \mathbb{R}^{N+1} \times \mathbb{R}^{N+1}, \mathbf{R} \in \mathbb{R}^{N+1} \times \mathbb{R}^{N}, \Delta\mathbf{d} \in \mathbb{R}^{N+1}, \mathbf{F} \in \mathbb{R}^{N}$.
$\Delta d :=\| \Delta\mathbf{d} \|$;
$\widehat{\mathbf{d}} := \mathbf{Q}^{\top}\Delta\mathbf{d}/\Delta d$;
$\widehat{\mathbf{f}} := \mathbf{F}/\Delta d$;
for $i := N-1$ to 0 **do**
  $C := \widehat{\mathbf{d}}[i]$;
  $S := \widehat{\mathbf{d}}[i+1]$;
  if $S \neq 0$ **then**
    $r := \sqrt{C^2 + S^2}$;
    $C := C/r$;
    $S := S/r$;
    $(\mathbf{Q}, \mathbf{R})$:= QR($\mathbf{Q}, \mathbf{R}, i, j, C, S$, A1, A2); % Performs Givens Rotations on % $\mathbf{Q}$ and $\mathbf{R}$ (see Algorithm 6)
    $A1 := C \cdot \widehat{d}[i] + S \cdot \widehat{d}[i+1]$
    $A2 := C \cdot \widehat{d}[i+1] - S \cdot \widehat{d}[i]$
    $\widehat{d}[i] := A1$
    $\widehat{d}[i+1] := A2$
  end if
end for
for $i := 0$ to $N-1$ **do**
  $R[0][i] := R[0][i] + \widehat{d}[0] \cdot \widehat{f}[i]$;
end for
for $i = 0$ to $N-1$ **do**
  $C := R[i][i]$;
  $S := R[i+1][i]$;
  if $(S \neq 0)$ **then**
    $r := \sqrt{C^2 + S^2}$;
    $C := C/r$;
    $S := S/r$;
    $(\mathbf{Q}, \mathbf{R})$:= QR($\mathbf{Q}, \mathbf{R}, i, j, C, S$, A1, A2); % Performs Givens Rotations on % $\mathbf{Q}$ and $\mathbf{R}$ (see Algorithm 6)
  end if
end for
Return $(\mathbf{Q}, \mathbf{R})$;

Algorithm 6 QR

Input: $\mathbf{Q}, \mathbf{R}, i, j, C, S$, A1, A2.
for $j := 0$ to $N - 1$ **do**
  $A1 := C \cdot R[i][j] + S \cdot R[i+1][j]$;
  $A2 := C \cdot R[i+1][j] - S \cdot R[i][j]$;
  $R[i][j] := A1$;
  $R[i+1][j] := A2$;
  $A1 := C \cdot Q[j][i] + S \cdot Q[j][i+1]$;
  $A2 := C \cdot Q[j][i+1] - S \cdot Q[j][i]$;
  $Q[j][i] := A1$;
  $Q[j][i+1] := A2$;
end for
Return $(\mathbf{Q}, \mathbf{R})$;

The first three update schemes require only two full Jacobian calculations (to obtain the initial tangent and also to obtain $\mathbf{F}'(\mathbf{d}_0)$) for each point $\mathbf{c}(s_i)$ obtained along the solution curve. The trade-off for this computational savings is an increase in the number of Newton iterations required for convergence. The last option requires a full Jacobian calculation for each iteration but generally reduces the number of iterations needed to converge onto a solution. Thus, the optimal scheme is dependent on the problem at hand as well as the values of ds and ϵ. In practice, the QR update scheme has been found to provide the best performance in many cases. See Chapter 5 for some specific examples of running times of each scheme.

3.4.2 Steplength adaptation

The steplength adaptation strategy described here aims to maintain a certain "performance level" for the corrector iteration by using a-posteriori estimates of three local properties along the solution curve. As described by *Allgower* (1980), these three properties are: (1) the magnitude of the Newton correction, (2) the Newton contraction rate, and (3) the angle between the tangent vector at the previous solution point and the tangent vector at the current solution approximation.

Let $\Delta\mathbf{d}_i$, $i = 1, 2, 3, \ldots$ be the Newton correction vector for the i-th iteration of the corrector step. Thus, $\mathbf{d}_i = \mathbf{d}_0 + \sum_{j=1}^{i} \Delta\mathbf{d}_j$ is the i-th iteration approximation to the solution, where $\mathbf{d}_0$ is the prediction for the solution obtained from the prediction step and $\Delta\mathbf{d}_{i+1} = -\mathbf{F}'^{+}(\mathbf{d}_i)\mathbf{F}(\mathbf{d}_i)$. Then, the first property is the magnitude of $\Delta\mathbf{d}_i$

$$\delta_i = \| \Delta\mathbf{d}_i \|, \tag{3.19}$$

which is considered a measure of the distance from $\mathbf{d}_{i-1}$ to the solution curve.

The second property, the Newton contraction rate, is simply the ratio of successive corrector iterations

$$\kappa_i = \frac{\delta_i}{\delta_{i-1}}, \tag{3.20}$$

for $i \geq 2$, which is a measure of how fast the Newton iteration is converging to the solution curve. It can be shown (*Allgower* (1980)) that under mild assumptions, these properties depend quadratically on the stepsize ds used during the prediction stage. The goal is to achieve nearly constant maximum values of δ_i and κ_i over all solutions along the curve. Thus, at the end of a predictor-corrector step, each property provides an a-posteriori estimate for its optimal stepsize $d\tilde{s}$ in terms of the actual stepsize ds used

$$d\tilde{s}_\delta = ds\sqrt{\frac{\tilde{\delta}}{\max_i(\delta_i)}}, \tag{3.21}$$

$$d\tilde{s}_\kappa = ds\sqrt{\frac{\tilde{\kappa}}{\max_i(\kappa_i)}}, \tag{3.22}$$

where $\tilde{\delta}$ and $\tilde{\kappa}$ are prescribed target values. The third property is the angle between the tangent vector at $\mathbf{d}_i$ and the tangent at the previous solution. If the previous solution is $\mathbf{c}_i$, then the angle is given by

$$\alpha_i = \cos^{-1}[\mathbf{t}(\mathbf{F}'(\mathbf{c}_i)) \cdot \mathbf{t}(\mathbf{F}'(\mathbf{d}_j))], \tag{3.23}$$

which is a measure of the curvature at $\mathbf{c}_{i+1}$ ($\mathbf{d}_j$ is the "j-th" corrector point). It can be shown that α depends linearly on ds, and thus its a-posteriori estimate for the optimal stepsize is

$$d\tilde{s}_\alpha = ds\frac{\tilde{\alpha}}{\max_i(\alpha_i)}, \tag{3.24}$$

where $\tilde{\alpha}$ is the prescribed target value for α. To set the new stepsize for finding the next solution point, the minimum of $d\tilde{s}_\delta$, $d\tilde{s}_\kappa$ and $d\tilde{s}_\alpha$ is selected.

It should be emphasized that the a-posteriori estimates above are based on the Euler predictor and Pseudo-Inverse Newton Corrector scheme. If other schemes (such as the updating schemes described in the previous subsection) are used, then these formulas may need to be altered.

A final version of the algorithm with Euler predictor and updated Pseudo-Inverse Newton corrector as well as the practical steplength adaptation implementation is presented in Algorithm 7. Convergence is based on the magnitude of the residual $\| \mathbf{F}(\mathbf{d}) \|$. However, in practice, one of three convergence criteria may be used: (1) $\| \mathbf{F}(\mathbf{d}) \| < \epsilon$, (2) $\| \Delta\mathbf{d} \| < \epsilon$, or (3) $\| \mathbf{F}(\mathbf{d}) \| < \epsilon$ and $\| \Delta\mathbf{d} \| < \epsilon$.

The algorithm implements a steplength adaptation scheme, in the spirit of that described in Section 3.4.2, that (1) ensures a maximum $ds_{\max}$ and minimum $ds_{\min}$ stepsize along the path, (2) ensures δ, κ, and α are always below specified maximum values $\delta_{\max}$, $\kappa_{\max}$, and $\alpha_{\max}$, respectively, and (3) ensures that the stepsize ratio between successive prediction steps, called the deceleration factor "decel", is bounded above and below by specified values, "$\text{decel}_{\max}$" and "$1/\text{decel}_{\max}$", respectively. Thus, the scheme performs prediction steps (with uniformly decreasing stepsize) until the corrector process converges without exceeding the maximum values for δ, κ, and α.

Algorithm 7 PATA with updating and steplength adaptation path-following algorithm

Input: $N, \epsilon, \mathbf{c}_0 = (\mathbf{x}_0, \lambda_0), ds, ds_{\min}, ds_{\max}, \delta_{\max}, \kappa_{\max}, \alpha_{\max}, \text{decel}_{\max}$,
Input: UpdateType % NoUpdate, JacobianUpdate, QRUpdate or ExactUpdate
for $i = 0$ to $N - 1$ **do**
  $(\mathbf{Q}, \mathbf{R})$:= GetQR($\mathbf{F}'^{\top}(\mathbf{c}_i)$); % Returns QR decomposition of $(\mathbf{F}'(\mathbf{c}_i))^{\top}$
  $\mathbf{t}_i$:= Tangent($\mathbf{Q}$); % Obtains tangent vector $\mathbf{t}(\mathbf{F}'(\mathbf{c}_i)^{\top})$
  Converged := false;
  while Converged = false **do**
    if $ds < ds_{min}$ **then**
      exit; % Quit if ds is too small
    end if
    decel := $1/\text{decel}_{\max}$; % Initial deceleration as small as allowed
    $\mathbf{d} := \mathbf{c}_i + ds \cdot \mathbf{t}_i$; % Euler predictor
    $(\mathbf{Q}, \mathbf{R})$:= GetQR($\mathbf{F}'(\mathbf{d})^{\top}$);
    $\mathbf{t}_{i+1}$:= Tangent($\mathbf{Q}$);
    $\alpha_0 := acos[\mathbf{t}_i \cdot \mathbf{t}_{i+1}]$;
    if $\alpha_0 > \alpha_{\max}$ **then**
      $ds := ds/\text{decel}_{\max}$; % Decrease ds by maximum amount allowed
      continue; % Start current predictor step over. Go to Line 35
    end if
    decel := $\max\left\{\text{decel}, \text{decel}_{\max}\frac{\alpha_0}{\alpha_{\max}}\right\}$; % $\frac{1}{\text{decel}_{\max}} < \text{decel} < \text{decel}_{\max}$ is ensured
    $\Delta\mathbf{d}_0$:= MoorePenroseSolve($\mathbf{Q}, \mathbf{R}, \mathbf{F}(\mathbf{d})$);
    $\mathbf{d} := \mathbf{d} + \Delta\mathbf{d}_0$; % Update $\mathbf{d}$
    if $||\Delta\mathbf{d}_0|| > \delta_{\max}$ **then**
      $ds := ds/\text{decel}_{\max}$;
      continue; % Start current predictor step over. Go to Line 35
    end if
    decel := $\max\left\{\text{decel}, \text{decel}_{\max}\sqrt{\frac{||\Delta\mathbf{d}_0||}{\delta_{\max}}}\right\}$;
    $j := 1$; % Loop counter
    if $||\mathbf{F}(\mathbf{d})|| < \epsilon$ **then**
      Converged := true;
    end if
    (Converged, ds, $\mathbf{d}$) := CorrectorIteration($\mathbf{Q}, \mathbf{R}, \Delta\mathbf{d}_j, \Delta\mathbf{d}_{j-1}, \mathbf{d}$, decel, $\text{decel}_{\max}, \delta_{\max}, ds, \kappa_j, \kappa_{\max}, \epsilon, j$, Converged, UpdateType); % See Algorithm 8
  end while
  $ds := ds/\text{decel}$; % New steplength
  if $ds > ds_{\max}$ **then**
    $ds := ds_{\max}$;
  end if
  $\mathbf{c}_{i+1} := \mathbf{d}$;
end for
Return $\{\mathbf{c}_i\}$;

Algorithm 8 CorrectorIteration

1: *Input:* $\mathbf{Q}, \mathbf{R}, \Delta\mathbf{d}_j, \Delta\mathbf{d}_{j-1}, \mathbf{d}$, decel, $\text{decel}_{\max}, \delta_{\max}, ds, \kappa_j, \kappa_{\max}, \epsilon, j$, Converged, UpdateType.
2: **while** Converged = false **do**
3: $(\mathbf{Q}, \mathbf{R}) :=$ GetUpdateQR$(\mathbf{Q}, \mathbf{R}, \mathbf{F}(\mathbf{d}), \Delta\mathbf{d}_{j-1}$, UpdateType);
4: $\Delta\mathbf{d}_j :=$ MoorePenroseSolve$(\mathbf{Q}, \mathbf{R}, \mathbf{F}(\mathbf{d}))$;
5: $\mathbf{d} := \mathbf{d} + \Delta\mathbf{d}_j$;
6: **if** $||\Delta\mathbf{d}_j|| > \delta_{\max}$ **then**
7: $ds := ds/\text{decel}_{\max}$;
8: break; %Go to Line 22
9: **end if**
10: $\text{decel} := \max\left\{\text{decel}, \text{decel}_{\max}\sqrt{\frac{||\Delta\mathbf{d}_j||}{\delta_{\max}}}\right\}$;
11: $\kappa_j := ||\Delta\mathbf{d}_j||/||\Delta\mathbf{d}_{j-1}||$;
12: **if** $\kappa_j > \kappa_{\max}$ **then**
13: $ds := ds/\text{decel}_{\max}$;
14: break; %Go to Line 22
15: **end if**
16: $\text{decel} := \max\left\{\text{decel}, \text{decel}_{\max}\sqrt{\frac{\kappa_j}{\kappa_{\max}}}\right\}$;
17: $j := j + 1$;
18: **if** $||\mathbf{F}(\mathbf{d})|| < \epsilon$ **then**
19: Converged := true;
20: **end if**
21: **end while**
22: Return (Converged, $ds, \mathbf{d}$);

Chapter 4

Critical point identification and calculation

In Chapter 3, it was shown that by taking advantage of the symmetry-reduced equations and using one of the two discussed pseudo-arclength branch-following methods, it is possible, in principle, to indefinitely trace out a solution curve of the equations $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$. These special techniques are required in order to numerically eliminate critical points along the curve that would otherwise result in an ill-conditioned gradient and poor convergence of the Newton-Rhapson algorithm. However, it is often desirable to accurately compute the critical values $(\mathbf{x}_c, \lambda_c)$.

This chapter will detail a method to identify the presence of critical points between each successive solution of the equation $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$ and a way to accurately compute those critical points. In addition, a criterion to distinguish between the two types of critical points considered in this work, bifurcation points and turning points, will be presented. First, the term "critical point" will be formally defined.

Definition: Let $\mathbf{F}(\mathbf{x}, \lambda) : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$ be a smooth map. A point $(\mathbf{x}_c, \lambda_c) \in \mathbb{R}^N \times \mathbb{R}$ is a critical point of $\mathbf{F}$ if the gradient $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x}_c, \lambda_c)$ satisfies the following condition:

$$\operatorname{rank}(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x}_c, \lambda_c)) < N. \tag{4.1}$$

This implies that $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x}_c, \lambda_c)$ is singular and contains at least one zero eigenvalue.

4.1 Critical point detection and computation

In this work, to identify the presence of critical points along a solution curve of $\mathbf{F}$, the eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ are obtained at each numerical solution point. Since a critical point is defined by the presence of at least one zero eigenvalue, a change in sign of one or more eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ implies the presence of at least one critical point between two successive solution points.

4.1.1 Detecting zero eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ along the equilibrium curve $\mathbf{c}(s)$

To detect the presence of critical points between two successive solution points $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$, consider the set of eigenvalues $\{\Lambda_1(s_i), \ldots, \Lambda_N(s_i)\}$ and $\{\Lambda_1(s_{i+1}), \ldots, \Lambda_N(s_{i+1})\}$ of $\nabla_{\mathbf{x}}\mathbf{F}$ at $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$, respectively. Since the eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ depend continuously on $\mathbf{c}(s)$ (*N.P Van Der AA and Mattheij* (2007)), at least one critical point exists for some $s \in [s_i, s_{i+1}]$ if

$$\Lambda_k(s_i) \cdot \Lambda_k(s_{i+1}) \leq 0, \tag{4.2}$$

for one or more $k \in [1, N]$. If the expression in Eq. (4.2) is identically zero for some k, then it must be that either $\mathbf{c}(s_i)$, $\mathbf{c}(s_{i+1})$, or both are critical points. However, In general it can be assumed that a strict inequality applies in Eq. (4.2). Furthermore, we will assume that the stepsize $ds = s_{i+1} - s_i$ is sufficiently small to ensure that no eigenvalues change sign twice in the interval $[s_i, s_{i+1}]$. In the generic case, since a symmetry-breaking critical point can result in multiple eigenvalues going to zero at the same time, the maximum number of critical points present between $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$ is the number of eigenvalues that have changed sign. That is, if $S = \{s \in [s_i, s_{i+1}] \mid \operatorname{rank}(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s))) < N\}$ and $K = \{k \in [1, N] \mid \Lambda_k(s_i) \cdot \Lambda_k(s_{i+1}) \leq 0\}$, then $|S| \leq |K|$.

One difficulty that occurs in the practical implementation of this algorithm for detecting zero eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ along $\mathbf{c}(s)$ is the need to ensure the smooth variation of the computed eigenvalues $\Lambda_k(s)$. Theoretically, smoothness is assured (*N.P Van Der AA and Mattheij* (2007)). However, most numerical eigenvalue computations are concerned only with finding eigenvalues of a single matrix and, therefore, provide no guarantees about the order in which the eigenvalues of two closely related matrices are obtained. This leads to the possibility of a naive implementation of Eq. (4.2) that may provide incorrect conclusions.

4.1.2 Preconditioning of the gradient

To ensure a consistent ordering (smoothness) of the eigenvalues, the gradient at the solution point $\mathbf{c}(s_{i+1})$ is preconditioned before it is passed to a numerical eigenvalue solver. Figure 4.1 shows two possible scenarios. In the figure, the black and gray lines are two actual eigenvalue paths while the black and gray dots are computed eigenvalues. In the first case (with preconditioning), the computed eigenvalues follow each path respectively while in the second case (without preconditioning), the computed eigenvalues are mismatched once the two paths have crossed.

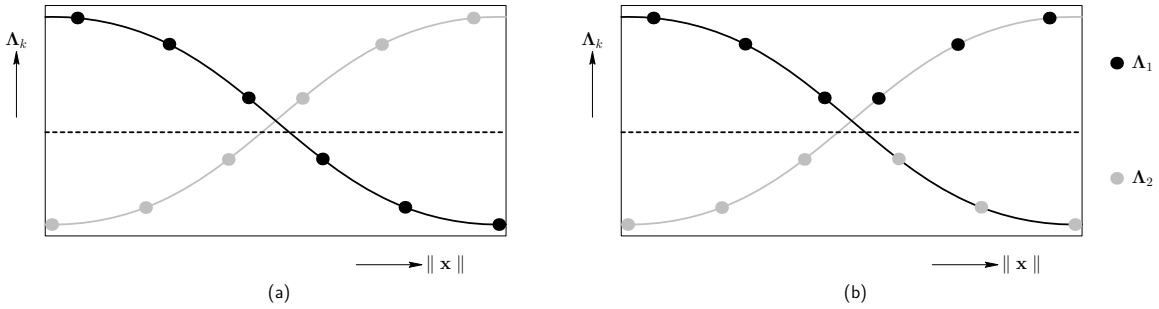

Figure 4.1: (a) With preconditioning, (b) without preconditioning.

In general, since we have assumed that $\mathbf{F}$ is derived from a potential function ($\mathbf{F} = \nabla_{\mathbf{x}}\mathcal{E}$), there is a rotation matrix $\mathbf{R}$ that rotates the basis in order to diagonalize $\nabla_{\mathbf{x}}\mathbf{F}$. That is, there exists $\mathbf{R} \in SO(N)$ such that

$$\mathbf{R}^{\top}\nabla_{\mathbf{x}}\mathbf{F}\mathbf{R} = \mathbf{\Lambda}, \tag{4.3}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, $\mathbf{R} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N]$, and $\{\mathbf{v}_j \mid j = 1, 2, \ldots, N\}$, is a set of orthonormal eigenvectors of $\nabla_{\mathbf{x}}\mathbf{F}$. Now, let $\mathbf{R}_i$ and $\mathbf{R}_{i+1}$ be such that $\mathbf{R}_i^{\top}\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_i))\mathbf{R}_i = \mathbf{\Lambda}_i$ and $\mathbf{R}_{i+1}^{\top}\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_{i+1}))\mathbf{R}_{i+1} = \mathbf{\Lambda}_{i+1}$, where $\mathbf{\Lambda}_i$ and $\mathbf{\Lambda}_{i+1}$ are the diagonal matrices of eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_i))$ and $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_{i+1}))$, respectively for any two nearby solution points $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$ on the path $\mathbf{c}(s)$. The requirement of a consistent ordering of the eigenvalues implies that the rotation that diagonalizes $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_{i+1}))$ must be a rotation by a small angle ($\theta \ll 1$) of the basis in which $\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_i))$ is diagonal. Therefore, $\mathbf{R}_{i+1}$ must be a composition of $\mathbf{R}_i$ and a small angle rotation $\mathbf{R}_\theta$ such that

$$\mathbf{R}_{i+1} = \mathbf{R}_\theta\mathbf{R}_i. \tag{4.4}$$

Naturally, the rotation angle θ of the matrix $\mathbf{R}_\theta$ is highly dependent on the stepsize taken

between $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$. Generally the larger the stepsize, the larger θ will be.

Based on this formulation, the consistent eigenvalues of s_{i+1} may be obtained by finding the eigenvalues of the preconditioned matrix $\mathbf{R}_i^\top \nabla_{\mathbf{x}} \mathbf{F}(\mathbf{c}(s_{i+1}))\mathbf{R}_i$ which will be nearly diagonal. Most eigenvalue finding algorithms will perform the equivalent of a similarity transformation using the small rotation $\mathbf{R}_\theta$ to obtain $\mathbf{\Lambda}_{i+1}$. However, this is not guaranteed and, thus, one should confirm this property for the algorithm to be employed.

It is clear that if the two solution points are "far" apart, then the small angle rotation requirement for Eq. (4.4) will be violated and, as such, there could be instances where the presence of critical points between two solution points may be undetected or a critical point is predicted when none exists. A steplength adaptation criterion can be developed to control how large a stepsize can be taken to ensure a reliable detection of all critical points. However, this has not been pursued as part of this work.

4.1.3 Computation of critical points

A systematic approach to accurately compute critical points using the criterion in Eq. (4.2) is now presented. As in Chapter 3, critical points are divided into two types: symmetry-breaking and turning (symmetry-preserving) points. For critical points of the first type, it was shown that a reduced set of equilibrium equations can be obtained whereby the gradient of the reduced system will be non-singular at the symmetry-breaking critical point. Similarly, for critical points of the second type, it was shown in Chapter 3 that use of a pseudo-arclength algorithm will result in a system of linear incremental equations that will be non-singular at the turning point.
Thus, by keeping track of the eigenvalues of the full system gradient as discussed in the previous section, one can use the reduced system of equations, an appropriate pseudo-arclength branch-following algorithm, and a method such as "Bisection method" or "Brent's method" (both discussed later in this subsection) to accurately compute all critical points along the path $\mathbf{c}(s)$. For the remainder of this discussion on critical point calculation, it is assumed that a reduced system of equations is employed. Furthermore, it is also assumed that the path-following algorithm has identified at least one eigenvalue (and its associated index k) that has changed sign between the two solution points $\mathbf{c}(s_i)$ and $\mathbf{c}(s_{i+1})$ as per Eq. (4.2).

Once one or more critical points have been bracketed between s_i and s_{i+1}, the PACA algorithm can be used to iteratively solve for the critical point. To see why the PACA (and not PATA) must be used, recall that the PACA augments the system of equations with a

constraint that forces the algorithm to find only solutions on a ball of given radius ds. That is, if $\mathbf{c}(s_i)$ is an arbitrary point along the equilibrium path, using the constraint approach, a stepsize of ds'' or ds' will yield points along the equilibrium path $\mathbf{c}(s''_{i+1})$ and $\mathbf{c}(s'_{i+1})$, respectively, where $s''_{i+1} > s'_{i+1}$ if and only if $ds'' > ds'$ (Figure 4.2). In comparison, the PATA provides no such guarantee as to the ordering of s'_{i+1} and s''_{i+1} (Figure 4.3). Thus, the constraint method provides access to the *function* $\mathbf{c}(s)$ whereas the tangent method provides access only to the *set of solutions* $\{\mathbf{c}(s)\}$. Since the critical point calculation methods presented in this work are based on root finding algorithms for *functions* of a single argument (in this case $\Lambda_k(s)$), the PACA must be used. Thus, one of the following two methods can be used to compute each critical point that has been bracketed between s_i and s_{i+1}.

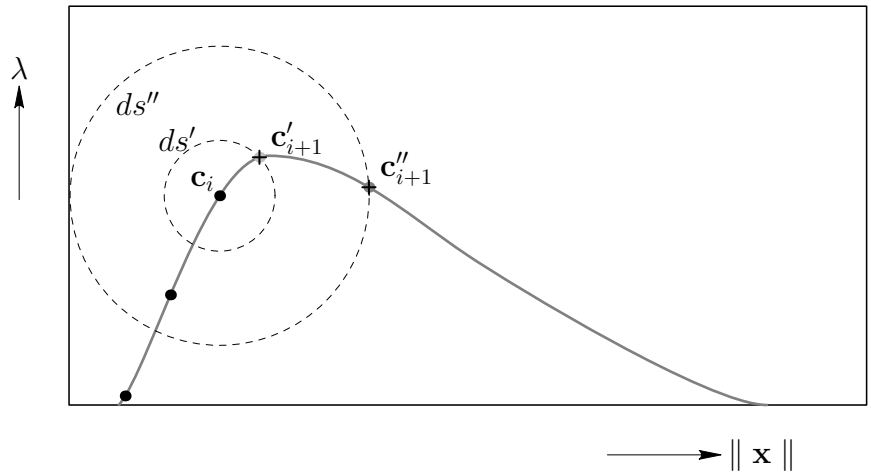

Figure 4.2: Branch-following with PACA.

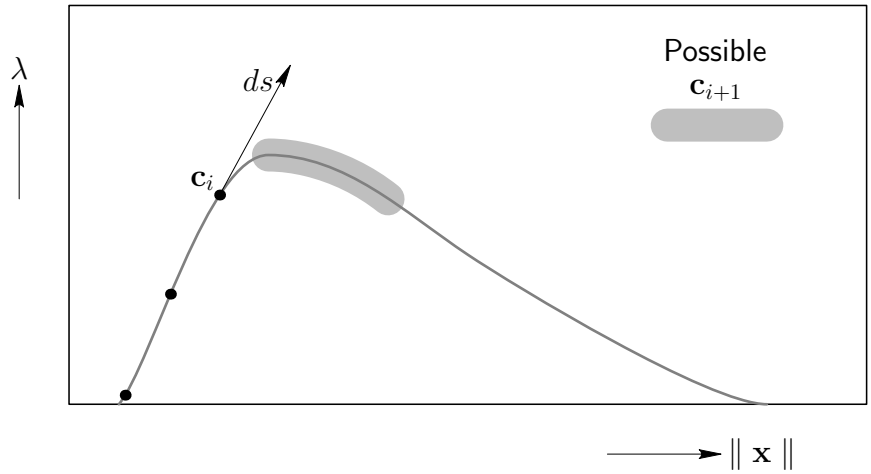

Figure 4.3: Branch-following with PATA.

1. *Bisection*

   The bisection method is a reliable method to iteratively solve for the critical point $\mathbf{c}(s_c)$ such that $\mathbf{F}(\mathbf{c}(s_c)) = \mathbf{0}$ and $\Lambda_k(s_c) = 0$. The bisection method is desirable since it is an unfailing method to find the zero crossing of $\Lambda_k(s)$. The idea is that over the interval $[s_i, s_{i+1}]$, the function $\Lambda_k(s)$ is known to pass through zero because it changes sign. The function $\Lambda_k(s)$ is first evaluated at the interval's midpoint $(s_i + s_{i+1})/2$ and its sign examined in order to determine in which half of the interval the critical point resides. With each iteration, the interval in which the point satisfying $\Lambda_k(s_c) = 0$ decreases in size by a factor of two. Therefore, after some number of iterations one has that $|\Lambda_k(s)| < \epsilon$ for all s in the interval. Here, $\epsilon > 0$ is the prescribed tolerance. *Press et al.* (1988), shows that the number of iterations n required to achieve a given tolerance ϵ is $n = \log_2 \frac{\Delta}{\epsilon}$ where Δ is the size of the initial bracketing interval. Algorithm 5 gives the pseudo-code for the computation of critical points using the bisection method. Recall that $\mathbf{F}_G(\mathbf{v}, \lambda) \equiv \mathbf{\Psi}_G^\top \mathbf{F}(\mathbf{\Psi}_G \mathbf{v}, \lambda)$ are the symmetry-reduced equations along the path $\mathbf{c}(s)$ and $\mathbf{F}'_G \equiv [\nabla_{\mathbf{v}} \mathbf{F}_G, \frac{\partial \mathbf{F}_G}{\partial \lambda}]$ is the Jacobian of $\mathbf{F}_G$. Also, note that $\mathbf{c}_G(s) \equiv (\mathbf{v}(s), \lambda(s))$ is the G-symmetric solution path of $\mathbf{F}_G$ which can be mapped back to the solution path of $\mathbf{F}$ as $\mathbf{c}(s) \equiv (\mathbf{\Psi}_G \mathbf{v}(s), \lambda(s))$.

2. *Brent's method*

   The bisection algorithm converges linearly to the root $\Lambda_k(s_c) = 0$. Ideally, an algorithm with superlinear convergence properties would be used. Although there are several methods that converge superlinearly onto a root (such as secant, false position, and Ridder's method), discontinuous functions, or even smooth functions whose second derivative changes dramatically near the root can pose a problem for some of these methods. As such, it is desirable to combine superlinear convergence with the sureness of the bisection method as detailed above. Brent's method is one such method. Brent's method combines root bracketing, bisection, and inverse quadratic interpolation to converge from the neighborhood of a root. The idea behind this method is that we can keep track of whether a supposedly superlinear method is actually converging the way it is supposed to. If it is not, we can perform a bisection step to guarantee at least linear convergence. With reference to *Press et al.* (1988) for the Brent's method algorithm, the implementation of Brent's method for computing critical points follows immediately as an extension of the bisection Algorithm.

Algorithm 9 Critical point calculation using Bisection method and PACA

Input: $\mathbf{c}_{G_i}, \mathbf{c}_{G_{i+1}}, \epsilon$
$\Delta\mathbf{c} := \mathbf{c}_{G_{i+1}} - \mathbf{c}_{G_i}$;
$ds :=\| \Delta\mathbf{c} \|$;
$\mathbf{dR} := \mathbf{c}_{G_{i+1}}$;
$\ell := ds$; %Current interval length
$ds := ds/2.0$;
$\Delta\mathbf{c} := \Delta\mathbf{c}/ \| \Delta\mathbf{c} \|$; %Approximation to the tangent at $\mathbf{c}_{G_i}$
$\mathbf{d} := \mathbf{c}_{G_i}$;
while $|\Lambda_k(\mathbf{d})| > \epsilon$ **do**
  $\mathbf{d} := \mathbf{c}_{G_i} + ds\Delta\mathbf{c}$;
  while $\sqrt{\| \mathbf{F}_G(\mathbf{d}) \|^2 + \frac{1}{2}(\| \mathbf{d} - \mathbf{c}_{G_i} \|^2 - ds^2)} > \epsilon$ **do**
    $\mathbf{d} := \mathbf{d} - \begin{bmatrix} \mathbf{F}'_G(\mathbf{d}) \\ (\mathbf{d} - \mathbf{c}_{G_i})^\top \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F}_G(\mathbf{d}) \\ \frac{1}{2}(\| \mathbf{d} - \mathbf{c}_{G_i} \|^2 - ds^2) \end{bmatrix}$; % Find current interval midpoint solution
  end while
  $\ell := \ell/2.0$; % New interval length
  if $\Lambda_k(\mathbf{d}) \cdot \Lambda_k(\mathbf{dR}) \geq 0$ **then**
    $ds := ds - \ell/2.0$; %Solution is in left half of interval
    $\mathbf{dR} := \mathbf{d}$;
  else
    $ds := ds + \ell/2.0$;
  end if
end while
$\mathbf{c}_c := \mathbf{d}$;
Return $\mathbf{c}_c$;

4.2 Classification of critical points

Once a critical point has been accurately computed, it may be desirable to determine whether it is a symmetry-breaking bifurcation point or a symmetry-preserving turning point. In order to distinguish between these two types of critical points, consider the parameterization of the equilibrium path $\mathbf{c}(s) = (\mathbf{x}(s), \lambda(s))$. Taking the derivative with respect to s of $\mathbf{F}(\mathbf{c}(s)) = \mathbf{0}$, we obtain the following equation:

$$\frac{d}{ds}[\mathbf{F}(\mathbf{c}(s))] = \left(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s)) \frac{d\mathbf{x}(s)}{ds} + \frac{\partial \mathbf{F}(\mathbf{c}(s))}{\partial \lambda} \frac{d\lambda(s)}{ds} \right) = \mathbf{0}. \tag{4.5}$$

In the general case, it may be that several eigenvalues of $\nabla_{\mathbf{x}}\mathbf{F}$ have simultaneously vanished at a critical point. Let $\{\mathbf{v}_i |\ i = 1, 2, \ldots, m\}$ be the set of m eigenvectors of $\nabla_{\mathbf{x}}\mathbf{F}$ with zero eigenvalue. Since $\mathbf{F} \equiv \nabla_{\mathbf{x}}\mathcal{E}$, then $\nabla_{\mathbf{x}}\mathbf{F}$ is real and symmetric which implies that the left eigenvectors are equal to the right eigenvectors and the vectors may be chosen such that $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ for $i \neq j$. Dotting Eq. (4.5) at a critical point with each $\mathbf{v}_i$, the following is obtained

$$\left(\mathbf{v}_i \cdot \nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_c)) \frac{d\mathbf{x}(s_c)}{ds} + \mathbf{v}_i \cdot \frac{\partial \mathbf{F}(\mathbf{c}(s_c))}{\partial \lambda} \frac{d\lambda(s_c)}{ds} \right) = 0, \quad i = 1, \ldots, m. \tag{4.6}$$

By the definition of a critical point, $\mathbf{v}_i \cdot \nabla_{\mathbf{x}}\mathbf{F}(\mathbf{c}(s_c)) = \mathbf{0}$ for each i, and thus, Eq. (4.6), reduces to

$$\mathbf{v}_i \cdot \mathbf{F}^c_\lambda \frac{d\lambda(s_c)}{ds} = 0, \quad i = 1, \ldots, m, \tag{4.7}$$

where $\mathbf{F}^c_\lambda \equiv \frac{\partial}{\partial \lambda}\mathbf{F}(\mathbf{c}(s_c))$. Refering to Eq. (4.7), two types of critical points can be identified:

1. *Bifurcation points*

   A bifurcation point is a critical point that satisfies

   $$\mathbf{v}_i \cdot \mathbf{F}^c_\lambda = 0, \quad \forall i. \tag{4.8}$$

   This, in general implies that $\frac{d\lambda(s_c)}{ds} \neq 0$.

   Bifurcation points are characterized by the presence of multiple solution paths intersecting. Moreover, bifurcation points are further categorized into two types: "asymmetric" bifurcation points that are characterized by non-zero tangent with respect to the loading at the bifurcation point (Fig. (4.4)); and "symmetric" bifurcation points

which have zero tangent at the bifurcation point (Fig. (4.5)). The number of solution paths branching out of a bifurcation point can be more than one and is dependent on the number of eigenmodes of the gradient with zero eigenvalues. Furthermore, the number of paths branching out of a bifurcation point depends on the type of bifurcation point, whether it is symmetric or asymmetric. If m is the multiplicity of the zero eigenvalue and M the number of paths that pass through the point, then for a symmetric bifurcation point, $M_{symmetric} \leq \frac{3^m-1}{2}$ while for an asymmetric bifurcation point, $M_{asymmetric} \leq 2^m - 1$ (refer to *Triantafyllidis and Peek* (1992)).

One method to guide the numerical solution to $\mathbf{F}(\mathbf{x}, \lambda) = \mathbf{0}$ near bifurcation points is to perform an asymptotic analysis of the bifurcated paths at the bifurcation points. *Triantafyllidis and Peek* (1992) present an apporach based on the Lyapunov-Schmidt-Koiter decomposition which shows that the tangents of the bifurcated paths at a bifurcation point are linear combinations of the zero eigenvectors of the gradient at the bifurcation point. As such, it may be important that the bifurcation point be computed precisely in order to obtain the eigenvectors with eigenvalue equal to zero.

2. *Turning points*
   A turning point is a critical point that satisfies the following condition:

$$\mathbf{v}_i \cdot \mathbf{F}^c_\lambda \neq 0, \quad \text{for some } \; i \in [1, m]. \tag{4.9}$$

This generally implies that $\frac{d\lambda(s_c)}{ds} = 0$. Thus, a turning point is characterized by a local extremum of the solution path $\mathbf{c}(s)$ as a function of the loading parameter and as such is refered to as a point of "limit load" (refer to Fig. (4.6)). In physics and engineering applications, a turning point can often signify a change in the stability of the solutions. Unlike bifurcation points, the gradients of the full equations and the reduced equations are singular at turning points. However, since $\mathbf{v}_i \cdot \mathbf{F}^c_\lambda \neq 0$ for some i, there exists a unique path $\mathbf{c}(s)$ extending through the critical point that can be computed by a branch-following method such as the pseudo-arclength methods.

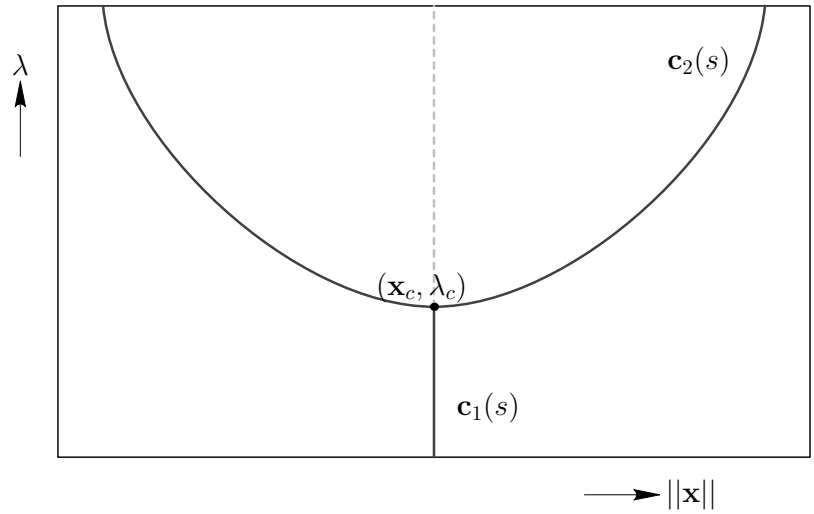

Figure 4.4: A symmetric bifurcation point at $(\mathbf{x}_c, \lambda_c)$.

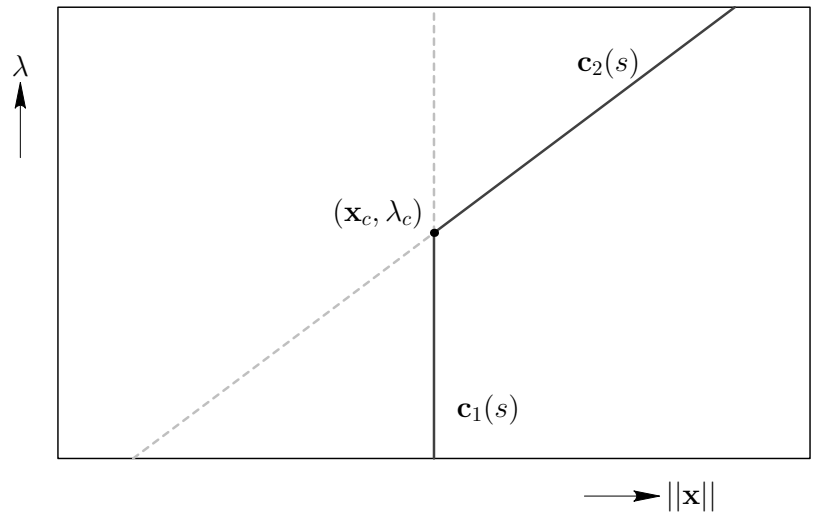

Figure 4.5: An asymmetric bifurcation point at $(\mathbf{x}_c, \lambda_c)$.

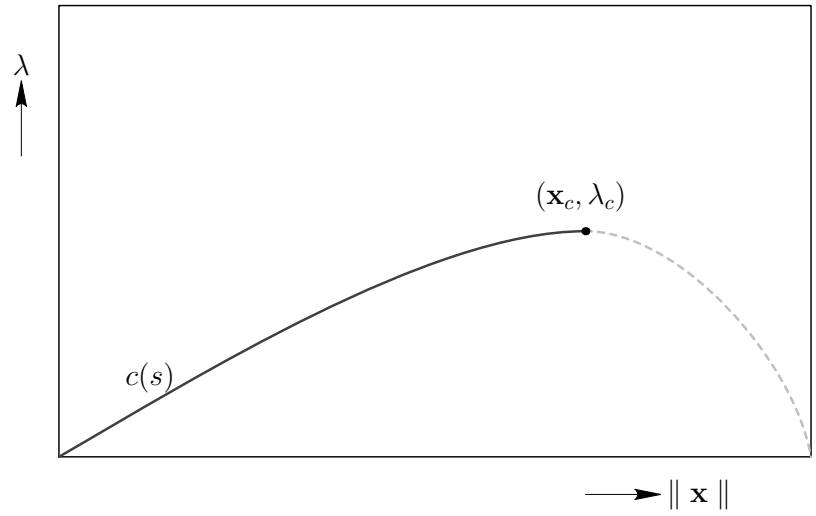

Figure 4.6: A turning point at $(\mathbf{x}_c, \lambda_c)$.

Chapter 5

Applications

In this chapter, two application problems will be used to illustrate the methods described in the preceding chapters and to evaluate the relative performance of the methods. First, the principal equilibrium path of a three-dimensional atomistic model with a large number of degrees of freedom (DOFs) will be studied and the performance of the PACA and PATA algorithms will be compared. Next, to illustrate the benefits of the methods used to deal with discrete and continuous symmetry (discussed in Chapter 2), a system with a small number of DOFs will be studied. The symmetry-reduction method will be used to follow the principal equilibrium path of a one-dimensional atomistic binary chain model. Additionally, all critical points on this branch will be accurately computed and identified. The advantages of using the Projection and Phantom Energy methods as well as the relative performance of the two methods will then be discussed.

5.1 A short rectangular column of FCC nickel under compression

To compare the performance of the two path-following algorithms and their Jacobian updating variants discussed in Chapter 3, the techniques were used to compute the primary equilibrium branch for an energy function with 324 degrees of freedom. 500 solution points were computed using a desktop with 1.86 GHz Intel Core2 CPU and 1 Gigabyte of RAM.

5.1.1 Problem description

The study computed the principal compressive deformation path of a three-dimensional perfect FCC crystal of a short rectangular column consisting of nickel atoms subject to uniaxial displacement control (Fig. (5.1)). This problem is studied in more depth by *Sendova*

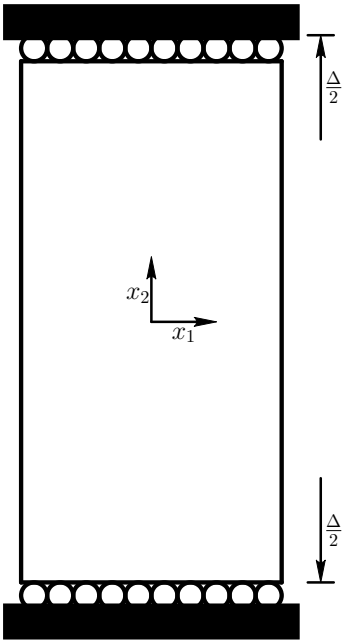

Figure 5.1: Rectangular column under displacement control.

et al. (2010). Periodic boundary conditions were used to simulate bulk behavior in the x_3 direction. The interaction between the nickel atoms is described using a smoothed version of the potential by *Angelo et al.* (1995) as described in *Dupuy et al.* (2005). Using the interatomic potential, the energy $\mathcal{E}(\mathbf{u}, \Delta)$ (per unit of x_3-length) of the column can be computed in a straightforward manner (for example, refer to *Tadmor and Miller* (2010)). The equilibrium equations are then obtained as in Eq. (2.1). Here, $\mathbf{u}$ is a vector of atomic displacement components. These include three spatial components (x_1, x_2, x_3) for atoms away from the boundaries. For atoms at the top and bottom of the column, only x_1 components of the displacement are free degrees of freedom, while the x_2 and x_3 displacements are determined by the boundary conditions and the loading parameter Δ. Thus, the current position $\mathbf{x}[\alpha]$ of atom $\alpha \in \{1, 2, \ldots, N\}$ is given by

$$\mathbf{x}[\alpha] = \mathbf{X}[\alpha] + \mathbf{u}[\alpha], \tag{5.1}$$

where $\mathbf{X}[\alpha]$ and $\mathbf{u}[\alpha]$ are the reference position and displacement vectors, respectively.

In this simulation, all atoms in the crystal were allowed to translate freely in the x_1 direction. It can be shown that in this case, there exists a vector $\widehat{\mathbf{T}}$ such that the energy function is invariant under the elements $\mathbf{T} = t \cdot \widehat{\mathbf{T}}$ of the translation group $\mathcal{T}$, where $t \in \mathbb{R}$ and $\widehat{\mathbf{T}}$ is a unit vector in $\mathbb{R}^{324}$ corresponding to all equal x_1 displacements and zero displacement in

the x_2 and x_3 directions. That is

$$\mathcal{E}(\mathbf{u}, \Delta) = \mathcal{E}(\mathbf{u} + \mathbf{T}, \Delta), \qquad \forall \mathbf{T} \in \mathcal{T}. \tag{5.2}$$

Using this information, the restriction of the equilibrium equations onto the space $\mathcal{T}^{\perp}$, as described in Chapter 2, can be obtained from the Projection method or Phantom Energy method.

5.1.2 Principal equilibrium path

The principal equilibrium path (the branch corresponding to the initial symmetry of the problem) of the system is plotted in Fig. (5.2) where Δ and F are the end displacements and conjugate force to the displacement, respectively. Here, solid green lines and dashed red lines indicate stable and unstable equilibrium solutions, respectively. The plot is generated by computing 2500 solution points using the PATA with QR updating scheme (discussed in Chapter 3) and the Phantom Energy method (discussed in Chapter 2) with a maximum stepsize of $ds = 0.1$. As Δ is incrementally decreased, the configuration of the nickel column deforms smoothly from the initial configuration shown in Fig. (5.3), to a final configuration shown in Fig. (5.6). Figures (5.4) and (5.5) show intermediate configurations of the column. Note that the reference configuration ($\Delta = 0$) is a state of tension for the system and that a stable state of zero stress is observed at around $\Delta = -1.2$ $\overset{o}{\text{A}}$. As Δ is decreased further, the system continues to deform smoothly until a bifurcation point is encountered at a displacement of about $\Delta = -2.5$ $\overset{o}{\text{A}}$, at which point the equilibrium path of the original symmetry becomes unstable. However, stable solutions are encountered again further along the equilibrium path as can be seen from Fig. (5.2). Also note the presence of numerous critical points starting at around $\Delta = -2.48$ $\overset{o}{\text{A}}$ with a turning point present at $\Delta = -2.53$ $\overset{o}{\text{A}}$.

5.1.3 Performance of the Jacobian updating schemes

Next, the simulation was repeatedly performed to evaluate the relative performance between the Jacobian updating schemes used in the PATA algorithm. 500 solution points were computed using three different stepsizes of $ds = 1.0$, 0.1, and 0.01 to evaluate the stepsize dependency of the various Jacobian updating schemes. The following tables summarize the results of the simulations.

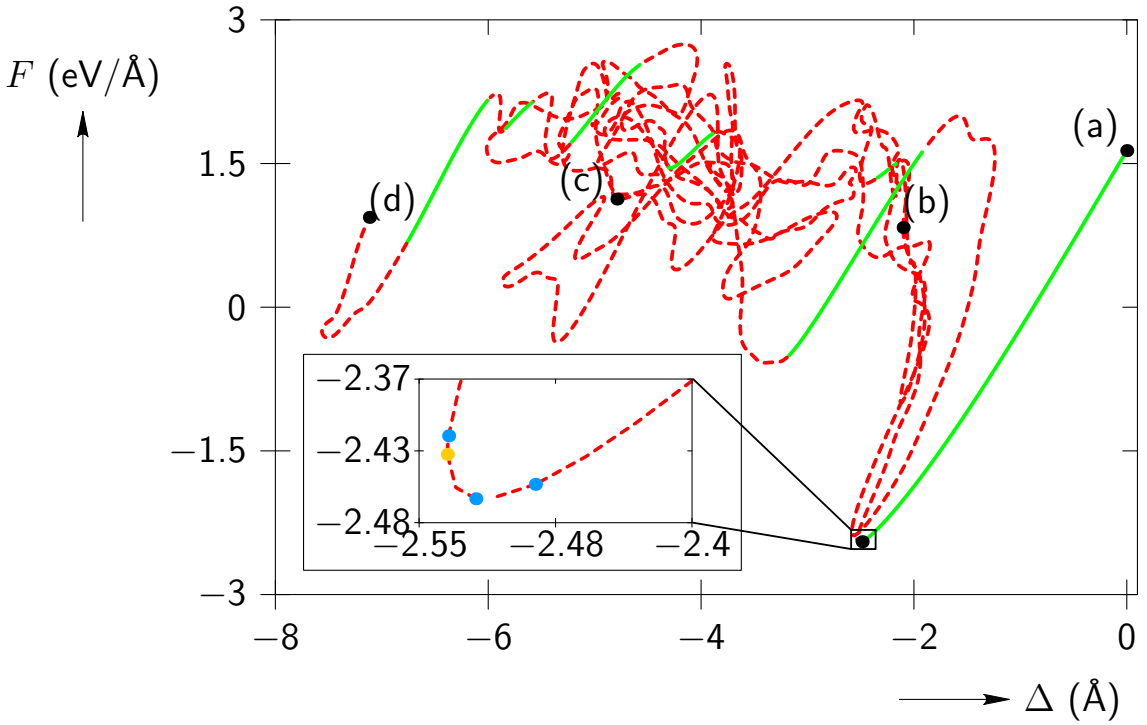

Figure 5.2: Plot of the behavior of a rectangular perfect crystal of nickel under compression. Δ and F are the end displacement and conjugate force to the displacement in the x_2 direction respectively. Inset is a zoomed in picture with the yellow and blue dots indicating turning points and bifurcation points, respectively.

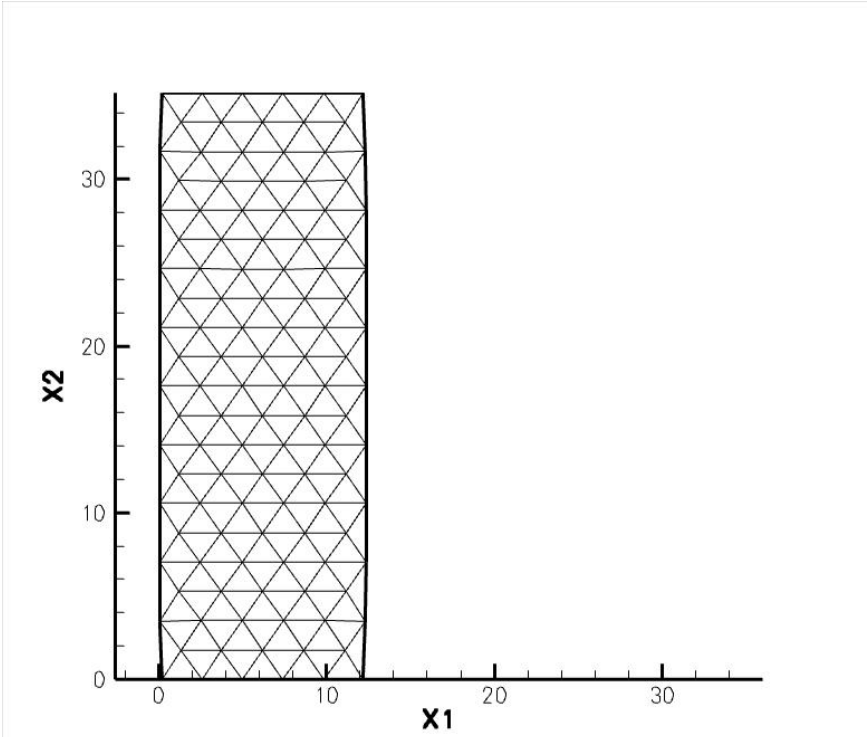

Figure 5.3: Point (a) - 1st solution point.

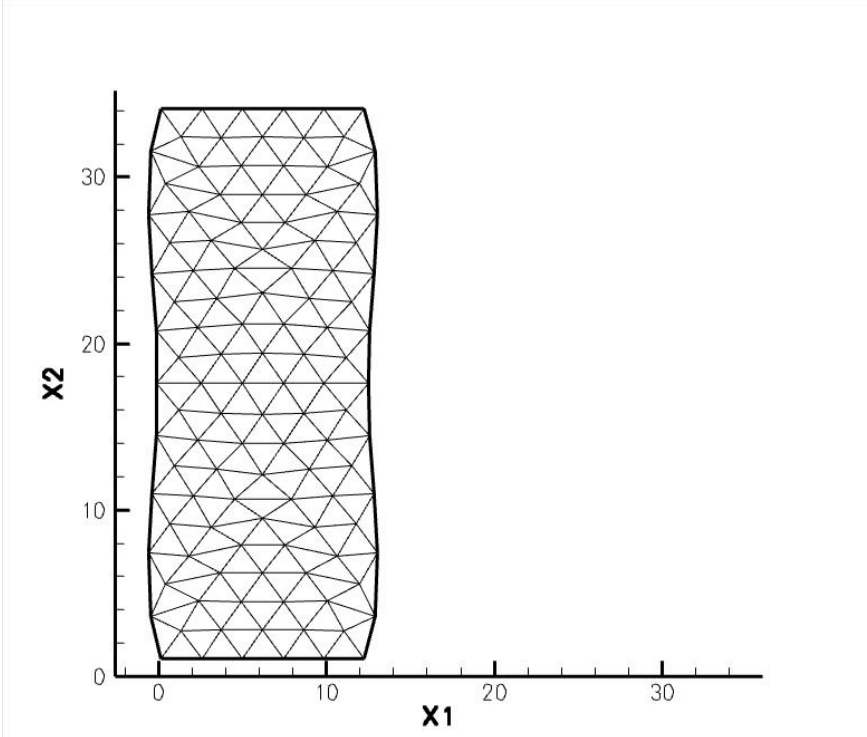

Figure 5.4: Point (b) - 250th solution point.

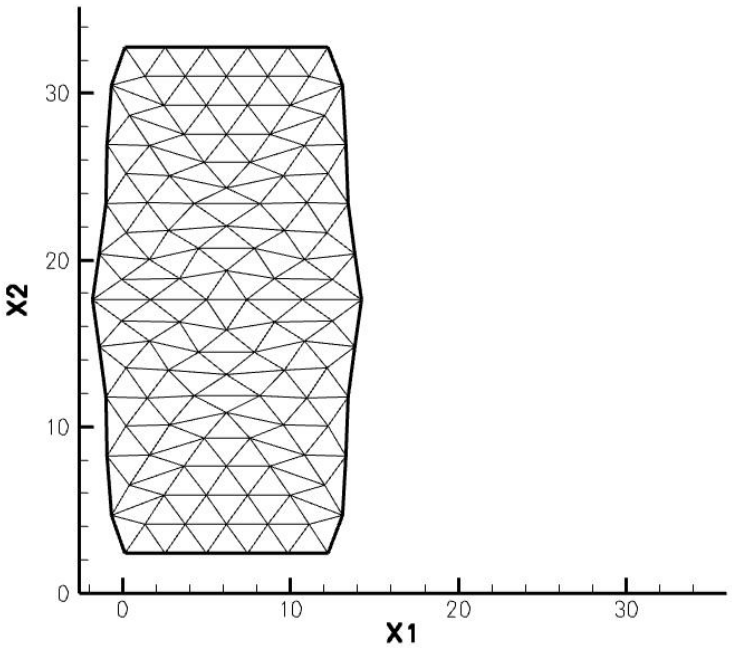

Figure 5.5: Point (c) - 500th solution point.

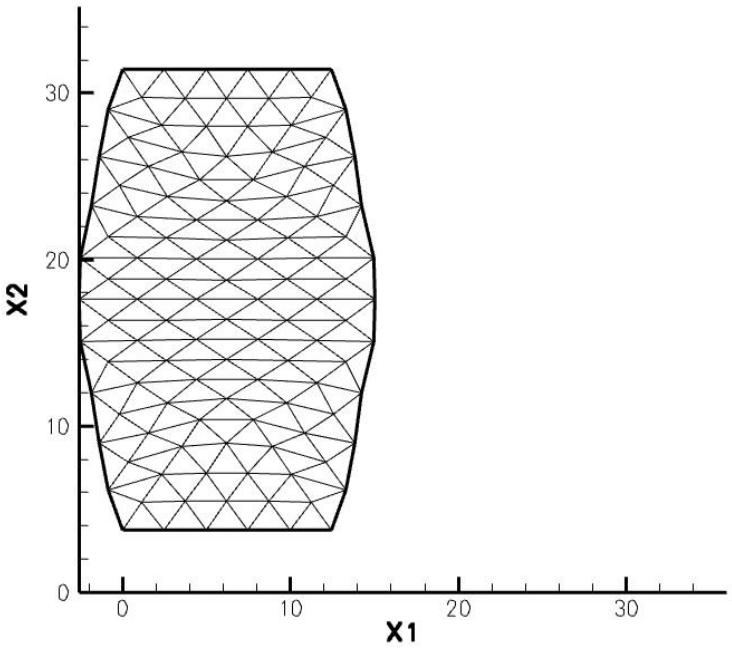

Figure 5.6: Point (d) - 2500th solution point.

5.1.3.1 Comparison of results using the Projection method

Tables 2 to 4 present data from the simulation of 500 solution points using three different stepsizes. The data presented are normalized by, and are presented in the tables as a percentage of, the values obtained using a stepsize of $ds = 1.0$ with the Exact updating scheme. These results are provided in Table 1 and were chosen since that simulation took the longest time to complete. The following describes the headings in the table:

1. **Time**: Gives the total time taken.
2. **GetQR**: Function call to obtain an updated QR decomposition.
3. **MoorePenrose**: Function call to perform a linear solve using the Moore-Penrose inverse.
4. **E1**: Function call to compute the first derivative of the energy.
5. **E2**: Function call to compute the second derivative of the energy.

Note that all numbers are normalized and given as % of the values obtained for the Projection method and Exact updating.

Time (Clock ticks)	**GetQR (# calls)**	**MoorePenrose (# calls)**	**E1 (# calls)**	**E2 (# calls)**
2,860,695,187	1095	1627	2675	2645

Table 1: Values obtained using PATA with Exact updating and Projection method.

The data presented in Table 2 were obtained using a stepsize of $ds = 0.01$ which is the smallest stepsize used in the simulations in this subsection. Here, all updating schemes perform similarly and resulted in nearly identical computation times. This occurs because the small stepsize used in this simulation ensures that the Jacobian has not deviated significantly from the Jacobian computed at the previous solution point. As such, differences between the Jacobians obtained using the various updating methods are very small and result in similar convergence rates of the corrector process. Finally, note that a faster time is obtained using QR updating. However, since in theory No updating should be the most efficient scheme (given equal convergence rates), we suspect that this marginal difference in performance is due to a difference in the workload on the system that is outside of the computations performed in the simulations.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
No update	29.90	11.78	38.66	60.93	56.82
QR update	29.86	11.78	38.66	60.93	56.82
Jacobian update	31.68	11.78	38.66	60.93	56.82
Exact update	36.92	11.78	38.66	60.93	61.70

Table 2: Updating comparison using $ds = 0.01$ (with PATA and Projection method).

Table 3 shows data obtained using a stepsize of $ds = 0.1$, ten times larger than the stepsize in the previous set of simulations. Note that the GetQR, MoorePenrose, and E1 functions are called more frequently than the previous case implying a slower convergence rate in the corrector process. Furthermore, it can be seen that, when compared to the $ds = 0.01$ results, the No updating and QR updating schemes show a marginal increase in the computation time, while the Jacobian and (more drastically) the Exact updating schemes show a marked increase in the time to compute the set of 500 solution points. This implies that: (1) the QR decomposition of the Jacobian is computationally expensive since QR updating updates the QR matrices directly and thus bypasses the QR decomposition of the Jacobian entirely. This explains the increase in time from the QR to the Jacobian updating schemes: and (2) almost 45% of the computational cost in PATA with Exact updating is associated with the computation of the Jacobian matrix.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
No update	29.97	87.58	89.86	92.19	56.94
QR update	30.07	86.39	89.06	91.74	56.98
Jacobian update	45.94	86.39	89.06	91.74	56.98
Exact update	81.35	80.00	84.70	89.01	90.02

Table 3: Updating comparison using $ds = 0.1$ (with PATA and Projection method).

The data in Table 4 were obtained using a stepsize of $ds = 1.0$, which is the largest stepsize performed in the simulations in this subsection. The values for the Exact updating in this table are those that are used to normalize all other values in the preceding tables with the actual numbers given in Table 1. Note that as in Table 3, there is an increase in the number of GetQR, MoorePenrose, and E1 function calls. Furthermore, conclusion (2) in the above paragraph is reaffirmed.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
No update	33.10	130.68	120.84	112.38	58.30
QR update	34.23	128.49	118.81	110.92	58.07
Jacobian update	57.40	128.49	119.00	111.14	58.19
Exact update	100.00	100.00	100.00	100.00	100.00

Table 4: Updating comparison using $ds = 1.0$ (with PATA and Projection method).

5.1.3.2 Comparison of results using the Phantom Energy method

Tables 5 to 7 present data for the same set of calculation scenarios as those given in the preceding three tables but with the use of the Phantom Energy method as opposed to the Projection method. As before, the data is normalized by the values (given in Table 1) obtained from the simulation using Projection method and Exact updating with a maximum stepsize of $ds = 1.0$ and is presented in the tables as a percentage of those values.

Similar to the data in Table 2, it is observed from Table 5 that, with a stepsize of $ds = 0.01$ and using the Phantom Energy method, the various updating schemes were found to be similar in performance with computation times that are nearly identical. This is expected since the small stepsize implies that the Jacobian does not significantly change from step to step and results in similar convergence rates in the corrector process.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
No update	10.14	11.78	38.66	60.93	56.82
QR update	10.15	11.78	38.66	60.93	56.82
Jacobian update	12.28	11.78	38.66	60.93	56.82
Exact update	12.75	11.78	38.66	60.93	61.70

Table 5: Updating comparison using $ds = 0.01$ (with PATA and Phantom Energy method).

Table 6 gives the data obtained using $ds = 0.1$. Note that there is a marked increase in the computation time between the QR and Jacobian updating schemes reaffirming conclusion (1) in the previous subsection — that the computational cost of performing a QR decomposition of the Jacobian is high. Also observe that the difference in computation time between the Jacobian and Exact updating is not as drastic as that observed for the Projection method (Table 3). This implies that the main reason for the drastic increase in computation time

for the Exact updating in Table 3 is due to the matrix multiplication $\widehat{\mathbf{K}} = \mathbf{\Psi}_{\mathcal{T}^\perp}^\top \mathbf{K} \mathbf{\Psi}_{\mathcal{T}^\perp}$ necessary in the Projection method.

	Time (%)	**GetQR (%)**	**MoorePenrose (%)**	**E1 (%)**	**E2 (%)**
No update	10.39	87.58	89.86	92.19	56.94
QR update	10.76	86.39	89.06	91.74	56.98
Jacobian update	25.90	86.39	89.06	91.74	56.98
Exact update	28.10	80.00	84.70	89.01	90.02

Table 6: Updating comparison using $ds = 0.1$ (with PATA and Phantom Energy method).

The data given in Table 7 is obtained from simulations performed using a maximum stepsize of $ds = 1.0$. Here, it is observed that the Jacobian updating scheme yields the largest time needed to complete the simulation. This implies that by using the Phantom Energy method, and thereby removing the need to perform the matrix multiplications of $\widehat{\mathbf{K}} = \mathbf{\Psi}_{\mathcal{T}^\perp}^\top \mathbf{K} \mathbf{\Psi}_{\mathcal{T}^\perp}$, faster convergence rates at large stepsizes can be achieved with the exact computation of the Jacobian.

	Time (%)	**GetQR (%)**	**MoorePenrose (%)**	**E1 (%)**	**E2 (%)**
No update	11.14	130.68	120.84	112.37	58.30
QR update	11.44	128.49	118.81	110.92	58.07
Jacobian update	34.42	128.49	118.99	111.14	58.19
Exact update	33.49	100.00	100.00	100.00	100.00

Table 7: Updating comparison using $ds = 1.0$ (with PATA and Phantom Energy method).

In all cases, it is found that the Phantom Energy method is significantly more efficient than the Projection method. This is especially evident in the case where the Jacobian of $\mathbf{F}$ is exactly computed for each corrector step (Exact update). After removing the added complications in computing the Jacobian using the Projection method, one would expect the Phantom Energy method to be slowest when using the Exact updating method and fastest when using the No updating method. This appears to be confirmed by the data provided in Tables 5 and 6. However, in Table 7 the No updating scheme is again the fastest scheme but slower convergence in the corrector process made the Jacobian updating scheme yield the longest time needed to compute all 500 solution points. Furthermore, QR updating of

the Jacobian yields an efficiency close to that of No updating while providing a better approximation to the QR decomposition of the Jacobian at a given point (and therefore easier convergence in the corrector process) than those provided with No updating.

Observe that the time taken to compute solutions using the Jacobian updating (while not as high as Exact updating) is considerably higher than using either the No updating or QR updating scheme with the differences being more exaggerated as the maximum stepsize is increased. This implies that the process of performing a QR decomposition on the Jacobian is of considerable computational cost. This is especially evident with larger stepsizes where the function GetQR is called numerous times since more corrector steps are needed to converge onto each solution point.

Taking possible varying convergence rates into account, it seems that the best strategy is to use an updating scheme that ensures easy convergence in the corrector process (and therefore some form of updating) without actually computing the Jacobian exactly in each instance that it is required. Furthermore, since it is found that a QR decomposition of the Jacobian is of considerable computational cost, the best updating scheme, in this author's opinion, is the QR Updating scheme.

5.1.4 Benefit of using symmetry-reduction techniques

Next, to illustrate the benefit of the symmetry-reduction techniques discussed in Chapter 2, 500 solution points were computed using the three stepsizes mentioned in the preceding subsection with the QR updating scheme and the Phantom Energy method, a scheme found in the previous section to be highly efficient. Since the system possesses D_{2h} symmetry (symmetry of a rectangular block), the system can be symmetry-reduced so that the principal equilibrium path can be computed using a set of lower-dimensional equilibrium equations with 64 DOFs. This symmetry reduction was found to lead to a further decrease in the time taken to compute the solution sets for all of the stepsizes used. The following tables summarize the results of this study.

Table 8 presents the same set of data given for QR updating in Tables 5 to 7. It is reproduced here for ease of comparison.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
ds = 0.01	10.15	11.78	38.66	60.93	56.82
ds = 0.1	10.76	86.39	89.06	91.74	56.98
ds = 1.0	11.44	128.49	118.81	110.92	58.07

Table 8: Full set of 324 equations with QR updating and Phantom Energy method.

The data in Table 9 is obtained using QR updating and with the Phantom Energy method. Here, the full set of 324 equilibrium equations have been symmetry-reduced to a set of 64 equilibrium equations. Since the function calls are equal between the two sets of simulations, the time savings shown in Table 9 is entirely due to the reduction in the size of the system.

	Time (%)	GetQR (%)	MoorePenrose (%)	E1 (%)	E2 (%)
ds = 0.01	4.86	11.78	38.66	60.93	56.82
ds = 0.1	6.30	86.39	89.06	91.74	56.98
ds = 1.0	7.79	128.49	118.81	110.92	58.07

Table 9: Symmetry-Reduced set of 64 equations with QR updating and Phantom Energy method.

5.1.5 Summary of short rectangular column of FCC nickel under compression study

In this section, PATA was used to numerically compute the principal equilibrium path of a short rectangular column of FCC nickel under compression. A comparison study was performed to determine the most efficient scheme that is able to deal with the presence of continuous symmetry. Additionally, the presence of discrete symmetry allows a reduction in the size of the system of equations thereby leading to a more efficient calculation of the principal equilibrium path. Finally, this timing study suggests that for the general case, the best scheme seems to be the Phantom Energy method (discussed in Chapter 2) in conjunction with the QR updating technique (discussed in Chapter 3) applied to the symmetry-reduced equilibrium equations.

5.2 Equilibrium path-following of a one-dimensional crystal under uniaxial load

An infinite one-dimensional crystal under uniaxial load was simulated using the Projection method and Phantom Energy method that was detailed in Chapter 2. An exploration of the benefits of these methods to bifurcation analyses is presented in this section. Additionally, to illustrate the benefit of the symmetry-reduction techniques for cases with discrete symmetry, these techniques are used to compute the paths branching away from several of the symmetry-breaking bifurcation points along the principal equilibrium path.

5.2.1 Problem description

Here, the atomistic model studied by *Dobson et al.* (2007) is considered. In particular, an infinite one-dimensional bi-atomic perfect crystal in tension (Figure 5.7) is simulated in this problem. The system is simulated by taking 24 representative atoms and also by assuming periodic boundary conditions in order to simulate the bulk behavior of the material. The interaction between the atoms is described using a Lennard-Jones potential. Using PATA, 10,000 solution points were computed using a computer with a 1.86 GHz Intel Core2 CPU and 1 Gigabyte of RAM.

Figure 5.7: One-dimensional bi-atomic crystal under uniaxial tension.

The kinematics of the problem can be obtained much in the same way as that described in the previous problem (*Tadmor and Miller* (2010)). The current configuration of the system can be described by a uniform deformation of the crystal as well as the interatomic shift of each atom. That is, the current configuration $x[\alpha]$ of atom α can be obtained by the following equation (known as Cauchy-Born kinematics):

$$x[\alpha] = F(X[\alpha] + u[\alpha]). \tag{5.3}$$

Here, F is the uniform deformation of the crystal and $X[\alpha]$ and $u[\alpha]$ are the reference position and displacement away from reference position of atom α, respectively. Once the kinematics of the problem have been chosen and an interatomic potential has been specified, the energy $\mathcal{E}(\mathbf{u}, \lambda)$ density and the equilibrium equations of the crystal can be derived.

Observe that the system possesses a continuous symmetry in one direction. That is, the energy of the one-dimensional crystal is invariant under a rigid-body translation so that

$$\mathcal{E}(\mathbf{u}, \lambda) = \mathcal{E}(\mathbf{u} + \mathbf{T}, \lambda), \tag{5.4}$$

where $\mathbf{T} \in \mathcal{T}$ is a rigid-body translation. It can be shown that the translation subspace $\mathcal{T}$ is spanned by the null vector $\mathbf{T}$ of $\mathcal{E}_{,\mathbf{uu}}$. In this case, since the energy density is of the form $\mathcal{E}(F, S[1], \ldots, S[24], \lambda)$, $\mathbf{T}$ is given by $\mathbf{T} = t\,\widehat{\mathbf{T}}$ where $\widehat{\mathbf{T}} = [0, 1/\sqrt{24}, 1/\sqrt{24}, \ldots, 1/\sqrt{24}] \in \mathbb{R}^{25}$. Once this information is known, the restriction to the orthogonal complement subspace ($\mathcal{T}^{\perp}$) can be obtained through the use of either the Projection method or the Phantom Energy method as discussed in Chapter 2.

5.2.2 Simulation results

Figure 5.8(a) shows the principal equilibrium path as well as several select bifurcating branches (see also *Dobson et al.* (2007)). The label associated with a particular path indicates the number of atoms that are in the essential unit cell — the smallest unit cell that reproduces the entire crystal by periodic extension — whereas the subscript in the label indicates the corresponding crystal geometry associated with that path. For example, the label "2b" on the plot indicates that the essential unit cell in the crystal consists of two atoms per cell with the atoms equally spaced. Figure 5.8(b) shows the corresponding crystal geometry. The paths were computed using the symmetry-reduced equations, as described in Chapter 2, associated with a given symmetry subgroup of the problem. Since each path has a different symmetry, there is little danger of path "switching" in following a path of a given symmetry. In essence, by solving the symmetry-reduced equations, only those equilibrium solutions that preserve the symmetry of the desired path will be accessible. As such, path switching onto a path of lower symmetry is not possible and switching to a path of higher symmetry is found to occur only rarely in practice. The path associated with 2b is defined here as the principal path and is the path of highest symmetry. This principal equilibrium path and its associated crystal geometry can be described solely by the stretching F of the crystal as λ is varied. Other paths 5.8(c) to 5.8(g) require additional degrees of freedom (the associated atomic shift displacements) in order to fully describe the deformation of the crystal as λ is varied. The symmetry subgroup of the path bifurcating from a symmetry-breaking bifurcation point can be obtained through an asymptotic analysis as described in *Elliott et al.* (2002).

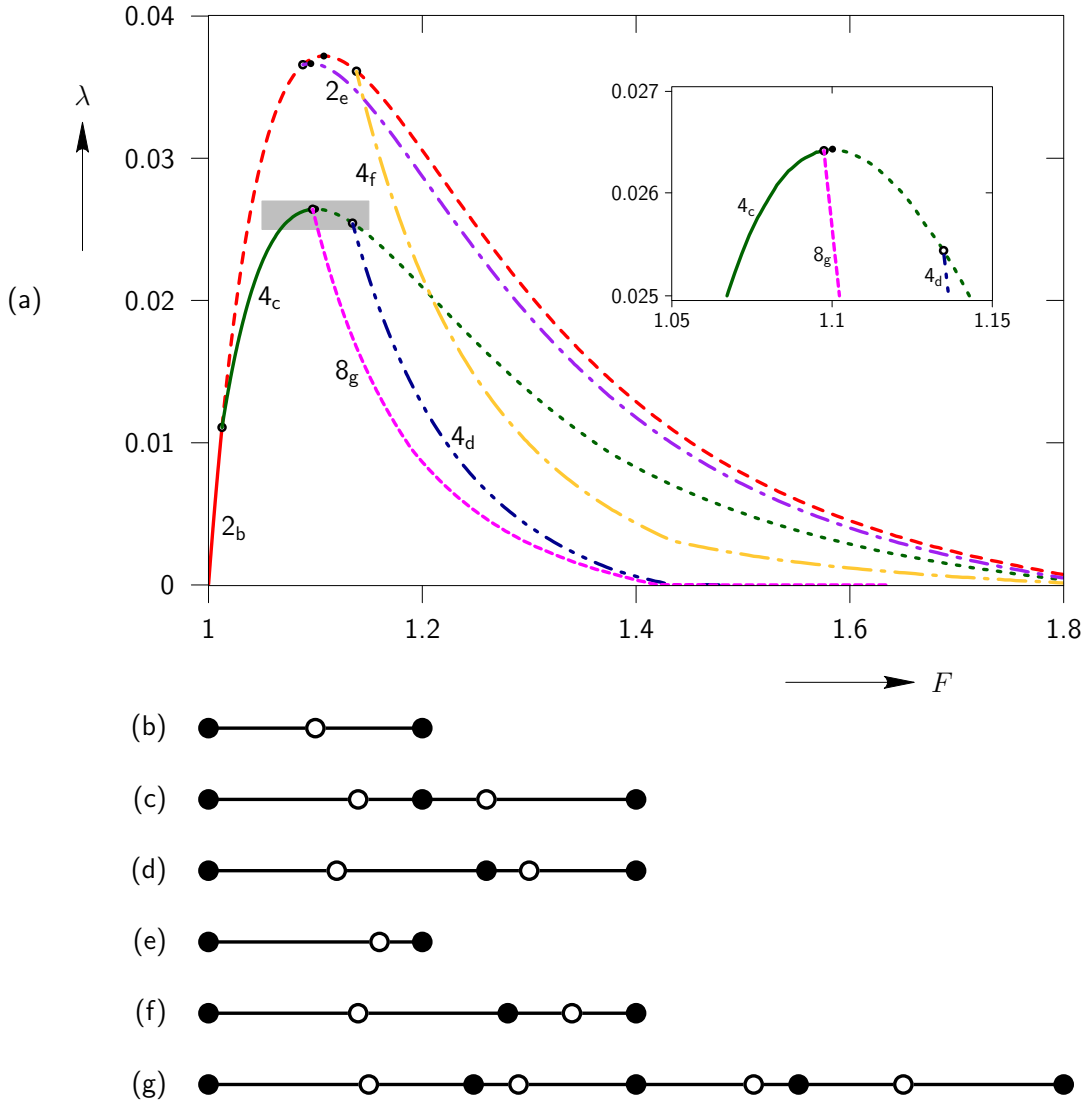

Figure 5.8: Principal path along with select bifurcating paths (*Dobson et al.* (2007)).

5.2.3 A benefit of the Projection and Phantom Energy methods

Aside from the Projection method and Phantom Energy method, a commonly used technique to eliminate rigid-body modes in simulations of crystalline materials is to set one atom in the unit cell to have zero shift (for example *Dobson et al.* (2007), *Elliott et al.* (2006), and *Pitteri* (2003)). In this way, translation modes are removed by fixing one sublattice of the crystal in space. However, this "artificial constraining" of an atom is not physical and is undesirable for bifurcation analyses as it breaks the symmetry of the system with respect to interchange of like atoms. The benefit of the Projection and Phantom Energy methods is that they retain the symmetry of the full Cauchy-Born kinematics (with

translation). Figures (5.9) and (5.10) are plots of select eigenvalues of the stiffness matrix $\mathcal{E}_{,\mathbf{uu}}$ as a function of F evaluated along the principal equilibrium path. Figure (5.9) is obtained from a conventional simulation (by setting one atom in the unit cell to have zero atomic shift) of the one-dimensional crystal, whereas Fig. (5.10) is obtained by using the Phantom Energy method (both Projection method and Phantom Energy method yield the same plot). By comparing Figs. (5.9) and (5.10), it is observed that what seem to be distinct eigenvalues Λ_1, Λ_2 and Λ_3, Λ_4 in the conventional method are clearly revealed to be symmetry related with $\Lambda_1 = \Lambda_2$ and $\Lambda_3 = \Lambda_4$ by the Phantom Energy method. This important information is needed in order to identify whether there exist multiple critical points between successive solution points (where Λ_1 and Λ_2 cross the horizontal axis, for example) or whether only a single critical point with multiple symmetry related modes occurs.

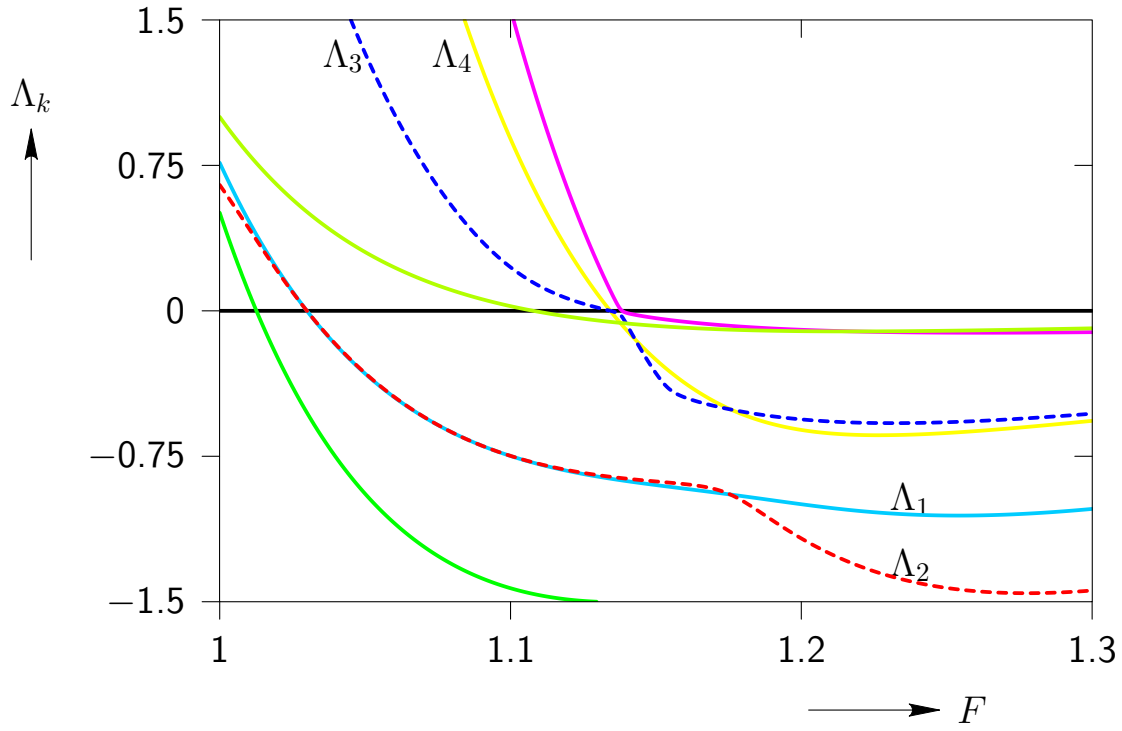

Figure 5.9: Plot of the Eigenvalue of the stiffness matrix using conventional method. Observe that eigenvalues Λ_1, Λ_2, Λ_3, and Λ_4 do not seem to be related.

5.3 Summary

In this work, general methods have been presented to compute equilibrium solutions of systems that can be described by a potential function. In Chapter 2, various techniques were detailed to deal with those systems that possess discrete and/or continuous symmetry.

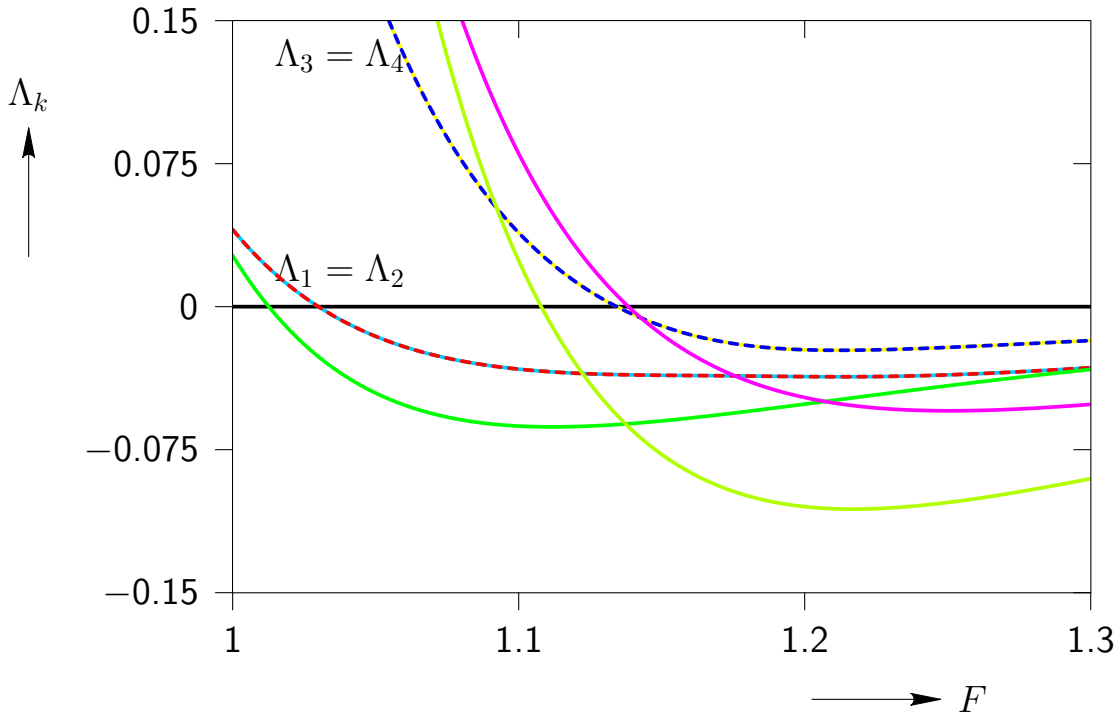

Figure 5.10: Plot of the Eigenvalue of the stiffness matrix using Phantom Energy method. Eigenvalues Λ_1, Λ_2, Λ_3, and Λ_4 are found to be symmetry-related.

In Chapter 3, path-following methods to deal with the presence of critical points along the equilibrium paths were presented. It was discussed that, in the general case, the PACA and PATA pseudo-arclength methods can be used with the symmetry-reduced equilibrium equations to compute points along the equilibrium paths regardless of what type of critical points (symmetry-breaking bifurcation points or symmetry-preserving turning points) are present along the path. Furthermore, in the pursuit of computational efficiency, various Jacobian updating schemes as well as stepsize adaptation techniques were presented in Chapter 3. In Chapter 4, criteria to identify the existence of critical points between two successive equilibrium solution points were presented. It was also discussed in this chapter that the smoothness of the Eigenvalues of the stiffness matrix must be ensured in order to correctly implement the criteria used in this work. Additionally, once a critical point has been detected, two methods were presented to compute the critical point accurately. In Chapter 5, two problems were studied to illustrate the benefits of and to compare the various techniques detailed in the preceeding chapters. With the first problem, it was found that QR updating using the Phantom Energy method on the symmetry-reduced equations was the most computationally efficient of the path-following methods. Furthermore, a benefit of the use of the Projection or Phantom Energy method for crystalline problems employing the

Cauchy-Born kinematic method was explored using the second problem in this Chapter.

These applications were used to illustrate the cumulative benefit of using the methods discussed in this work for problems that possess certain invariances of the potential function. While other methods may exist to deal with each of the issues discussed in this work, the techniques presented here have been found to be reliable and relatively easy to implement.

Bibliography

E. L. Allgower. *Numerical Continuation Methods*. Springer-Verlag, 1980.

J. E. Angelo, N. R. Moody, and M. I. Baskes. Trapping of hydrogen to lattice defects in nickel. *Modelling and Simulation in Materials Science and Engineering*, 3(3):289–307, 1995.

G. P. Berman and F. M. Izrailev. The fermi-pasta-ulam problem: 50 years of progress. *CHAOS*, 15(1):015104, 2005.

S. Bernstein. Sur la généralisation du probléme de dirichlet. *Mathematische Annalen*, 69:82–136, 1910.

B. Budiansky. Theory of buckling and post-buckling behavior of elastic structures. *Advances in applied mechanics*, 14:1–65, 1974.

S. L. Campbell and C. D. M. Jr. *Generalized Inverses of Linear Transformations*. Dover Publications, 1991.

M. Dobson, R. S. Elliott, M. Luskin, and E. B. Tadmor. A multilattice quasicontinuum for phase transforming materials: Cascading Cauchy-Born kinematics. *Journal of Computer-Aided Materials Design*, 14(Supplement 1):219–237, 2007.

L. M. Dupuy, E. B. Tadmor, R. E. Miller, and R. Phillips. Finite-temperature quasicontinuum: Molecular dynamics without all the atoms. *Phys. Rev. Lett.*, 95(6):060202, 2005.

R. S. Elliott, J. A. Shaw, and N. Triantafyllidis. Stability of thermally-induced martensitic transformations in bi-atomic crystals. *Journal of the Mechanics and Physics of Solids*, 50(11):2463 – 2493, 2002. ISSN 0022-5096.

R. S. Elliott, J. A. Shaw, and N. Triantafyllidis. Stability of crystalline solids–ii: Application to temperature-induced martensitic phase transformations in a bi-atomic crystal.

Journal of the Mechanics and Physics of Solids, 54(1):193 – 232, 2006. ISSN 0022-5096.

E. Fermi, J. Pasta, and S. Ulam. Studies of nonlinear problems. *Los Alamos Report LA-1940*, 1955.

F. A. Ficken. The continuation method for functional equations. *Communications on Pure and Applied Mathematics*, 4(4):435–456, 1951.

K. Gatermann and A. Hohmann. Symbolic exploitation of symmetry in numerical pathfollowing. *IMPACT of Computing in Science and Engineering*, 3(4):330 – 365, 1991. ISSN 0899-8248.

T. J. Healey. A group-theoretic approach to computational bifurcation problems with symmetry. *Computer Methods in Applied Mechanics and Engineering*, 67(3):257 – 295, 1988. ISSN 0045-7825.

H. B. Keller. *Lectures on Numerical Methods in Bifurcation Problems*. TATA Institute of Fundamental Research, 1987.

F. Klein. Neue beitrage zur riemannschen funktionentheorie. *Mathematische Annalen*, 21:141–218, 1883.

E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

R. McWeeny. *Symmetry: An Introduction to Group Theory and Its Applications*. Dover Publications, 2002.

H. T. M. N.P Van Der AA and R. Mattheij. Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem. *Electronic Journal of Linear Algebra*, 16:300–314, 2007.

M. Pitteri. Full kinematics of β-quartz in the view of its weak phase transformations. In *Notices of Universities. South of Russia. Natural sciences., 2nd special issue: Nonlinear problems of Continuum Mechanics*, pages 3145–3172. , 2003. Dedicated to 60th birthday of Professor L.M. Zubov.

H. Poincaré. Sur les courbes defiué par une équation differentielle, i-iv. *Oeuvres I. Gauthier-Villars, Paris*, 1881-1886.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++*. Cambridge University Press, 1988.

C. C. Pugh. *Real Mathematical Analysis*. Springer, 2000.

R. C. Robinson. *An Introduction to Dynamical Systems: Continuous and Discrete*. Pearson Prentice Hall, 2004.

T. Sendova, R. S. Elliott, and E. B. Tadmor. Non-uniquenes in energy minimization of atomistic problems: A branch-following and bifurcation investigation. , 2010. In preparation.

E. B. Tadmor and R. E. Miller. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. Cambridge University Press, 2010.

N. Triantafyllidis and R. Peek. On stability and the worst imperfection shape in solids with nearly simultaneous eigenmodes. *International Journal of Solids and Structures*, 29(18):2281–2299, 1992.

Appendix A

Symmetry of the square

This appendix details the steps to construct the m vectors that span the invariant subspace S_G in problems that possess the symmetry of a square. First, referring to Fig. (A.1) we obtain the square symmetry group $G = \{e, c_4, c_4^{-1}, c_2, \sigma_{x_1}, \sigma_{x_2}, \sigma_{13}, \sigma_{24}\}$, where the elements are given as follows: (1) e is the Identity element; (2) c_4 is the 90 degree counter-clockwise rotation about the x_3-axis; (3) c_4^{-1} is the 90 degree clockwise rotation about the x_3-axis; (4) c_2 is the 180 degree rotation about the x_3-axis; (5) σ_{x_1} is a mirror across the x_1-x_3 plane; (6) σ_{x_2} is a mirror across the x_2-x_3 plane; (7) σ_{13} is a mirror across the forward diagonal plane; and (8) σ_{24} is a mirror across the backward diagonal plane.

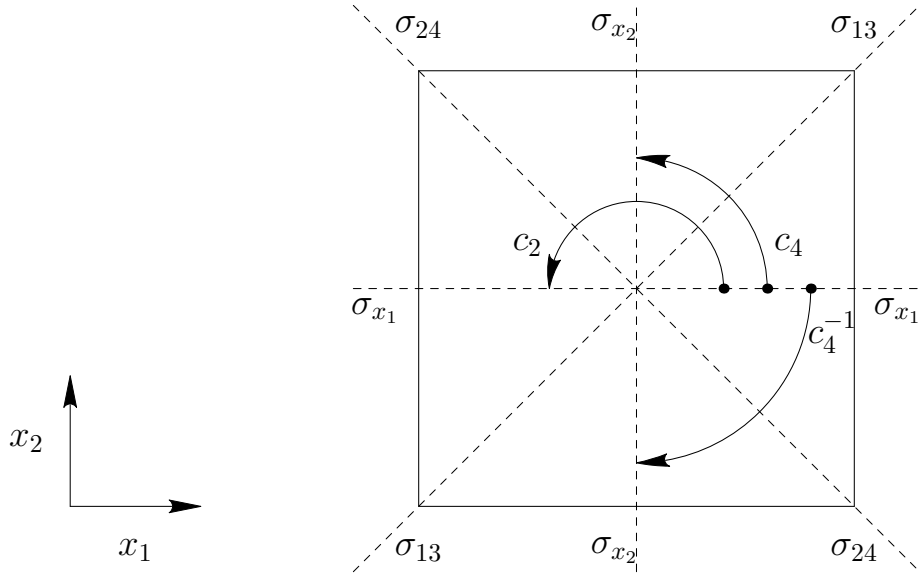

Figure A.1: Elements of Square symmetry group.

Given the degree of freedom vector $\mathbf{u} = [u_{11}, u_{12}, u_{21}, u_{22}, u_{31}, u_{32}, u_{41}, u_{42}]^\top \in \mathbb{R}^8$, which are the nodal displacements of the square spring structure in Chapter 2, we can obtain the matrix representation on $\mathbb{R}^8$ of the symmetry elements $\{\mathbf{T}_g | \ g \in G\}$. The following lists the matrix representations:

$$\mathbf{T}_e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{c_4} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{c_4^{-1}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{c_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\sigma_{x_1}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\sigma_{x_2}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\sigma_{13}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\sigma_{24}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Then, from Eq. (2.6), it is found that $m = 1$ and therefore $\mathbf{\Psi}_G$ consists of one vector $\mathbf{\Psi}$ such that $\mathbf{P}\mathbf{\Psi} = \mathbf{\Psi}$ where $\mathbf{P}$ is given by Eq. (2.8)

$$\mathbf{P} = \frac{1}{|G|}\sum_{g\in G}\mathbf{T}_g = \frac{1}{8}[\mathbf{T}_e + \mathbf{T}_{c_4} + \mathbf{T}_{c_4^{-1}} + \mathbf{T}_{c_2} + \mathbf{T}_{\sigma_{x_1}} + \mathbf{T}_{\sigma_{x_2}} + \mathbf{T}_{\sigma_{13}} + \mathbf{T}_{\sigma_{24}}]. \quad \text{(A.1)}$$

Applying the above formula to the matrix representation of the elements of the group of square symmetry, it is found that $\mathbf{P}$ is:

$$\mathbf{P} = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}. \quad \text{(A.2)}$$

Now, by solving the Eigenvalue problem $\mathbf{P}\boldsymbol{\Psi} = \boldsymbol{\Psi}$ for the m (in this case $m = 1$) eigenvectors with eigenvalue equal to one, that span the invariant subspace S_G, we find that $\boldsymbol{\Psi}_G = \frac{1}{\sqrt{8}}[1, 1, 1, -1, -1, -1, -1, 1]^T$. Having obtained $\boldsymbol{\Psi}_G$, we can now proceed to solve the reduced set of equations as detailed in Chapter 2.

Appendix B

One dimensional projection operator algorithm

In the one dimensional chain example detailed in Chapter 5, $\mathbf{T}$ is given by $\mathbf{T} = [0, t, t, \ldots, t]^T \in \mathbb{R}^{N+1}$ where $t = 1/\sqrt{N}$. Algorithm 10 gives the N sparse orthonormal vectors that span $\mathbb{R}^{N+1}$ and are mutually orthogonal to $\mathbf{T}$.

In the case of $N = 6$, the following projection operator $\mathbf{Q}$ is obtained:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{2} & \frac{1}{\sqrt{12}} \\ 0 & \frac{-1}{\sqrt{2}} & 0 & 0 & \frac{1}{2} & \frac{1}{\sqrt{12}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{2} & \frac{1}{\sqrt{12}} \\ 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{-1}{2} & \frac{1}{\sqrt{12}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{-2}{\sqrt{12}} \\ 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{-2}{\sqrt{12}} \end{bmatrix}$$

Algorithm 10 Projection Operator Algorithm for $\mathbf{Q} \in \mathbb{R}^N \otimes \mathbb{R}^{N+1}$

Input: N
$\mathbf{Q} := \mathbf{0}$;
$k := 0$;
$\mathbf{Q}[k][0] := 1$;
$k := k + 1$;
for $i := 0$ to $i < \lfloor \log_2(N) \rfloor$ **do**
  $m := \lfloor \frac{N}{2^{i+1}} \rfloor$;
  $R := N - 2^{i+1}m$;
  $q := 1$;
end for
for $j := 0$ to $j < m$ **do**
  $sum := 0$;
  for $p := 0$ to $p < 2^i$ **do**
    $\mathbf{Q}[k][q] := 1$;
    $sum := sum + 1$;
    $q := q + 1$;
  end for
  for $p := 0$ to $p < 2^i$ **do**
    $\mathbf{Q}[k][q] := -1$;
    $sum := sum + 1$;
    $q := q + 1$;
  end for
  $sum := \sqrt{sum}$;
  for $h := 0$ to $h < (N + 1)$ **do**
    $\mathbf{Q}[k][h] := \mathbf{Q}[k][h]/sum$;
  end for
  $k := k + 1$;
  if $j = m - 1$ and $R \geq 2^i$ **then**
    $sum := 0$;
    $r := 2^{i+1}m$;
    for $g := 0$ to $g < r$ **do**
      $\mathbf{Q}[k][g + 1] := 1$;
      $sum := sum + 1$;
    end for
    $y := -g/2^i$;
    for $t := 0$ to $t < 2^i$ **do**
      $\mathbf{Q}[k][g + 1] := y$;
      $g := g + 1$;
    end for
    for $h := 0$ to $h < (N + 1)$ **do**
      $\mathbf{Q}[k][h] := \mathbf{Q}[k][h]/y$;
    end for
    $k := k + 1$;
  end if
end for
Return $\mathbf{Q}$;

Appendix C

Implicit function theorem

In Chapter 3, the use of local continuation methods to get global solutions of the general problem $\mathbf{F}(\mathbf{u}, \lambda) = \mathbf{0}$ was discussed. The main tool that allows for this study is the implicit function theorem (IFT) which states that given appropriate conditions, the equation $\mathbf{F}(\mathbf{u}, \lambda) = \mathbf{0}$ defines a unique, locally continuous curve. This appendix reviews the basic assumptions and results of the theorem. The presentation here is presented according to *Keller* (1987). We will proceed towards the Implicit Function Theorem through incremental theorems and lemmas. First recall the following theorem:

Theorem 1. ***Banach Contraction Mapping Principle***.
Let B be a banach space and $\bar{\Omega} \subset B$ be a complete metric space. Let the contraction mapping $\mathbf{F} : B \to B$ satisfy the following conditions:

1. $\mathbf{F}(\bar{\Omega}) \subset \bar{\Omega}$
2. $d(\mathbf{F}(\mathbf{u}), \mathbf{F}(\mathbf{v})) \leq \Theta d(\mathbf{u}, \mathbf{v})$ *for some* $\Theta \in (0, 1)$, $\forall \mathbf{u}, \mathbf{v} \in \bar{\Omega}$,

where $d(f, g)$ is a metric defined on $\bar{\Omega}$.Then the equation $\mathbf{u} = \mathbf{F}(\mathbf{u})$ has one and only one solution $\mathbf{u}^* \in \bar{\Omega}$ where $\mathbf{u}^* = \lim_{n\to\infty} \mathbf{u}_n$ and the sequence $\{\mathbf{u}_n\}$ is such that $\mathbf{u}_{n+1} = \mathbf{F}(\mathbf{u}_n)$ for $n = 0, 1, 2, ...$ and for any arbitrary $\mathbf{u}_o \in \bar{\Omega}$. Furthermore, the convergence is such that $d(\mathbf{u}^*, \mathbf{u}_n) \leq \frac{\Theta^n}{1-\Theta} d(\mathbf{u}_o, \mathbf{F}(\mathbf{u}_o))$.

Proof. Let $\mathbf{u}_0 \in \bar{\Omega}$ be arbitrary. Define $\mathbf{u}_n = \mathbf{F}^n(\mathbf{u}_o)$. By condition 2 above, then $d(\mathbf{u}_n, \mathbf{u}_{n+1}) \leq \Theta d(\mathbf{u}_{n-1}, \mathbf{u}_n) \leq \Theta^2 d(\mathbf{u}_{n-2}, \mathbf{u}_{n-1}) \leq ... \leq \Theta^n d(\mathbf{u}_o, \mathbf{u}_1) \Rightarrow d(\mathbf{u}_n, \mathbf{u}_{n+1}) \leq \Theta^n d(\mathbf{u}_o, \mathbf{u}_1)$.

Therefore, this implies that $\forall \epsilon > 0$ there exists an $N \in \mathbb{N}$ such that

$$\Theta^N d(\mathbf{u}_o, \mathbf{u}_1) < (1-\Theta)\epsilon. \tag{C.1}$$

Furthermore, $\forall n, m > N$ then,

$$\begin{aligned} d(\mathbf{u}_m, \mathbf{u}_n) &\leq d(\mathbf{u}_m, \mathbf{u}_{m+1}) + d(\mathbf{u}_{m+1}, \mathbf{u}_{m+1}) + ... + d(\mathbf{u}_{n-1}, \mathbf{u}_n) \\ &\leq \Theta^m d(\mathbf{u}_0, \mathbf{u}_1) + \Theta^{m+1} d(\mathbf{u}_0, \mathbf{u}_1) + ... + \Theta^{n-1} d(\mathbf{u}_0, \mathbf{u}_1) \\ &\leq \Theta^m (1 + \Theta + ... + \Theta^{n-m-1}) d(\mathbf{u}_0, \mathbf{u}_1) \\ &\leq \Theta^N \left(\sum_{\ell=0}^{\infty} \Theta^\ell \right) d(\mathbf{u}_0, \mathbf{u}_1) = \tfrac{\Theta^N}{1-\Theta} d(\mathbf{u}_0, \mathbf{u}_1) < \epsilon. \end{aligned}$$

Therefore, $\forall n, m > N$, if $d(\mathbf{u}_m, \mathbf{u}_n) < \epsilon \Rightarrow \{\mathbf{u}_n\}$ is Cauchy and since $\bar{\Omega}$ is complete, then $\{\mathbf{u}_n\}$ converges to some $\mathbf{u}^* \in \bar{\Omega}$. It remains to show that $d(\mathbf{u}^*, \mathbf{F}(\mathbf{u}^*)) \to 0$ as $n \to 0$. This is easily seen since by the triangle inequality

$$\begin{aligned} d(\mathbf{u}^*, \mathbf{F}(\mathbf{u}^*)) &\leq d(\mathbf{u}^*, \mathbf{x}_n) + d(\mathbf{u}_n, \mathbf{F}(\mathbf{u}_n)) + d(\mathbf{F}(\mathbf{u}_n), \mathbf{F}(\mathbf{u}^*)) \\ &\leq d(\mathbf{u}^*, \mathbf{u}_n) + \Theta^n d(\mathbf{u}_0, \mathbf{u}_1) + \Theta d(\mathbf{u}_n, \mathbf{u}^*) \to 0 \text{ as } n \to \infty. \end{aligned}$$

Therefore, since $d(\mathbf{u}^*, \mathbf{F}(\mathbf{u}^*))$ is independent of n, $d(\mathbf{u}^*, \mathbf{F}(\mathbf{u}^*)) = 0$ and $\mathbf{u}^* = \mathbf{F}(\mathbf{u}^*)$. $\square$

Now, we will state a lemma which shows how to find a set $\bar{\Omega}$ such that the conditions in the Banach Contraction Principle are satisfied.

Lemma 1. ***Contracting Ball Lemma***.
Let $\rho > 0$, $\Theta \in (0,1)$ be such that for some $\mathbf{u}_o \in B$, $\mathbf{F} : B \to B$ satisfies

1. $d(\mathbf{u}_o, \mathbf{F}(\mathbf{u}_o)) \leq (1-\Theta)\rho$
2. $d(\mathbf{F}(\mathbf{u}), \mathbf{F}(\mathbf{v})) \leq \Theta d(\mathbf{u}, \mathbf{v}), \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{B}_\rho(\mathbf{u}_o)$ *where* $\mathbb{B}_\rho(\mathbf{u}_o) = \{\mathbf{u} \in B : d(\mathbf{u}, \mathbf{u}_o) \leq \rho\}$

Then, $\mathbf{F} : \mathbb{B}_\rho(\mathbf{u}_o) \to \mathbb{B}_\rho(\mathbf{u}_o)$ has a fixed point $\mathbf{u}^* \in \mathbb{B}_\rho(\mathbf{u}_o)$.

Proof. Note that condition (2) in Theorem 1 is identically satisfied by the assumptions in this lemma, therefore it remains to prove condition (1) of Theorem 1. That is, we need to show that $\mathbf{F}(\bar{\Omega}) \subset \bar{\Omega}$. This is easily seen, since for any $\mathbf{u} \in \mathbb{B}_\rho(\mathbf{u}_o)$, by the triangle inequality,

$$d(\mathbf{F}(\mathbf{u}), \mathbf{u}_o) \leq d(\mathbf{F}(\mathbf{u}), \mathbf{F}(\mathbf{u}_o)) + d(\mathbf{F}(\mathbf{u}_o), \mathbf{u}_o) \leq \Theta\rho + (1-\Theta)\rho = \rho.$$

Therefore, $d(\mathbf{F}(\mathbf{u}), \mathbf{u}_o) \leq \rho$ which means that if $\mathbf{u} \in \mathbb{B}_\rho(\mathbf{u}_o)$, then $\mathbf{F}(\mathbf{u}) \in \mathbb{B}_\rho(\mathbf{u}_o)$ which proves condition (1) of Theorem 1. Since conditions (1) and (2) of Theorem 1 are satisfied, we have identified $\bar{\Omega} = \mathbb{B}_\rho(\mathbf{u}_o)$ such that there exists a fixed point $\mathbf{u}^* \in \mathbb{B}_\rho(\mathbf{u}_o)$. □

The Implicit Function Theorem is discussed next.

Theorem 2. ***Implicit Function Theorem.***
Let $B_1 = \mathbb{R}^n$ and $B_2 = \mathbb{R}$ be a Banach space and parameter space, respectively.
Let $\mathbf{F} : B_1 \times B_2 \to B_1$, satisfy, for some for some $\rho_1 > 0$, and $\rho_2 > 0$ (sufficiently small), the following conditions:

1. $\mathbf{F} \in C^r$, $1 \leq r \leq \infty$
2. $\mathbf{F}(\mathbf{u}_o, \lambda_o) = \mathbf{0}$ *for some* $\mathbf{x}_o \in B_1, \lambda_o \in B_2$
3. $\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_o, \lambda_o)$ *is non-singular and is bounded (i.e.* $||\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_o, \lambda_o)||^{-1} \leq C$ *for some* $C \in \mathbb{R}$*)*
4. $\mathbf{F}(\mathbf{u}_o, \lambda_o)$ *and* $\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_o, \lambda_o)$ *are continuous on* $\mathbb{B}_{\rho_1}(\mathbf{u}_o) \times \mathbb{B}_{\rho_2}(\lambda_o)$ *(where* $\mathbb{B}_{\rho_1}(\mathbf{u}_o) = \{\mathbf{u} \in B_1 : ||\mathbf{u} - \mathbf{u}_o|| < \rho_1 \}$ *and* $\mathbb{B}_{\rho_2}(\lambda_o) = \{\lambda \in B_2 : ||\lambda - \lambda_o|| < \rho_2 \}$*).*

Then, for all $\lambda \in \mathbb{B}_{\rho_2}(\lambda_o)$, there exists a $\mathbf{u}(\lambda) \in \mathbb{B}_{\rho_1}(\mathbf{u}_o)$ such that:

1. $\mathbf{u}(\lambda_o) = \mathbf{u}_o$
2. $\mathbf{F}(\mathbf{u}(\lambda), \lambda) = \mathbf{0}$
3. *For a given $\lambda \in \mathbb{B}_{\rho_2}$ the solution to $\mathbf{F}(\mathbf{u}(\lambda), \lambda) = \mathbf{0}$ is unique*
4. *The solution $\mathbf{u}(\lambda)$ depends continuously on λ.*

Proof. Let $(\mathbf{u}_0, \lambda_0) \in \mathbb{R}^n \times \mathbb{R}$ such that $\mathbf{F}(\mathbf{u}_0, \lambda_0)$ be arbitrary. By condition (3), since $\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0, \lambda_0)$ is nonsingular, it follows that:

$$\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0, \lambda_0)\mathbf{u} = \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0, \lambda_0)\mathbf{u} - \mathbf{F}(\mathbf{u}, \lambda) \Leftrightarrow \mathbf{F}(\mathbf{u}, \lambda) = \mathbf{0}. \tag{C.2}$$

Therefore, $\mathbf{F}(\mathbf{u}, \lambda) = \mathbf{0}$ is equivalent to

$$\mathbf{u} = \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0, \lambda_0)^{-1}[\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0, \lambda_0) - \mathbf{F}(\mathbf{u}, \lambda)]. \tag{C.3}$$

Defining $\mathbf{G}(\mathbf{u},\lambda) \equiv \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0,\lambda_0)^{-1}[\nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0,\lambda_0) - \mathbf{F}(\mathbf{u},\lambda)]$, the problem of solving $\mathbf{F}(\mathbf{u},\lambda) = \mathbf{0}$ reduces to finding the fixed point of $\mathbf{G}(\mathbf{u},\lambda)$ for a given λ.

To use the contraction mapping theorem, the conditions for the contracting ball lemma must first be verified. Choose any $\lambda \in \mathbb{B}_{\rho_2}(\lambda_0)$ and use $\mathbf{G}(\mathbf{u}_0,\lambda_0) = \mathbf{u}_0$ (since $\mathbf{u}_0$ is a fixed point for $\lambda = \lambda_0$) to get the following:

$$\begin{aligned} \parallel \mathbf{u}_0 - \mathbf{G}(\mathbf{u}_0,\lambda) \parallel &= \parallel \mathbf{G}(\mathbf{u}_0,\lambda_0) - \mathbf{G}(\mathbf{u}_0,\lambda) \parallel \\ &\leq M_0 \parallel \mathbf{F}(\mathbf{u}_0,\lambda_0) - \mathbf{F}(\mathbf{u}_0,\lambda) \parallel \\ &\leq M_0\omega_0|\lambda_0 - \lambda| \\ &\leq M_0\omega_0(\rho_2), \end{aligned}$$

where $\omega_0(\rho) = \sup_{\{\lambda,\tilde{\lambda}\in\mathbb{B}_{\rho_2}(\lambda_0);\mathbf{u},\mathbf{v}\in\mathbb{B}_{\rho_1}(\mathbf{u}_0);\|\lambda-\tilde{\lambda}\|\leq\rho\}} \parallel \mathbf{F}(\mathbf{u},\lambda) - \mathbf{F}(\mathbf{v},\tilde{\lambda}) \parallel$.

It can be shown that ω_0 is nonnegative, nondecreasing and that $\omega_0 \to 0$ as $\rho \to 0$ by condition (4). Therefore,

$$\parallel \mathbf{u}_0 - \mathbf{G}(\mathbf{u}_0,\lambda) \parallel \leq M_0\omega_0(\rho_2). \tag{C.4}$$

Similarly, for $\mathbf{u},\mathbf{v} \in \mathbb{B}_{\rho_1}$, it can be shown that $\parallel \mathbf{G}(\mathbf{u},\lambda) - \mathbf{G}(\mathbf{v},\lambda) \parallel \leq M_0(\omega_1(\rho_1) + \omega_2(\rho_2)) \parallel \mathbf{u}-\mathbf{v} \parallel$, where $\omega_1(\rho_1) = \sup_{\{\lambda\in\mathbb{B}_{\rho_2}(\lambda_0);\mathbf{w}\in\mathbb{B}_{\rho_1}(\mathbf{u}_0)\}} \parallel \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u}_0,\lambda) - \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{w},\lambda) \parallel$ and $\omega_2(\rho_2) = \sup_{\{\mathbf{u},\mathbf{v}\in\mathbb{B}_{\rho_1}(\mathbf{u}_0);\lambda,\bar{\lambda}\in\mathbb{B}_{\rho_2}(\lambda_0);|\lambda-\bar{\lambda}|\leq\rho_2\}} \parallel \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{u},\lambda) - \nabla_{\mathbf{u}}\mathbf{F}(\mathbf{v},\bar{\lambda}) \parallel$.Therefore,

$$\parallel \mathbf{G}(\mathbf{u},\lambda) - \mathbf{G}(\mathbf{v},\lambda) \parallel \leq M_0(\omega_1(\rho_1) + \omega_2(\rho_2)) \parallel \mathbf{u}-\mathbf{v} \parallel \tag{C.5}$$

For any fixed $\theta \in (0,1)$, we can choose ρ_1, ρ_2 such that $M_0(\omega_1(\rho_1)+\omega_2(\rho_2) \leq \theta < 1$. From here, fix ρ_1 and reduce ρ_2 such that $M_0\omega_0(\rho_2) \leq (1-\theta)\rho_1$. Then, Eq. (C.4) and Eq. (C.5) shows that the conditions for the contracting ball lemma has been satisfied for some ρ_1 and ρ_2. Thefore, we can apply the contraction mapping theorem which proves the results (1), (2), and (3) of the implicit function theorem. To prove result (4), for $\lambda, \bar{\lambda} \in \mathbb{B}_{\rho_2}(\lambda_0)$, it can be shown that $\parallel \mathbf{u}(\lambda) - \mathbf{u}(\bar{\lambda}) \parallel \leq M_0\omega_0|\lambda-\bar{\lambda}|/(1-\theta)$ which shows that $\mathbf{u}(\lambda)$ is continuous in $\mathbb{B}_{\rho_2}(\lambda_0)$ (with ω_0 as defined as before). □