Cornell Theory Center Technical Report CTC96TR264

Symmetry, Nonlinear Bifurcation Analysis, and Parallel Computation*

J.C. Wohlever

Advanced Computing Research Institute
Cornell Theory Center
Cornell University, Ithaca, NY 14853.

Abstract

In the natural and engineering sciences the equations which model physical systems with symmetry often exhibit an invariance with respect to a particular group G of linear transformations. G is typically a linear representation of a symmetry group $\mathcal{G}$ which characterizes the symmetry of the physical system. In this work, we will discuss the natural parallelism which arises while seeking families of solutions to a specific class of nonlinear vector equations which display a special type of group invariance, referred to as equivariance. The inherent parallelism stems from a *global* de-coupling, due to symmetry, of the full nonlinear equations which effectively splits the original problem into a set of smaller problems. Numerical results from a symmetry-adapted numerical procedure, (MMcontcm.m), written in MultiMATLAB[1] are discussed.

1 Introduction

Consider the task of finding solutions to the following vector equilibrium equation

$$\begin{aligned} \mathbf{f}(\mathbf{u},\lambda) &= \mathbf{0}, \\ \mathbf{f}:\mathbb{R}^n\times\mathbb{R} &\mapsto \mathbb{R}^n. \end{aligned} \tag{1}$$

In eq. (1) $\mathbf{u}\in\mathbb{R}^n$, $\lambda\in\mathbb{R}$ is a free parameter and $\mathbf{f}\in C^m, m\geq 3$. The task of computing the solution paths of eq. (1) falls under the framework of the well developed field of nonlinear bifurcation analysis and continuation [1]-[4].

* This research was partially supported by the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

[1]MultiMATLAB is an ongoing research project at the Cornell Theory Center (CTC) whose goal is to provide a portable programming tool which will enable users to run MATLAB on multiple processors. For more information see *http://www.cs.cornell.edu/Info/People/lnt/multimatlab.html.* MATLAB is a registered trademark of The MathWorks, Inc.

If eq. (1) is an equilibrium equation for a physical system with symmetry, it is well known that the symmetry will be reflected in (1). It is further well known that group theoretic techniques can greatly facilitate the nonlinear bifurcation analysis of eq. (1). Over the past fifteen years, there has been a quite a bit of activity within the mechanics and mathematical community on the application of group theoretic methods to aid in the computation of the *global* bifurcation diagrams of symmetric structures [1],[5]-[27]. Within the context of a numerical continuation procedure, group theory helps one systematically find an "optimal" set of basis vectors which reflect the symmetry of a given problem. The immediate payoff in formulating the numerical procedure with respect to the symmetry-adapted basis is a global de-coupling of the equilibrium equations which in turn leads to: (1) a dimensional reduction in the problem size; (2) improved numerical conditioning while computing solutions in the vicinity of singular points; (3) a systematic method for detecting and diagnosing for symmetry-breaking bifurcations. Thus, group theory, or more specifically group representation theory [28]-[31], provides the tools for a well-conditioned and efficient numerical algorithm for computing the global bifurcation diagram of symmetric problems by taking "optimal" advantage of the inherent symmetry.

In the field of nonlinear structural mechanics, a group theoretic approach to the buckling and post-buckling analysis of *imperfection-sensitive* shell structures has proved to be an invaluable tool. As discussed in [24] and [32], it is often the case that the global solution curves for imperfection sensitive shell structures are riddled with closely-packed singular points. The existence of these singular points, coupled with the need for large degree-of-freedom systems, render standard numerical continuation routines extremely ill-conditioned, *e.g.,* [33]-[37]. As a result of the numerical ill-conditioning, computing the global response of these types of structures requires a group theoretic approach.

From a safety point-of-view, being able to accurately compute the buckling and post-buckling response of shell structures is critical in predicting the ultimate load-carrying capacity of structures and structural components in aerospace, civil, mechanical and nuclear engineering — it also has applications in others fields such as understanding the phenomena of fabric drape in textiles [38] and predicting the equilibrium shapes of biological structures such as red blood cells [39]. However, as noted in [32] even with the numerical stability and reduction in problem size offered by a group theoretic approach, the extremely rich global behavior of imperfection-sensitive shell structures necessitates the use of parallel computers to make a complete global analysis computationally feasible.

In this work, it will be discussed how the global de-coupling of nonlinear equilibrium equations sets the stage for a conceptually straight forward implementation in a *distributed-memory* MIMD (multiple instruction multiple data) computational environment. It will further be discussed how the *a priori* information provided by a group analysis aids in developing a coarse-grained load-balanced parallel algorithm. The goal of this work is to demonstrate an efficient and systematic parallel approach to an important class of nonlinear equilibrium equations in engineering and science.

The outline of this paper is as follows. In §2, a brief review of a computational approach to the nonlinear continuation and bifurcation analysis of one-parameter vector equilibrium equations of the form shown in eq. (1) is given. The purpose of §2 is to familiarize the reader with some basic concepts and terminology from nonlinear bifurcation analysis. An introduction to a group theoretic approach to nonlinear bifurcation analysis is given in §3. After some introductory remarks on group theory, it is discussed how group theoretic techniques may be used to construct a dimensionally-reduced problem based upon the symmetry of a given system. This reduced problem captures *exact* solutions of the full problem of a specific symmetry type. The discussion then turns to describing how results from group representation theory can be used to partition the solution space into a set of mutually orthogonal subspaces. This global partitioning of the solution space is the key to a systematic and efficient diagnosis of symmetry breaking bifurcations. §4 outlines a general group theoretic continuation/bifurcation algorithm. In §4 it is pointed out how the global de-coupling of the equilibrium equations leads to a coarse-grained multi-processor algorithm for the nonlinear bifurcation analysis of problems with symmetry. The pseudo code for two MATLAB-based symmetry-adapted nonlinear continuation/bifurcation codes contcm, (single-processor code), and MMcontcm, (multi-processor code), are presented. Numerical results and a discussion about different aspects of MMcontcm are presented in §5. It it shown how the *a priori* information about the structure of the symmetry-adapted equations provides the analyst with valuable information which allows for a load-balanced procedure. We end the paper in §6 with a brief summary of the important points of this work and discuss some possible directions for future research.

2 Nonlinear Bifurcation Analysis and Continuation

A *solution point* of eq. (1) is a point $(\mathbf{u}^*, \lambda^*) \in V \times \mathbb{R}$ such that $\mathbf{f}(\mathbf{u}^*, \lambda^*) \;=\; \mathbf{0}$. The *solution set* of eq. (1) is the set of all solution points

$$\{(\mathbf{u}^*, \lambda^*) \in V \times \mathbb{R} : \mathbf{f}\,(\mathbf{u}^*, \lambda^*) \;=\; \mathbf{0}\}\,. \tag{2}$$

With a single free parameter λ, the solution set of eq. (1) generically consists of a set of 1-dimensional curves (manifolds) contained within the $(n+1)$-dimensional space $V \times \mathbb{R}$. Thus one may view the solution set of eq. (1) to be a collection of smooth *solution branches(paths)* $\mathbf{x}(s)$,

$$\mathbf{x}(s) \;=\; [\mathbf{u}(s), \lambda(s)] \quad s_1 \le s \le s_2 \tag{3}$$

parameterized by an (as of yet) unspecified path parameter s.

In this work it is assumed that $(\mathbf{u}, \lambda) = (\mathbf{0}, 0)$ is a solution point and the solution branch which contains $(\mathbf{0}, 0)$ is referred to as the *primary solution branch.* At certain points in the solution space $V \times \mathbb{R}$ two or more distinct solution branches may intersect one another. These points of intersection reveal a local loss of uniqueness in the solutions to eq. (1) and are called *bifurcation points.* A point which is common to two distinct transverse solution branches is referred to as a *simple bifurcation point* while a point which is shared by three or more distinct solution branches is known as a *multiple bifurcation point.* A solution branch which intersects the primary solution path is called a *primary bifurcation path.* Similarly, a solution branch which intersects a primary bifurcation path is referred to as a *secondary bifurcation path. Tertiary, quaternary* and higher order branches are defined analogously. A typical bifurcation diagram is shown in Fig. 1. Since a solution branch $\mathbf{x}(s)$ is contained in an $(n+1)$-dimensional vector space Fig. 1 necessarily depicts a projection of $\mathbf{x}(s)$ onto a two-dimensional space where $||\mathbf{u}||$ is some appropriately defined measure of $\mathbf{u}$.

For problems of the form described in eq. (1) it is not difficult to derive a necessary condition for bifurcation to occur. Consider the primary solution branch $\mathbf{x}(s)$ shown in Fig. 1. Since $\mathbf{f}(\mathbf{u}, \lambda) = \mathbf{0}$ everywhere on a solution branch differentiating eq. (1) with respect to s along $\mathbf{x}(s)$ gives

$$\frac{d}{ds}\mathbf{f}\,[\mathbf{u}(s), \lambda(s)] \;=\; \mathbf{J}\dot{\mathbf{u}}(s) + \mathbf{f}_\lambda \dot{\lambda}(s) \;=\; \mathbf{0}. \tag{4}$$

The Jacobian $\mathbf{J} = \mathbf{f_u} = \frac{d\mathbf{f}}{d\mathbf{u}}$ is the Frechet derivative of $\mathbf{f}$ with respect to $\mathbf{u}$, $\dot{()} \equiv \frac{d}{ds}$ and $\mathbf{f}_\lambda = \frac{d\mathbf{f}}{d\lambda}$. In this work the notation $\mathbf{f_u}$ and $\mathbf{J}$ are used interchangeably.

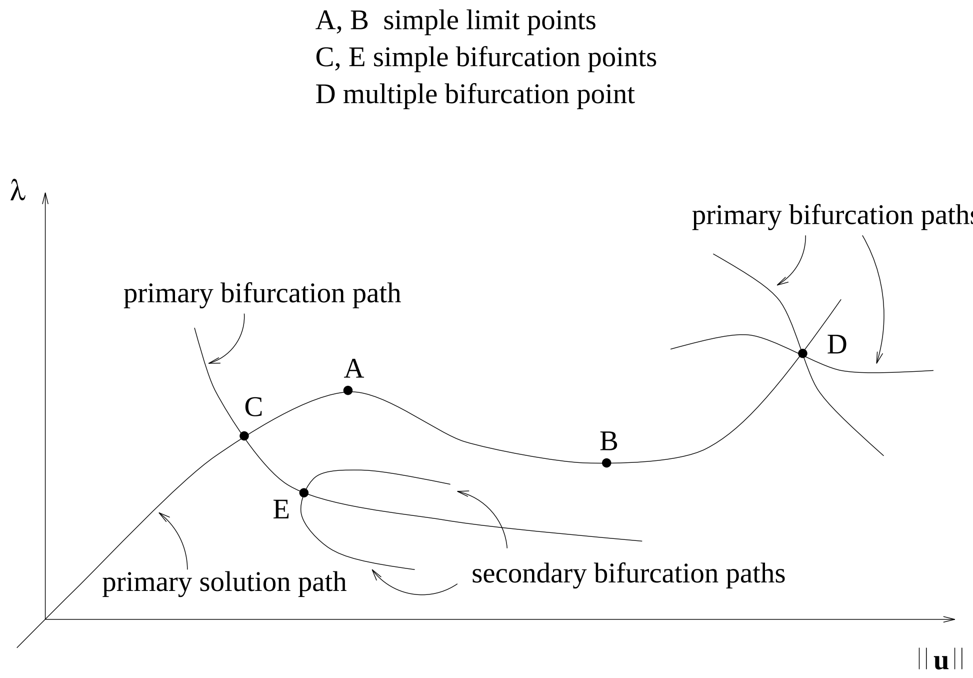

Figure 1: Typical Bifurcation Diagram.

Suppose $(\mathbf{u}_o, \lambda_o) = [\mathbf{u}(s_o), \lambda(s_o)]$ is a solution point to eq. (1). If $\mathbf{J}_o \equiv \mathbf{J}(\mathbf{u}_o, \lambda_o)$ is invertible, then $(\mathbf{u}_o, \lambda_o)$ is called a *regular solution point.* If $\mathbf{J}_o$ is not invertible, $(\mathbf{u}_o, \lambda_o)$ is referred to as a *singular point.* It is well known that at a regular solution point, the solution to eq. (4) uniquely defines a tangent vector to the solution curve at $(\mathbf{u}_o, \lambda_o)$ and the Implicit Function Theorem (IFT) [2] guarantees the existence of a unique solution $[\mathbf{u}(s), \lambda(s)]$ in a neighborhood of s_o,

$$s \in [s_o - \epsilon, s_o + \epsilon], \quad \epsilon > 0.$$

At a singular point the IFT is not valid. Therefore, the occurrence of a singular point is a necessary, though not sufficient, condition for bifurcation to occur. An important example of when a singular $\mathbf{J}$ does not lead to a bifurcation is at a simple limit point (points A and B in Fig. 1). A point $(\mathbf{u}_o, \lambda_o)$ is a *simple (normal) limit point* of eq. (1) if $\dim \mathrm{Null}\,(\mathbf{J}_o) = \dim \mathrm{Null}\left(\mathbf{J}_o^T\right) = 1$ and $\mathbf{f}_\lambda^o \equiv \mathbf{f}_\lambda(\mathbf{u}_o, \lambda_o) \notin \mathrm{Range}\,(\mathbf{J}_o)$.

Computing the solution set of eq. (1) is the fundamental goal of a continuation/bifurcation procedure. To carry out this task, a practical procedure must be able to do the following:

- compute solution branches through both regular points as well as singular points;
- detect and diagnose bifurcation points;
- switch branches at bifurcation points.

2.1 Continuation

In practice it is rare that an analytical solution to eq. (1) can be found, although it is commonly assumed that at least one solution point $(\mathbf{u}_o, \lambda_o)$ is known *a priori.* Due to the difficulty in finding analytical solutions, a great deal of work in the technical literature has been directed towards developing efficient and accurate numerical continuation schemes to compute approximate solutions to (1). A numerical *continuation scheme* is an algorithm which begins with a known solution point $(\mathbf{u}_o, \lambda_o)$ and proceeds to compute discrete points along the solution branch which contains $(\mathbf{u}_o, \lambda_o)$. The two basic steps of a continuation scheme are:

1. A predictor step to generate an initial guess of a new solution point $\mathbf{x}_{k+1} = (\mathbf{u}_{k+1}, \lambda_{k+1})$ near the last converged solution point $\mathbf{x}_k$. Two common techniques for computing a prediction step are Euler's method (a first order Taylor expansion) to compute an approximate tangent to the solution curve or by using a polynomial interpolation from previous solution points. A popular prediction scheme in the structural analysis community is a three-point (quadratic) Lagrangian interpolation formula,

$$\mathbf{x}_{k+1} = N_1(\Delta s_{k+1})\mathbf{x}_{k-2} + N_2(\Delta s_{k+1})\mathbf{x}_{k-1} + N_3(\Delta s_{k+1})\mathbf{x}_k$$

   where N_1, N_2 and N_3 are quadratic Lagrangian polynomials. In practice, it has been found that a quadratic predictor is fairly robust and is a good compromise between stability and economy. The quantity Δs_{k+1} is defined as the *step length*, which is a measure of how "far" $\mathbf{x}_{k+1}$ will be from $\mathbf{x}_k$.

2. The second step of a continuation scheme is an iterative correction algorithm designed to successively update and improve the initial guess until a convergence criteria is met. Standard iterative correction schemes generally fall under the category of either Newton's method or a Secant method, *e.g.,* Broyden updates [40].

With regard to generating an initial prediction step, an issue of some importance is controlling the size of the step length Δs_{k+1}. Since the convergence characteristics of the correction procedure varies along different parts of a solution curve it is important to have an adaptive step length control strategy to ensure reasonable performance of the continuation procedure. For example, in regions of strong curvature or near singular points of interest, relatively small step sizes may be necessary to accurately resolve details of the solution branch and to ensure convergence of the correction scheme in a reasonable number of iterations. In other less interesting parts of the solution curve, it may be more prudent from a computational expense point of view to use larger step sizes. Many step length control procedures have been developed and choosing one is dependent on the type of problems being studied. Two common design goals for developing step length control procedures are choosing Δs to maintain a certain user-defined *target* iteration count and to control the maximum angle between any two successive prediction steps, which is important in regions of strong curvature. For more information on step length control, the interested reader is referred to [3], [41] and [42].

The capabilities of any practical continuation scheme must include the ability to compute solution branches through singular points. One of the inherent difficulties of computing a solution branch through a singular point lies in the fact that in the vicinity of a singular point the Jacobian matrix $\mathbf{J}$ becomes ill-conditioned. Furthermore, at a bifurcation point the continuation scheme must also be able to contend with multiple solution branches and pick out the desired path. The above notwithstanding, under "mild smoothness conditions" [2] a given continuation procedure can jump over bifurcation points. On the other hand, a continuation algorithm might not be able to get past a simple limit point unless special care is taken in choosing an appropriate parametrization for the solution curve [2], [3], [43].

The importance of limit points[2] has prompted numerous investigations aimed at overcoming the difficulties associated with computing a solution path through a limit point. Perhaps the most popular solution algorithm with this capability is the *pseudo arc-length* continuation scheme [1]-[4]. In an pseudo arc-length continuation procedure, an iterative nonlinear equation solver is applied to an extended system of equations of the form

$$\mathbf{F}\left[\mathbf{x}(s), s\right] \;=\; \left[\begin{array}{c} \mathbf{f}[\mathbf{u}(s), \lambda(s)] \\ N[\mathbf{u}(s), \lambda(s), s] \end{array}\right] \;=\; \mathbf{0} \tag{5}$$

where $\mathbf{F} : V \times \mathbb{R} \times \mathbb{R} \mapsto V \times \mathbb{R}$. $N[\mathbf{u}(s), \lambda(s), s] = 0$ is a scalar constraint equation which defines the path parameter s in terms of unknowns of the problem. The objective of the constraint equation is to maintain a well defined parametrization for the solution path.

A simple geometrical interpretation of the constraint equation, as discussed in [3], can be had if we assume that N is of the form

$$N[\mathbf{u}(s), \lambda(s), s] \;=\; G[\mathbf{u}(s), \lambda(s)] - s \;=\; G[\mathbf{x}(s)] - s = 0. \tag{6}$$

Since $G[\mathbf{x}(s)] - s$ equals zero every along a solution curve, differentiating G with respect to s gives

$$\frac{d}{ds}\left(G[\mathbf{x}(s)] - s\right) = \underbrace{\frac{dG}{d\mathbf{x}}}_{\mathbf{n}} \cdot \underbrace{\frac{d\mathbf{x}}{d\sigma}}_{\mathbf{x}'} \frac{d\sigma}{ds} - 1 = \left(\mathbf{n} \cdot \mathbf{x}'\right) \frac{d\sigma}{ds} - 1 = 0 \tag{7}$$

where σ is the true arc-length from differential geometry, (See [45] p.245). By definition $\frac{d\mathbf{x}}{d\sigma} = \mathbf{x}'$ is the unit tangent along the solution curve and $G_{\mathbf{x}} = \mathbf{n}$ is the normal to the hypersurface defined by eq. (6), (Fig. 2). An "allowable" path parameter s, ([46], pp.17-29), must locally satisfy the condition $\frac{ds}{d\sigma} \neq 0$, which according to eq. (7) is equivalent to saying that $\mathbf{n} \cdot \mathbf{x}' \neq 0$. Therefore, to maintain a well defined parametrization for the solution path the constraint equation $N[\mathbf{u}(s), \lambda(s), s] = 0$ must define an adaptive hypersurface whose normal at the point of intersection with the solution curve $\mathbf{x}(s)$ is never orthogonal to $\mathbf{x}'$.

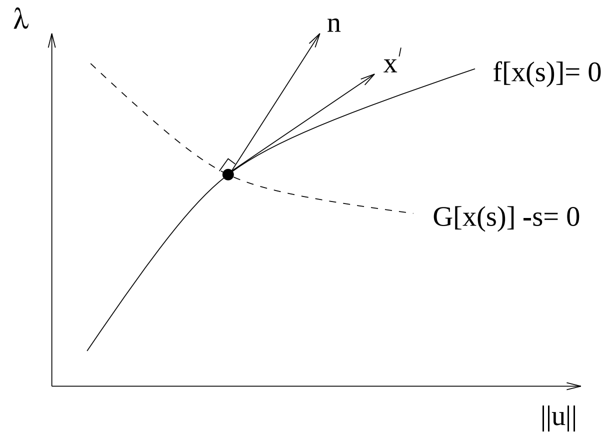

Figure 2: Relationship between a solution path and the constraint equation $G[\mathbf{x}(s)] - s = 0$.

[2] In structural stability problems, the existence of limit points often indicates the occurrence of a snap-through phenomena [44].

From (5) it can be shown that along a solution branch

$$\frac{d}{ds}\mathbf{F}\left[\mathbf{x}(s), s\right] \;=\; \underbrace{\begin{bmatrix} \mathbf{J} & \mathbf{f}_\lambda \\ \frac{\partial N}{\partial \mathbf{u}} & \frac{\partial N}{\partial \lambda} \end{bmatrix}}_{\mathbf{F_x}} \underbrace{\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\lambda} \end{bmatrix}}_{\dot{\mathbf{x}}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \frac{\partial N}{\partial s} \end{bmatrix}}_{\mathbf{F}_s} \;=\; \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \tag{8}$$

The salient feature of eq. (5) is that with a properly chosen constraint equation $N(\mathbf{u}, \lambda, s) = 0$, the Jacobian of the extended system $\mathbf{F_x}$ is of full rank at a simple limit point even though $\mathbf{J}$ has a null space of dimension one. Solving (8) for $\dot{\mathbf{x}}$ defines the tangent vector to a solution branch $\mathbf{x}(s)$. A particularly useful definition for the constraint equation is [43]

$$\hat{N}[\mathbf{u}(s), \lambda(s), s] \;=\; \dot{\mathbf{u}}_o \cdot [\mathbf{u}(s) - \mathbf{u}_o] + \dot{\lambda}_o \cdot [\lambda(s) - \lambda_o] - (s - s_o) \;=\; 0. \tag{9}$$

By enforcing the path constraint displayed in eq. (9), the tangent vector $\dot{\mathbf{x}}_o$ is constrained such that $\| \dot{\mathbf{x}}_o \| \approx 1$.

2.1.1 Bordering Algorithm: A Practical Path Following Method

Beginning with equations (5) and (9), the following "Bordering Algorithm" (Fig. 3), has been developed as a practical and efficient continuation procedure which can effectively compute a solution branch through a limit point [1], [43].

Bordering Algorithm Assuming a solution point $[\mathbf{u}_{k-1}, \lambda_{k-1}]$ is known, a new solution point $[\mathbf{u}_k, \lambda_k]$ located by $s_k = s_{k-1} + \Delta s_k$ is sought where Δs_k is a sufficiently small step length.

1. Define an initial guess for $(\mathbf{u}_k, \lambda_k)$
2. Refine the initial guess by successive updates $(\mathbf{u}_k, \lambda_k) \;=\; (\mathbf{u}_k + \Delta\mathbf{u}_k, \lambda_k + \Delta\lambda_k)$ where $\Delta\mathbf{u}_k$ and $\Delta\lambda_k$ are defined as

$$\Delta\lambda_k = \frac{\widehat{N}_k - \dot{\mathbf{u}}_{k-1} \cdot \mathbf{y}_k}{\dot{\mathbf{u}}_{k-1} \cdot \mathbf{z}_k - \dot{\lambda}_o}, \qquad \Delta\mathbf{u}_k = -\mathbf{y}_k - \mathbf{z}_k \Delta\lambda_k, \tag{10}$$

and

$$\widehat{N}_k \equiv \dot{\mathbf{u}}_{k-1} \cdot [\mathbf{u}_k - \mathbf{u}_{k-1}] + \dot{\lambda}_{k-1} \cdot [\lambda_k - \lambda_{k-1}] - \Delta s.$$

The vectors $\mathbf{y}_k$ and $\mathbf{z}_k$ are the respective solutions to the so-called *incremental equations*

$$\mathbf{J}(\mathbf{u}_k, \lambda_n)\mathbf{y}_k = \mathbf{f}(\mathbf{u}_k, \lambda_k) \;\text{ and }\; \mathbf{J}(\mathbf{u}_k, \lambda_k)\mathbf{z}_k = \mathbf{f}_\lambda(\mathbf{u}_k, \lambda_k) \tag{11}$$

and the derivatives $\dot{\mathbf{u}}_{k-1}$ and $\dot{\lambda}_{k-1}$ are approximated as

$$\dot{\mathbf{u}}_{k-1} \approx \frac{\mathbf{u}_{k-1} - \mathbf{u}_{k-2}}{\Delta s_{k-1}} \;\text{ and }\; \dot{\lambda}_{k-1} \approx \frac{\lambda_{k-1} - \lambda_{k-2}}{\Delta s_{k-1}}. \tag{12}$$

Continue step 2 until a convergence criteria, *e.g.*, $\| \mathbf{F}(\mathbf{x}_k, s_k) \| < \epsilon$, is satisfied.

3. Once a new solution point $(\mathbf{u}_k, \lambda_k, s_k)$ is located, the above algorithm may be repeated to search for the next solution point or the procedure may be terminated.

2.2 Bifurcation Detection and Branch Switching

As a solution branch is being computed, the continuation algorithm should be checking for the occurrence of singular points. In general, detecting a singular point amounts to noting the occurrence of a zero eigenvalue in the Jacobian matrix $\mathbf{J}$. Common methods of detecting a zero eigenvalue in numerical computation include: (1) a partial spectral analysis of $\mathbf{J}$; (2) looking for a sign change in the determinant of $\mathbf{J}$; (3) counting the number of negative entries on the diagonal of the matrix U from an LU decomposition of $\mathbf{J}$ [47].

Once a singular point is detected, it is necessary to determine whether or not a bifurcation has occurred. This refers to the previous note that the existence of a singular point is a necessary but not sufficient condition for bifurcation to occur. A well known statement and proof of sufficiency, taken from [2], is included here for completeness. Assume that $(\mathbf{u}_o, \lambda_o)$ is a singular solution point of (1) along a smooth solution branch such that $\mathbf{u}_o = \mathbf{u}(s_o)$, $\lambda_o = \lambda(s_o)$,

$$\text{dim Null}\,(\mathbf{J}_o) \;=\; \text{dim Null}\left(\mathbf{J}_o^T\right) \;=\; m$$

and

$$\mathbf{f}_\lambda \in \text{Range}\,(\mathbf{J}_o)\,.$$

Now define the sets

$$\left.\begin{array}{rcl} \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_m\} & = & \text{span Null}\,(\mathbf{J}_o) \\ \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_m\} & = & \text{span Null}\left(\mathbf{J}_o^T\right) \end{array}\right\} \;\boldsymbol{\psi}_i \cdot \boldsymbol{\phi}_j \;=\; \delta_{ij} \tag{13}$$

and $\boldsymbol{\phi}_o$ as the unique solution to

$$\mathbf{J}_o \boldsymbol{\phi}_o = -\mathbf{f}_\lambda^o \;\;\text{such that}\;\; \boldsymbol{\phi}_o \cdot \boldsymbol{\psi}_j = 0 \;\; (j = 1, 2, \ldots, m). \tag{14}$$

It follows from eqs. (4), (13) and (14) that

$$\dot{\mathbf{u}}(s_o) = \dot{\mathbf{u}}_o = \sum_{j=0}^{m} \alpha_j \boldsymbol{\phi}_j \tag{15}$$

where $\alpha_o = \dot{\lambda}_o$. Differentiating eq. (4) with respect to s, and rearranging terms gives

$$\mathbf{f}_\mathbf{u}^o \ddot{\mathbf{u}}_o + \mathbf{f}_\lambda^o \ddot{\lambda}_o = -\mathbf{f}_{\mathbf{u}\mathbf{u}}^o \dot{\mathbf{u}}_o \dot{\mathbf{u}}_o - 2\mathbf{f}_{\mathbf{u}\lambda}^o \dot{\lambda}_o \dot{\mathbf{u}}_o - \mathbf{f}_{\lambda\lambda}^o \dot{\lambda}_o \dot{\lambda}_o \tag{16}$$

Since both $\mathbf{f}_\mathbf{u}^o \ddot{\mathbf{u}}_o$ and $\mathbf{f}_\lambda^o \ddot{\lambda}_o$ are in the range of $\mathbf{f}_\mathbf{u}^o$ it follows that expression on the right hand side of eq. (16) is also in the range of $\mathbf{f}_\mathbf{u}^o$. Using results from the Fredholm alternative, ([7], pp. 290-292), it can be shown that

$$\boldsymbol{\psi}_i \cdot \left[\mathbf{f}_{\mathbf{u}\mathbf{u}}^o \dot{\mathbf{u}}_o \dot{\mathbf{u}}_o + 2\mathbf{f}_{\mathbf{u}\lambda}^o \dot{\lambda}_o \dot{\mathbf{u}}_o + \mathbf{f}_{\lambda\lambda}^o \dot{\lambda}_o \dot{\lambda}_o\right] = 0, \;\; (i = 1, 2, \ldots, m). \tag{17}$$

Equations (15) and (17) lead to the well known result from bifurcation theory that if a transverse bifurcating branch at the point $(\mathbf{u}_o, \lambda_o)$ exists, it must have a local form

$$\begin{aligned} \mathbf{u}(s) &= \mathbf{u}_o + s\left(\sum_{j=0}^{m} \alpha_j \boldsymbol{\phi}_j\right) + o(s) \\ \lambda(s) &= \lambda_o + \alpha_0 s + o(s) \end{aligned} \tag{18}$$

where $\alpha_0, \alpha_1, \ldots, \alpha_m$ are isolated roots to the quadratic bifurcation equations

$$\sum_{j=1}^{m}\sum_{k=1}^{m} A_{ijk}\alpha_j\alpha_k + 2\sum_{j=1}^{m} B_{ij}\alpha_j\alpha_0 + C_i\alpha_0^2 \;=\; 0, \quad 1 \le i \le m \tag{19}$$

where

$$\begin{aligned} A_{ijk} &= \boldsymbol{\psi}_i \cdot \mathbf{f}^o_{\mathbf{uu}}\boldsymbol{\phi}_j\boldsymbol{\phi}_k, \\ B_{ij} &= \boldsymbol{\psi}_i \cdot [\mathbf{f}^o_{\mathbf{uu}}\boldsymbol{\phi}_0 \;+\; \mathbf{f}^o_{\mathbf{u}\lambda}]\,\boldsymbol{\phi}_j, \\ C_i &= \boldsymbol{\psi}_i \cdot [\mathbf{f}^o_{\mathbf{uu}}\boldsymbol{\phi}_0\boldsymbol{\phi}_0 \;+\; 2\mathbf{f}^o_{\mathbf{u}\lambda}\boldsymbol{\phi}_0 \;+\; \mathbf{f}^o_{\lambda\lambda}]\,, \\ \mathbf{f}^o_{\mathbf{uu}} &= \frac{d^2\mathbf{f}}{d\mathbf{u}^2}\Big|_{(\mathbf{u}_o,\lambda_o)}\,. \end{aligned} \tag{20}$$

If a solution to eq. (19) exists, the coefficients $\alpha_0, \alpha_1, \ldots, \alpha_m$ can be substituted into eq. (18) to get a prediction step for a branch switching routine. Note that solving eq. (19) involves computing the third order tensor $\mathbf{f}_{\mathbf{uu}}$ which can be prohibitively expensive. Therefore, instead of using the "exact" tangent which can be had by solving eq. (19) it is often more prudent to use less exact but more economical approximations for a prediction step onto a bifurcation branch, (see for example [3], pp.251-253).

Of particular importance in practical applications is the case of a simple bifurcation point ($m = 1$). At a simple bifurcation point the tangent vectors to the two independent solution branches through $(\mathbf{u}_o, \lambda_o)$ are given by

$$\dot{\mathbf{u}}_o = \alpha_0\boldsymbol{\phi}_0 + \alpha_1\boldsymbol{\phi}_1, \quad \dot{\lambda}_o = \alpha_0 \tag{21}$$

where $[\alpha_0, \alpha_1]$ are the roots to

$$A_{111}\alpha_1^2 + 2B_{11}\alpha_1\alpha_0 + C_1\alpha_0^2 \;=\; 0. \tag{22}$$

Due to the strongly convergent nature of Newton's and Newton-like methods, instead of solving for α_0 and α_1 exactly it is often sufficient to use a simple prediction step of the form

$$\dot{\mathbf{u}}_o = \Delta s\boldsymbol{\phi}_1, \quad \dot{\lambda}_o = 0. \tag{23}$$

to switch branches at a simple bifurcation point.

2.3 Bifurcation in One-Parameter Problems

As discussed earlier, a necessary condition for bifurcation to occur at a solution point $(\mathbf{u}_o, \lambda_o)$ is that the Jacobian matrix $\mathbf{J}_o$ be singular. We present here a simple well-known heuristic argument which implies that bifurcation in one-parameter problems of the form shown in eq. (1) is a *non-generic* occurrence.

Consider solving eq. (4) at a singular point $(\mathbf{u}_o, \lambda_o)$ when $\mathbf{J}_o$ has a one-dimensional null space which satisfies

$$\mathbf{J}_o\boldsymbol{\phi}_1 = 0 \;\text{ and }\; \mathbf{J}_o^T\boldsymbol{\psi}_1 = 0 \tag{24}$$

with $\boldsymbol{\phi}_1, \boldsymbol{\psi}_1 \neq 0$. Using the Fredholm alternative, we can classify the singular point $(\mathbf{u}_o, \lambda_o)$ as either a limit point or bifurcation point depending on whether or not $\mathbf{f}^o_\lambda$ is in the range of $\mathbf{J}_o$.

- **Limit Point**
  $\mathbf{f}^o_\lambda \notin \text{Range}(\mathbf{J}_o) \;\Leftrightarrow\; \boldsymbol{\psi}_1 \cdot \mathbf{f}^o_\lambda \neq 0 \Rightarrow \dot{\lambda}_o = 0, \dot{\mathbf{u}}_o = \alpha \boldsymbol{\phi}_1$

- **Bifurcation Point**
  $\mathbf{f}^o_\lambda \in \text{Range}(\mathbf{J}_o) \;\Leftrightarrow\; \boldsymbol{\psi}_1 \cdot \mathbf{f}^o_\lambda = 0 \Rightarrow \dot{\mathbf{u}}_o = \dot{\lambda}_o \boldsymbol{\phi}_o + \alpha_1 \boldsymbol{\phi}_1$ where $\boldsymbol{\phi}_o$ is the unique solution to $\mathbf{J}_o \boldsymbol{\phi}_o = -\mathbf{f}^o_\lambda$ such that $\boldsymbol{\phi}_o \cdot \boldsymbol{\psi}_1 = 0$.

For a bifurcation to occur in eq.(1), the loading vector $\mathbf{f}^o_\lambda$ must be *orthogonal* to the null vector of $\mathbf{J}^T_o$. Unless there is something special about eq. (1) that would enforce this requirement one would generically expect that $\boldsymbol{\psi}_1 \cdot \mathbf{f}^o_\lambda \neq 0$ in which case $(\mathbf{u}_o, \lambda_o)$ would be a limit point. In §3, we will show how bifurcation points can become generic if there is symmetry in the problem.

3 Group Theory Applied to Nonlinear Bifurcation Problems

Before discussing how group theory can be applied to the nonlinear bifurcation analysis of problems with symmetry, it is helpful to first to have at our disposal a few definitions and basic concepts from group representation theory. We begin by giving a definition of a group [30].

Definition 3.1 A *group* $\mathcal{G}$ is a set of objects $\{g, h, k, \ldots\}$ (not necessarily countable) together with a binary operation which associates with any ordered pair of elements $g, h \in \mathcal{G}$ a third element $gh \in \mathcal{G}$. The *order* of a group is the number of elements in the group. The binary operation (called group multiplication) is subject to the following requirements:

1. there exists an *identity* element e in $\mathcal{G}$ such that $ge = eg = g$ for all $g \in \mathcal{G}$;
2. for every $g \in \mathcal{G}$ there exists in $\mathcal{G}$ an *inverse element* g^{-1} such that $gg^{-1} = g^{-1}g = e$;
3. the identity $(gh)k = g(hk)$ is satisfied for $g, h, k \in \mathcal{G}$.

A *subgroup* $\mathcal{H}$ of $\mathcal{G}$ is a set $\mathcal{H} \subseteq \mathcal{G}$, including the possibility that $\mathcal{H} \equiv \mathcal{G}$, such that $\mathcal{H}$ is a group.

Consider the deformation of the three-bar truss depicted in Fig. 4. The truss is composed of three identical uniform one-dimensional elastic rods of length L. The rods are pinned together at point A_0 and simply supported at points A_1, A_2 and A_3. H is the height of the truss. Let $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ represent the standard orthonormal basis in Euclidean three space $\mathbb{E}^3$ and define a three-dimensional vector (configuration) space $V_{\text{truss}} = \text{span}(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$. The truss is deformed by a simple dead load $-\lambda \mathbf{E}_3$ and the deflection of point A_0 with respect to the initial configuration of the truss is determined by the displacement vector

$$\mathbf{u} = u_1 \mathbf{E}_1 + u_2 \mathbf{E}_2 + u_3 \mathbf{E}_3 \cong (u_1,\ u_2,\ u_3). \tag{25}$$

If one assumes that the rods can only deform in extension or compression, *i.e.,* the rods cannot shear or bend, then locating point A_0 provides a complete kinematic description of the deformed truss. The nonlinear equilibrium equations for the truss can be abstractly written as

$$\mathbf{f}_{\text{truss}}(\mathbf{u}, \lambda) = f_1(\mathbf{u}, \lambda)\mathbf{E}_1 + f_2(\mathbf{u}, \lambda)\mathbf{E}_2 + f_3(\mathbf{u}, \lambda)\mathbf{E}_3 = \mathbf{0} \tag{26}$$

where $\mathbf{f}_{\text{truss}} : V_{\text{truss}} \times \mathbb{R} \mapsto V_{\text{truss}}$. Equation (26) represents a set of three nonlinear algebraic equations whose solution $(\mathbf{u}, \lambda)$ represents an equilibrium configuration of the truss.

Let Ω represent the three-bar truss in the initial, undeformed configuration. The *body* or *structural* symmetry [48] of Ω is that of an equilateral triangle, which implies that Ω is unchanged by rotations of $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ radians about the $\mathbf{E}_3$ axis, as well as reflections across the $\mathbf{E}_1 - \mathbf{E}_3$ plane.

```
Start with x_0 = (u_0, λ_0) and Δs
stepcut=0; contflag=0; k = 1
while contflag = 0
        compute initial prediction x_k
        corrflag = 0
        while corrflag = 0
                solve J_k z_k = f_k
                solve J_k y_k = (f_λ)_k
                compute N_k
                x_k = x_k + Δx(z_k, y_k, N_k)
                check corrector stopping criteria†
        end
        if corrflag=1
           k = k + 1; stepcut=0; adjust Δs
           diagnose J_k for bifurcations
        else
           Δs = Δs/2; stepcut=stepcut+1
        end
        check continuation stopping criteria‡
end
```

† (corrflag=1) convergence attained $||\mathbf{f}_k|| < \epsilon$
(corrflag=2) divergence detected
(corrflag=3) maximum number of iterations exceeded

‡ (contflag=1) $k >$ maximum number of allowable steps
(contflag=2) stepcut $>$ maximum number of allowable step cuts
(contflag=3) $||\lambda_k|| >$ maximum allowable load factor

Figure 3: Pseudo code for the standard continuation/bifurcation procedure using the "Bordering Algorithm."

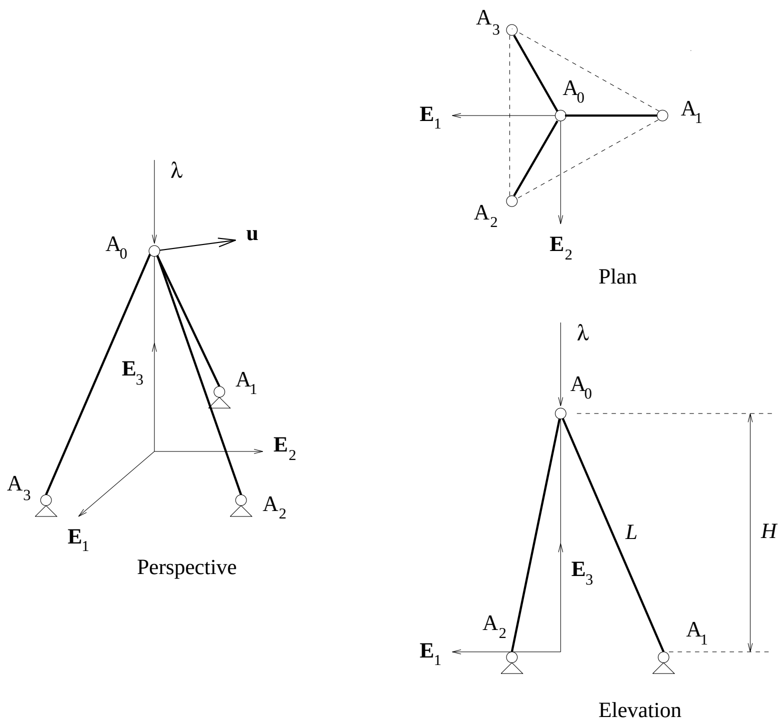

Figure 4: Three-Bar Truss.

In the language of group theory, the symmetry of Ω and the loading in Fig. 4 is characterized by the dihedral group D_3 [30], [31].

D_3 is the group of rotations and reflections which mathematically describe the symmetry of an equilateral triangle. The group D_3 is *generated* by two elements r and s. r is a rotation through an angle of $\frac{2\pi}{3}$ radians and s is a reflection. D_3 is a symmetry group of order 6 and can be represented by the set

$$D_3 = \left\{e, r, r^2, s, sr, sr^2\right\} \tag{27}$$

where $e \equiv r^3 \equiv s^2$ is the identity operator and $srs = r^{-1}$. Figure 5 illustrates a geometrical interpretation of the action of the elements of D_3 on an equilateral triangle.

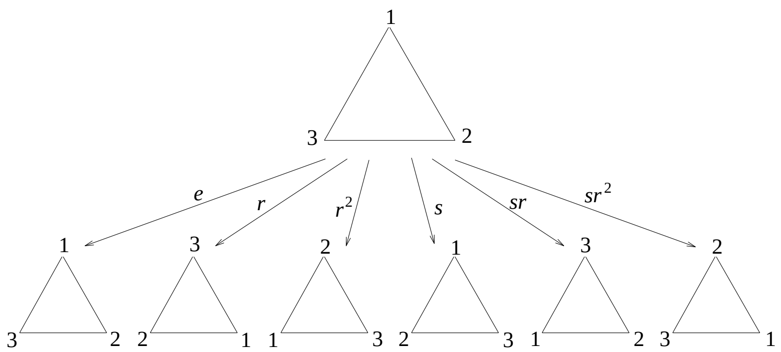

Figure 5: Action of D_3 on an Equilateral Triangle.

The symmetry group D_3 is an example of the dihedral group D_n. D_n is the group of rotations r^k through an angle $\frac{2\pi k}{n}$ $(k = 1, 2, \ldots, n)$ and reflections s which characterize the symmetry of a regular, n-sided polygon such that $e \equiv r^n \equiv s^2$ and $srs = r^{-1}$. D_n is a group of order $2n$ and can be abstractly represented by the set

$$D_n = \left\{e, r, r^2, \ldots, r^{n-1}, s, sr, sr^2, \ldots sr^{n-1}\right\}. \tag{28}$$

A symmetry group which is closely related to D_n, is the cyclic group C_n. C_n is the n^{th} order group of rotations r^k through an angle $\frac{2\pi k}{n}$ $(k = 1, 2, \ldots, n)$ such that $e \equiv r^n$. C_n is a subgroup of D_n and can be represented by the set

$$C_n = \left\{e, r, r^2, \ldots, r^{n-1}\right\}. \tag{29}$$

A key step in applying the abstract ideas and concepts of group theory to nonlinear structural mechanics is devising a method to "represent" the action of the elements of an abstract group $\mathcal{G}$ on a configuration space V. Before precisely defining what is meant by "representing the action of a group on a vector space," consider the action of the following specific orthogonal transformations on the displacement vector $\mathbf{u} \in V_{\text{truss}}$.

With respect to the orthonormal basis in Fig. 4, a clockwise rotation of $\frac{2\pi n}{3}$ radians, $n \in \mathbb{Z}$, of the displacement field $\mathbf{u}$ about the $\mathbf{E}_3$ axis has a matrix representation

$$\mathbf{R}_{\frac{2\pi n}{3}} = \begin{bmatrix} \cos\frac{2\pi n}{3} & \sin\frac{2\pi n}{3} & 0 \\ -\sin\frac{2\pi n}{3} & \cos\frac{2\pi n}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{30}$$

Similarly, a reflection of $\mathbf{u}$ across the $\mathbf{E}_1 - \mathbf{E}_3$ plane is represented by

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

We can now define a new set

$$\mathcal{D}_3 = \left\{\mathbf{I}, \mathbf{R}_{\frac{2\pi}{3}}, \mathbf{R}_{\frac{4\pi}{3}}, \mathbf{J}, \mathbf{J}\mathbf{R}_{\frac{2\pi}{3}}, \mathbf{J}\mathbf{R}_{\frac{4\pi}{3}}\right\} \tag{32}$$

where $\mathbf{I} = \mathbf{R}_{2\pi}$ is the identity matrix. A special characteristic of the elements of $\mathcal{D}_3$ is that they represent rigid motions of the truss which leave the truss "unchanged." In other words, "... no mechanical experiment is capable of detecting a difference between the body(truss) and the transformed body(truss)" [48].

An isomorphism may be defined between elements of $\mathcal{D}_3$ acting on $\mathbf{u}$ and elements of D_3 "acting" on an equilateral triangle, *i.e.*, for every element of D_3 which rotates and/or reflects the equilateral triangle in Fig. 5, there is a corresponding element of $\mathcal{D}_3$ which performs a "similar" transformation on the vector field $\mathbf{u}$. The following definition makes this correspondence precise [30].

Definition 3.2 Let V be a real, n-dimensional vector space and identify $\mathcal{O}_n$ as the space of all orthogonal $n \times n$ matrices such that $\mathcal{O}_n : V \mapsto V$. An *orthogonal, linear representation* of a group $\mathcal{G}$ on V is a homomorphism $\mathcal{T} : \mathcal{G} \mapsto \mathcal{O}_n$. In other words for each element $g \in \mathcal{G}$ there exist a $\mathcal{T}_g \equiv \mathcal{T}(g) \in \mathcal{O}_n$ such that $\mathcal{T}(g_1)\mathcal{T}(g_2) = \mathcal{T}(g_1 g_2) \Rightarrow \mathcal{T}(g^{-1}) = \mathcal{T}^{-1}(g)$ and $\mathcal{T}(e) = I$.

As an example of Definition 3.2, consider an operator $\tilde{\mathcal{T}} : D_3 \mapsto \mathcal{D}_3$, such that the following isometries exist between elements of D_3 and $\mathcal{D}_3$,

$$\begin{array}{lll} \tilde{\mathcal{T}}(e) = \mathbf{I}, & \tilde{\mathcal{T}}(r) = \mathbf{R}_{2\pi/3}, & \tilde{\mathcal{T}}(r^2) = \mathbf{R}_{4\pi/3}, \\ \tilde{\mathcal{T}}(s) = \mathbf{J}, & \tilde{\mathcal{T}}(sr) = \mathbf{J}\mathbf{R}_{2\pi/3}, & \tilde{\mathcal{T}}(sr^2) = \mathbf{J}\mathbf{R}_{4\pi/3}. \end{array} \tag{33}$$

From Definition 3.2 and eq. (33) it can be shown that $\tilde{\mathcal{T}}$ is an orthogonal, linear representation on $V_{\mathbf{truss}}$ of the symmetry group D_3.

3.1 Nonlinear Reduction

It is reasonable that in developing a mathematical model of a physical system with symmetry, one should require that the equilibrium equations of that model reflect the same symmetry found in the physical system. For example, the equilibrium eq. (26) should reflect the D_3-symmetry of the three-bar truss. The equilibrium equations for a broad class of symmetric problems in engineering and science frequently reflect the symmetry of the physical system through a property known as *equivariance* [5], [12].

Definition 3.3 Let $\mathcal{T} \in \mathcal{O}_n$ be an orthogonal, linear representation of a group $\mathcal{G}$ with $\mathcal{T}_g \equiv \mathcal{T}(g)$.
An operator $\mathbf{f}(\mathbf{u}, \lambda)$, $\mathbf{f} : V \times \mathbb{R} \mapsto V$, is said to be *equivariant* under the action of $\mathcal{T}$ if

$$\mathcal{T}_g \mathbf{f}(\mathbf{u}, \lambda) = \mathbf{f}(\mathcal{T}_g \mathbf{u}, \lambda) \qquad \forall\ g \in \mathcal{G}. \tag{34}$$

Equivariance allows one to systematically find proper subspaces of V which contain global solution paths to (1) of a specific symmetry type. The symmetry of a vector $\mathbf{u} \in V$ is characterized by its *isotropy subgroup* $\mathcal{H}$ of $\mathcal{G}$ defined as

$$\mathcal{H} \equiv \{g \in \mathcal{G} : \mathcal{T}_g \mathbf{u} = \mathbf{u}\} \tag{35}$$

where $\mathcal{T}$ is a representation of the group $\mathcal{G}$ on V. Complimenting the idea of an isotropy subgroup is the fixed point set. Let $\mathcal{H}$ be a subgroup of $\mathcal{G}$, including the possibility that $\mathcal{H} \equiv \mathcal{G}$. The *$\mathcal{H}$-fixed-point set* $V_{\mathcal{H}}$ is defined as

$$V_{\mathcal{H}} \equiv \{\mathbf{u} \in V : \mathcal{T}_g \mathbf{u} = \mathbf{u}\ \forall g \in \mathcal{H}\}. \tag{36}$$

$V_{\mathcal{H}}$ is a linear subspace of V which is invariant under the action of $\mathcal{H}$.

Assume that $\mathcal{H}$ is the isotropy group of a vector $\mathbf{u}$ and $V_{\mathcal{H}}$ is the corresponding fixed point set of $\mathcal{H}$. From the results in eqs. (34) and (36), it follows that

$$\mathcal{T}_g \mathbf{f}(\mathbf{u}, \lambda) = \mathbf{f}(\mathcal{T}_g \mathbf{u}, \lambda) = \mathbf{f}(\mathbf{u}, \lambda) \tag{37}$$

$\forall\ (\mathbf{u}, \lambda) \in V_{\mathcal{H}} \times \mathbb{R}$ and $\forall g \in \mathcal{H}_{\mathbf{u}}$. It is clear from (37) that $\mathbf{f} : V_{\mathcal{H}} \times \mathbb{R} \mapsto V_{\mathcal{H}}$. In other words $\mathbf{f}$ has $V_{\mathcal{H}}$ as an invariant subspace.

3.2 Nonlinear Reduction and Block Diagonalization

Let $\mathcal{G}$ represent the isotropy group which characterizes the symmetry of a solution path to eq. (1). Group representation theory offers a systematic methodology for choosing an "optimal" set of basis vectors, or *symmetry modes*, which resolve the configuration space V into a set of p mutually orthogonal subspaces $V^{(\mu)}$ such that

$$V = \sum_{\mu=1}^{p} \oplus V^{(\mu)}. \tag{38}$$

The parameter p is determined by the symmetry group $\mathcal{G}$. It can be shown that each one of these subspaces $V^{(\mu)}$ reflects part of the symmetry of the original system. The subspace $V^{(1)}$ is typically designated as the $\mathcal{G}$-invariant subspace which implies that the elements of $V^{(1)}$ reflect the symmetry of $\mathcal{G}$. For problems with circular symmetry, $V^{(1)}$ might represent the set of all axisymmetric solutions. The *$\mathcal{G}$-reduced problem*

$$\begin{aligned} \mathbf{f}^{(1)}(\mathbf{u}, \lambda) &= \mathbf{0} \\ \mathbf{f}^{(1)} : V^{(1)} \times \mathbb{R} &\rightarrow V^{(1)} \end{aligned} \tag{39}$$

is defined by restricting $\mathbf{f}(\mathbf{u}, \lambda)$ to $V^{(1)}$. Solutions to (39) are exact, global solutions to (1) which lie in the subspace $V^{(1)}$. Note that eq. (39) represents a dimensional reduction in problem size.

Once the decomposition of V in eq. (38) is carried out, an orthonormal basis of symmetry modes is chosen for each one of the subspaces. A remarkable result from group representation theory states that if the symmetry modes are used as basis vectors for V the Jacobian matrix, defined as $\mathbf{J} = \left.\frac{d\mathbf{f}}{d\mathbf{u}}\right|_{\mathbf{u} \in V^{(1)}}$, block diagonalizes, and the incremental equations for a Newton step in a continuation procedure take the form

$$\begin{bmatrix} \mathbf{J}^{(1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{(2)} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}^{(p)} \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(1)} & | & \mathbf{z}^{(1)} \\ \mathbf{y}^{(2)} & | & \mathbf{z}^{(2)} \\ \vdots & | & \\ \mathbf{y}^{(p)} & | & \mathbf{z}^{(p)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\lambda}^{(1)} & | & \mathbf{f}^{(1)} \\ \mathbf{0} & | & \mathbf{0} \\ \vdots & | & \\ \mathbf{0} & | & \mathbf{0} \end{bmatrix} \tag{40}$$

where $\mathbf{y}^{(\mu)}, \mathbf{z}^{(\mu)} \in V^{(\mu)}$. It can be shown generically that $\mathbf{f}(\mathbf{u}, \lambda)$, $\mathbf{f}_{\lambda}(\mathbf{u}, \lambda) \in V^{(1)}$ for all $(\mathbf{u}, \lambda) \in V^{(1)} \times \mathbb{R}$. Thus, to trace out a $\mathcal{G}$-symmetric solution path with an continuation procedure it is only necessary to assemble the matrix $\mathbf{J}^{(1)}$ and solve

$$\begin{aligned} \mathbf{J}^{(1)}\mathbf{y}^{(1)} &= \mathbf{f}_{\lambda}^{(1)} \\ \mathbf{J}^{(1)}\mathbf{z}^{(1)} &= \mathbf{f}^{(1)}. \end{aligned} \tag{41}$$

A group theoretic based, nonlinear continuation/bifurcation procedure would consist of 2 parts:

1. The reduced problem $\mathbf{f}^{(1)}$ with its associated Jacobian matrix $\mathbf{J}^{(1)}$ are used to calculate the $\mathcal{G}$-symmetric solution paths;

2. As solution points along a $\mathcal{G}$-symmetric path are found, bifurcation points can be diagnosed by assembling the matrices $\mathbf{J}(\mu)$ $(\mu = 2, 3, \dots, p)$ and checking for the occurrence of zero eigenvalues.

The block diagonalization of $\mathbf{J}$ is the key to circumventing the numerical ill-conditioning frequently encountered in problems such as shell stability analysis. It can be generically shown that the only type of singularity which occurs in $\mathbf{J}^{(1)}$ along a $\mathcal{G}$-symmetric solution branch is a limit point, which does not represent a loss of uniqueness of the solution and therefore poses little difficulty for an pseudo arc-length continuation procedure. The singular points encountered along a solution curve which could cause severe convergence problems for the correction algorithm are not an issue because they generically only occur in the orthogonal blocks $\mathbf{J}^{(\mu)}$ which are never used in the continuation process. To see this, recall the results in §2.3 which showed that at singular point with a one-dimensional null space satisfying eq. (24) a necessary condition for bifurcation to occur was shown to be $\boldsymbol{\psi}_1 \cdot \mathbf{f}_\lambda^o = 0$.

- Suppose $\mathbf{J}_o^{(1)}$ is singular with a one-dimensional null space satisfying $\mathbf{J}_o^{(1)} \boldsymbol{\phi}_1^{(1)} = 0$ and $\left(\mathbf{J}_o^{(1)}\right)^T \boldsymbol{\psi}_1^{(1)} = 0$. Since both $\mathbf{f}_\lambda^o$ and $\boldsymbol{\psi}_1^{(1)}$ are elements of $V^{(1)}$ one would generically expect that $\boldsymbol{\psi}_1^{(1)} \cdot \mathbf{f}_\lambda^o \neq 0$ in which case $(\mathbf{u}_o, \lambda_o)$ must be a limit point.
- Suppose $\mathbf{J}_o^{(\mu)}$, $(\mu = 2, 3, \ldots, p)$, is singular with a one-dimensional null space satisfying $\mathbf{J}_o^{(\mu)} \boldsymbol{\phi}_1^{(\mu)} = 0$ and $\left(\mathbf{J}_o^{(\mu)}\right)^T \boldsymbol{\psi}_1^{(\mu)} = 0$. In this case $\mathbf{f}_\lambda^o \in V^{(1)}$ **but** $\boldsymbol{\psi}_1^{(\mu)} \in V^{(\mu)}$. By definition every vector contained in $V^{(\mu)}$ is orthogonal to every vector contained in $V^{(1)}$, therefore $\boldsymbol{\psi}_1^{(\mu)} \cdot \mathbf{f}_\lambda^o = 0$ and $(\mathbf{u}_o, \lambda_o)$ would generically be a bifurcation point.

It is important to keep in mind that the above argument is not rigorous but only holds in a generic sense.

If a singular point along the $\mathcal{G}$-symmetric primary solution path turns out to be a bifurcation point, the isotropy group $\mathcal{H}$ of the primary bifurcating branch is generically not equivalent to $\mathcal{G}$ and is referred to as a *symmetry breaking bifurcation*. It generically turns out that $\mathcal{H}$ is a proper subgroup of $\mathcal{G}$ and the primary bifurcation branch typically exhibits less symmetry than the primary solution branch.[3] The block structure of the incremental equations along branches with different symmetry will in general be different. As one passes from a branch with more symmetry to less symmetry, the number of blocks decreases while the size of the individual blocks increases. Therefore, as one moves to branches associated with smaller symmetry groups, the advantages of a group theoretic approach diminishes.

3.3 D_6-Symmetric Dome

An instructive vehicle for demonstrating a group theoretic approach to nonlinear bifurcation analysis is the static-deformation of the hexagonal, *i.e.,* D_6-symmetric, lattice-dome structure which was originally analyzed in [1],[11] and further discussed in [17] and [18]. The details of the group analysis and finite element formulation for this problem can be found in [11] and will only be summarized here.

The governing equations of the lattice-dome discussed in [1] were cast in the form

$$\begin{aligned} \mathbf{f}(\mathbf{U}, \lambda) &= \mathbf{0}, \\ \mathbf{f} : V_{\text{dome}} \times \mathbb{R} &\mapsto V_{\text{dome}} \end{aligned} \tag{42}$$

where $V_{\text{dome}} \subseteq \mathbb{R}^{21}$, $\mathbf{U} \in V_{\text{dome}}$ and $\lambda \in \mathbb{R}$ was a loading parameter. Solving what will be referred to in this section as the *dome-problem*, entailed finding solutions to (42) which were connected to the zero state $(\mathbf{U}, \lambda) = (\mathbf{0}, 0)$.

[3] Though as noted in [11], it is often the case that some of the original symmetry is preserved on the bifurcating branch.

3.3.1 Dome Equations in a Standard Basis

The first coordinate system in which the dome-problem could be formulated is referred to as the "standard basis". Let the displacement vector $\mathbf{U}$ be represented by a column vector of components written as

$$\mathbf{U} \cong (u_1, u_2, u_3, u_4, \ldots, u_{19}, u_{20}, u_{21})^T. \tag{43}$$

In eq. (43), the displacement components $(u_1, u_2, \ldots, u_{21})$ are the displacement components with respect to the standard orthonormal basis

$$\mathbf{e}_i = (\overset{1}{0}, \overset{2}{0}, \overset{3}{0}, \ldots, \overset{i}{1}, \ldots, \overset{20}{0}, \overset{21}{0})^T, \qquad (i = 1, 2, \ldots, 21). \tag{44}$$

The coordinate system described above would be a natural choice for a standard commercial nonlinear finite element package. Figure 6 depicts the structure of the incremental equations in the standard basis and visually demonstrates that all twenty-one degrees of freedom in the standard coordinate system are coupled together.

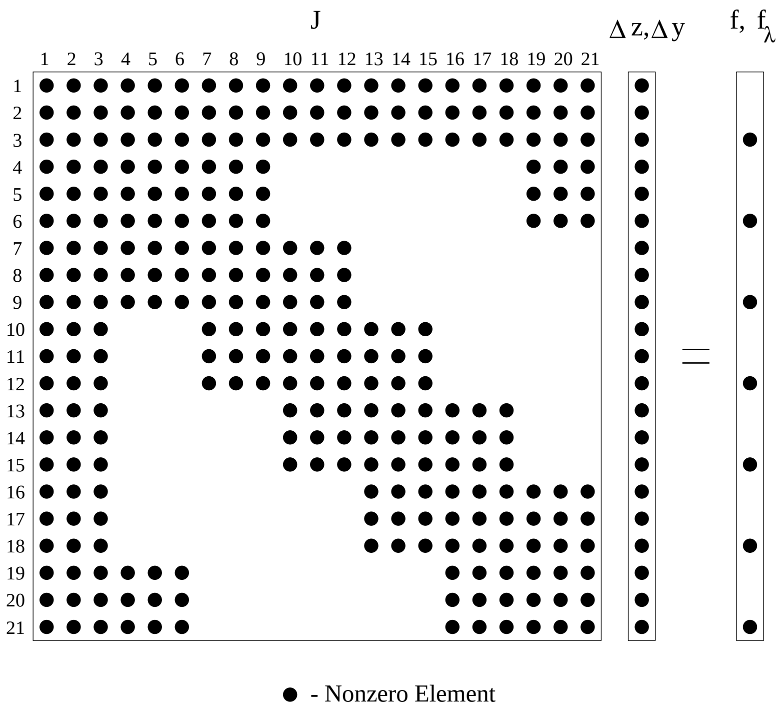

Figure 6: Incremental equations — standard coordinate system

3.3.2 Dome Equations in a Symmetry-Adapted Basis

In [14] a detailed algorithm is presented, which can be used to automatically compute a symmetry-adapted basis for lattice-dome structures with D_n symmetry. Making use of this algorithm, the set of basis vectors displayed in Table 1 was computed for a D_6-symmetric solution path. Note that V_{dome} has been partitioned into a set of eight mutually orthogonal subspaces and may represented as

$$V_{\text{dome}} = \sum_{\mu=1}^{6} \oplus V^{(\mu)} \tag{45}$$

Table 1: Symmetry-Adapted Basis for the Hexagonal Dome

<u>Basis for $\mathrm{V}^{(1)}$</u>	$\hat{\mathbf{e}}_1$	$=$	$\mathbf{e}_3$
	$\hat{\mathbf{e}}_2$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_6+\mathbf{e}_9+\mathbf{e}_{12}+\mathbf{e}_{15}+\mathbf{e}_{18}+\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_3$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_4-\mathbf{e}_{13})+\sqrt{3}(\mathbf{e}_8+\mathbf{e}_{11}-\mathbf{e}_{17}-\mathbf{e}_{20})+\mathbf{e}_7-\mathbf{e}_{10}-\mathbf{e}_{16}+\mathbf{e}_{19}\right]$
<u>Basis for $\mathrm{V}^{(2)}$</u>	$\hat{\mathbf{e}}_4$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_5-\mathbf{e}_{14})+\sqrt{3}(-\mathbf{e}_7-\mathbf{e}_{10}+\mathbf{e}_{16}+\mathbf{e}_{19})+\mathbf{e}_8-\mathbf{e}_{11}-\mathbf{e}_{17}+\mathbf{e}_{20}\right]$
<u>Basis for $\mathrm{V}^{(3)}$</u>	$\hat{\mathbf{e}}_5$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_6-\mathbf{e}_9+\mathbf{e}_{12}-\mathbf{e}_{15}+\mathbf{e}_{18}-\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_6$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_4+\mathbf{e}_{13})+\sqrt{3}(-\mathbf{e}_8+\mathbf{e}_{11}-\mathbf{e}_{17}+\mathbf{e}_{20})-\mathbf{e}_7-\mathbf{e}_{10}-\mathbf{e}_{16}-\mathbf{e}_{19}\right]$
<u>Basis for $\mathrm{V}^{(4)}$</u>	$\hat{\mathbf{e}}_7$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_5+\mathbf{e}_{14})+\sqrt{3}(\mathbf{e}_7-\mathbf{e}_{10}+\mathbf{e}_{16}-\mathbf{e}_{19})-\mathbf{e}_8-\mathbf{e}_{11}-\mathbf{e}_{17}-\mathbf{e}_{20}\right]$
<u>Basis for $\mathrm{V}_1^{(5)}$</u>	$\hat{\mathbf{e}}_8$	$=$	$\mathbf{e}_1$
	$\hat{\mathbf{e}}_9$	$=$	$\frac{1}{\sqrt{12}}\left[2\mathbf{e}_6+\mathbf{e}_9-\mathbf{e}_{12}-2\mathbf{e}_{15}-\mathbf{e}_{18}+\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_{10}$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_4+\mathbf{e}_7+\mathbf{e}_{10}+\mathbf{e}_{13}+\mathbf{e}_{16}+\mathbf{e}_{19}\right]$
	$\hat{\mathbf{e}}_{11}$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_4+\mathbf{e}_{13})+\sqrt{3}(\mathbf{e}_8-\mathbf{e}_{11}+\mathbf{e}_{17}-\mathbf{e}_{20})-\mathbf{e}_7-\mathbf{e}_{10}-\mathbf{e}_{16}-\mathbf{e}_{19}\right]$
<u>Basis for $\mathrm{V}_2^{(5)}$</u>	$\hat{\mathbf{e}}_{12}$	$=$	$\mathbf{e}_2$
	$\hat{\mathbf{e}}_{13}$	$=$	$\frac{1}{2}\left[\mathbf{e}_9+\mathbf{e}_{12}-\mathbf{e}_{18}-\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_{14}$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_5+\mathbf{e}_8+\mathbf{e}_{11}+\mathbf{e}_{14}+\mathbf{e}_{17}+\mathbf{e}_{20}\right]$
	$\hat{\mathbf{e}}_{15}$	$=$	$\frac{1}{\sqrt{24}}\left[2(-\mathbf{e}_5-\mathbf{e}_{14})+\sqrt{3}(\mathbf{e}_7-\mathbf{e}_{10}+\mathbf{e}_{16}-\mathbf{e}_{19})+\mathbf{e}_8+\mathbf{e}_{11}+\mathbf{e}_{17}+\mathbf{e}_{20}\right]$
<u>Basis for $\mathrm{V}_1^{(6)}$</u>	$\hat{\mathbf{e}}_{16}$	$=$	$\frac{1}{\sqrt{12}}\left[2\mathbf{e}_6-\mathbf{e}_9-\mathbf{e}_{12}+2\mathbf{e}_{15}-\mathbf{e}_{18}-\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_{17}$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_4-\mathbf{e}_{13})+\sqrt{3}(-\mathbf{e}_8-\mathbf{e}_{11}+\mathbf{e}_{17}+\mathbf{e}_{20})+\mathbf{e}_7-\mathbf{e}_{10}-\mathbf{e}_{16}+\mathbf{e}_{19}\right]$
	$\hat{\mathbf{e}}_{18}$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_4-\mathbf{e}_7+\mathbf{e}_{10}-\mathbf{e}_{13}+\mathbf{e}_{16}-\mathbf{e}_{19}\right]$
<u>Basis for $\mathrm{V}_2^{(6)}$</u>	$\hat{\mathbf{e}}_{19}$	$=$	$\frac{1}{2}\left[\mathbf{e}_9-\mathbf{e}_{12}+\mathbf{e}_{18}-\mathbf{e}_{21}\right]$
	$\hat{\mathbf{e}}_{20}$	$=$	$\frac{1}{\sqrt{24}}\left[2(\mathbf{e}_5-\mathbf{e}_{14})+\sqrt{3}(\mathbf{e}_7+\mathbf{e}_{10}-\mathbf{e}_{16}-\mathbf{e}_{19})+\mathbf{e}_8-\mathbf{e}_{11}-\mathbf{e}_{17}+\mathbf{e}_{20}\right]$
	$\hat{\mathbf{e}}_{21}$	$=$	$\frac{1}{\sqrt{6}}\left[\mathbf{e}_5-\mathbf{e}_8+\mathbf{e}_{11}-\mathbf{e}_{14}+\mathbf{e}_{17}-\mathbf{e}_{20}\right]$

where $V^{(5)} = V_1^{(5)} \oplus V_2^{(5)}$ and $V^{(6)} = V_1^{(6)} \oplus V_2^{(6)}$. The set of symmetry-adapted basis vectors, or symmetry-modes, $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \ldots, \hat{\mathbf{e}}_{21}\}$ displayed in Table 1 is an orthonormal basis for the configuration space V_{dome}.

As was done with the standard coordinate system in the previous section, the dome equations can be recast with respect to this new coordinate system. For an *arbitrary* deformed configuration, the incremental equations expressed in the symmetry-adapted coordinate system lead to a coupled system of twenty-one equations. However, the configuration of the deformed dome along the primary solution branch is not arbitrary, it maintains its original D_6 symmetry. It has been shown [11] that the exact global D_6-symmetric solutions of eq. (42) lie entirely within the three-dimensional subspace $V^{(1)} \subset V_{\text{dome}}$. The incremental equations expressed in the symmetry-adapted coordinate system has the special block-diagonal structure shown in Fig. 7.

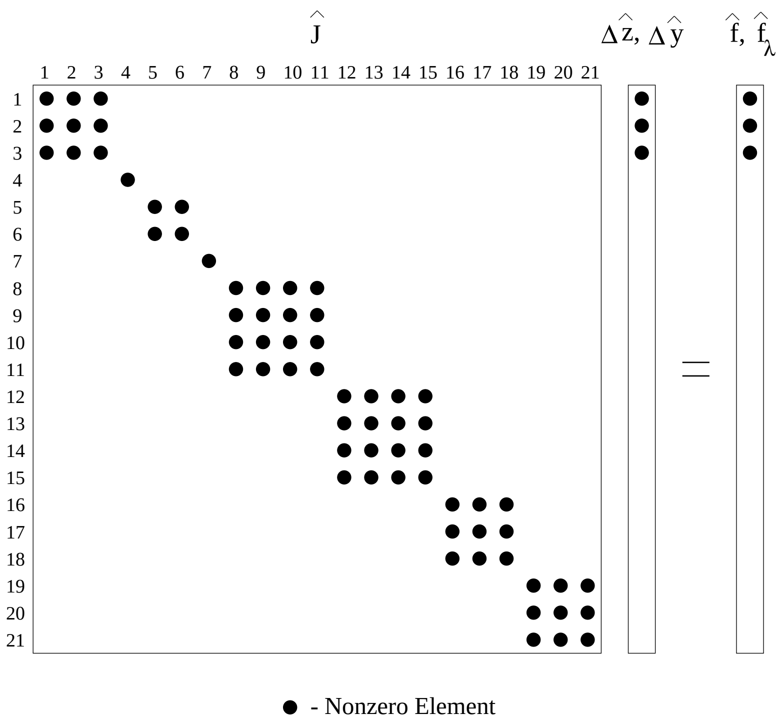

Figure 7: Incremental equations: Symmetry-adapted coordinate system

3.4 Symmetry Transformation

For reasons of programming efficiency and flexibility of use, most general PDE solvers and algorithms are set up to solve equations with respect to "standard" Euclidean, Cartesian or orthogonal curvilinear coordinate systems. The result of this is that while the Jacobian matrix $\mathbf{J}$ written with respect to the standard basis is easily computed the symmetry-adapted Jacobian blocks $\mathbf{J}^{(\mu)}$ are generally only available via similarity transformations. For example, referring back to the dome problem, suppose one were to define an orthogonal matrix

$$\mathbf{Q} \;=\; [\mathbf{Q}_1\ \mathbf{Q}_2\ \mathbf{Q}_3\ \mathbf{Q}_4\ \mathbf{Q}_5\ \mathbf{Q}_6] \in \mathbb{R}^{21\times 21} \quad \mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$$

where $\mathbf{Q}_\mu$, $(\mu = 1, 2, \ldots, 6)$, is a $21 \times |V^{(\mu)}|$ matrix whose columns are made up of the symmetry-adapted basis vectors for $V^{(\mu)}$ in Table 1. Once $\mathbf{Q}$ is defined, the transformation of the incremental equations in the standard basis, (Fig. 6), to the incremental equations in the symmetry-adapted basis, (Fig. 7), can be simply accomplished as shown in eq. (46).

$$
\begin{aligned}
\underbrace{\mathbf{Q}^T\mathbf{J}\mathbf{Q}}_{\hat{\mathbf{J}}}\underbrace{\mathbf{Q}^T y}_{\hat{y}} &= \underbrace{\mathbf{Q}^T\mathbf{f}_\lambda}_{\hat{\mathbf{f}}_\lambda} \;\Rightarrow\; \hat{\mathbf{J}}\hat{y} = \hat{\mathbf{f}}_\lambda \\
\underbrace{\mathbf{Q}^T\mathbf{J}\mathbf{Q}}_{\hat{\mathbf{J}}}\underbrace{\mathbf{Q}^T z}_{\hat{z}} &= \underbrace{\mathbf{Q}^T\mathbf{f}}_{\hat{\mathbf{f}}} \;\Rightarrow\; \hat{\mathbf{J}}\hat{z} = \hat{\mathbf{f}}
\end{aligned} \tag{46}
$$

The reduced problem, (eq. (41)), which can be used to compute the D_6-symmetric path of the dome is similarly formulated as

$$
\begin{aligned}
\underbrace{\mathbf{Q}_1^T\mathbf{J}\mathbf{Q}_1}_{\mathbf{J}^{(1)}}\underbrace{\mathbf{Q}_1^T y}_{y^{(1)}} &= \underbrace{\mathbf{Q}_1^T\mathbf{f}_\lambda}_{\mathbf{f}_\lambda^{(1)}} \;\Rightarrow\; \mathbf{J}^{(1)}y^{(1)} = \mathbf{f}_\lambda^{(1)} \\
\underbrace{\mathbf{Q}_1^T\mathbf{J}\mathbf{Q}_1}_{\mathbf{J}^{(1)}}\underbrace{\mathbf{Q}_1^T z}_{z^{(1)}} &= \underbrace{\mathbf{Q}_1^T\mathbf{f}}_{\mathbf{f}^{(1)}} \;\Rightarrow\; \mathbf{J}^{(1)}z^{(1)} = \mathbf{f}^{(1)}
\end{aligned} \tag{47}
$$

while the orthogonal Jacobian blocks are computed as

$$\mathbf{J}^{(\mu)} = \mathbf{Q}_\mu^T\mathbf{J}\mathbf{Q}_\mu \tag{48}$$

An important feature of the transformations in eqs. (46)-(48) is that $\mathbf{J}$ and $\mathbf{Q}$ are typically sparse matrices. Taking advantage of the inherent sparsity leads to very efficient coordinate transformations, which is crucial in large problems.

To use the above approach, the transformation matrix $\mathbf{Q}$ must be available to the solution procedure in one form or another. Computing $\mathbf{Q}$ entails a *group analysis* of the problem at hand — a detailed discussion of which is beyond the scope of this work. It suffices to say that it is often possible to develop efficient algorithms which allow specified columns of $\mathbf{Q}$ to be computed as needed [16]. For more information about group analyses of symmetric systems, the reader is referred to [5]-[27].

4 Group Theoretic Algorithms

The group theoretic ideas discussed in the previous section have been implemented in two new MATLAB-based symmetry-adapted nonlinear continuation codes contcm.m, (pseudo code Fig. 8), and MMcontcm.m, (pseudo code Fig. 9). The only significant difference between these two codes is that contcm.m was written for a single-processor environment in MATLAB while MMcontcm.m was written for a multi-processor environment in MultiMATLAB. Both codes are based on the following algorithm:

Symmetry Adapted Continuation/Bifurcation Algorithm From the beginning, it is assumed that a solution point $[\mathbf{u}_{k-1}, \lambda_{k-1}]$, with respect to the standard computational coordinate system, is known. A new solution point $[\mathbf{u}_k, \lambda_k]$ is sought.

1. Define an initial guess for $(\mathbf{u}_k, \lambda_k)$
2. Compute the Jacobian matrix $\mathbf{J}$, equilibrium equation $\mathbf{f}$ and the load vector $\mathbf{f}_\lambda$ with respect to the standard computational coordinate system.

3. Assemble and solve the dimensionally-reduced set of linear equations, by computing $\mathbf{J}^{(1)}$, $\mathbf{f}^{(1)}$ and $\mathbf{f}_{\lambda}^{(1)}$, as shown in eq. (47) and $\mathbf{u}_k^{(1)} = \mathbf{Q}_1^T \mathbf{u}_k$.
4. Using an iterative nonlinear solution method, such as Newton-Rhapson, refine the initial guess for $\left(\mathbf{u}_k^{(1)}, \lambda_k\right)$ by successive updates, transforming between the standard computational and symmetry-adapted coordinate systems as needed, until an appropriate convergence criteria is satisfied.
5. Once a new solution point $(\mathbf{u}_k^{(1)}, \lambda_k)$ in the symmetry-adapted coordinate system is found, transform the solution to the standard computational coordinate system via $\mathbf{u}_k = \mathbf{Q}_1 \mathbf{u}_k^{(1)}$.
6. If desired, one or more of the "orthogonal" blocks may now be evaluated at the converged solution point $(\mathbf{u}_k, \lambda_k)$ to check for the occurrence of an orthogonal bifurcation, *e.g.*, $\mathbf{J}^{(\mu)} = \mathbf{Q}_\mu^T \mathbf{J} \mathbf{Q}_\mu$ where $\mathbf{Q}_\mu \in \mathbb{R}^{|V| \times |V^{(\mu)}|}$.
7. The above algorithm may be repeated to search for the next solution point or the procedure may be terminated.

While contcm.m enjoys the improved numerical conditioning and reduction in problem size offered by a group theoretic approach, further computational advantage can be found in a parallel environment. To see this, consider that in contcm.m after a solution point is computed an orthogonal bifurcation analysis is performed and the continuation process is halted until the bifurcation analysis is complete. However, computing successive points along a solution branch is independent of the bifurcation analysis. Furthermore, the bifurcation analysis of the various blocks $\mathbf{J}^{(\mu)}$ $(\mu = 2, 3, \ldots, p)$ are also independent of one another. Therefore, assuming that one processor is dedicated to the continuation portion of the computation and one or more processors to the bifurcation analysis, a parallel version of contcm.m would partition the work as follows:

1. **Continuation Processor 1** The reduced problem $\mathbf{f}^{(1)}$ with its associated Jacobian matrix $\mathbf{J}^{(1)}$ are used to calculate the $\mathcal{G}$-symmetric solution paths. When a solution point $\mathbf{x}_k$ has been found, it can be sent to the bifurcation processors and the computation of the next solution point $\mathbf{x}_{k+1}$ can begin *without* waiting for the bifurcation analysis to be completed.

2. **Bifurcation Processor(s) (2:NP)** Once the solution point $\mathbf{x}_k$ has been received from the continuation procedure, $\mathcal{G}$-symmetry breaking singular points can be diagnosed by assembling the matrices $\mathbf{J}^{(\mu)}$ $(\mu = 2, 3, \ldots, p)$ and checking for the occurrence of zero eigenvalues. Once the bifurcation analysis is complete, the bifurcation processor is ready to receive the next solution point $\mathbf{x}_{k+1}$.

MMcontcm.m is based on the above partitioning of work. It is important to note that a key element in an efficient implementation of the parallel algorithm in Fig. 9 is to make sure that the time needed by the processor 1 to compute successive solution points is less than the time needed by processors 2:NP to perform the bifurcation analysis. This is to avoid the situation of having processors 2:NP sitting idle while waiting for the next solution point.

The potential difficulty which arises is that the continuation part of the procedure is iterative and will not always converge in a predetermined number of steps. Furthermore, in the region of high curvature in the solution path, several adjustments to the step length may be needed to attain convergence.

It is appropriate to note here that two parallel algorithms very similar in spirit to what is displayed in Fig. 9, respectively designed for the linear static and vibration analysis of rotationally

Start with $\mathbf{x}_0 = (\mathbf{u}_0, \lambda_0)$ and Δs
define μ
stepcut=0; contflag=0; $k = 1$
while contflag $= 0$
    compute initial prediction $\mathbf{x}_k$
    corrflag $= 0$
    while corrflag $= 0$
        transform $\mathbf{J}_k^{(1)} = \mathbf{Q}_1^T \mathbf{J}_k \mathbf{Q}_1$, $\mathbf{f}_k^{(1)} = \mathbf{Q}_1^T \mathbf{f}_k$, $(\mathbf{f}_\lambda^{(1)})_k = \mathbf{Q}_1^T (\mathbf{f}_\lambda)_k$
        solve $\mathbf{J}_k^{(1)} \mathbf{z}_k^{(1)} = \mathbf{f}_k^{(1)}$, $\mathbf{J}_k^{(1)} \mathbf{y}_k^{(1)} = (\mathbf{f}_\lambda^{(1)})_k$
        transform $\mathbf{z}_k = \mathbf{Q}_1 \mathbf{z}_k^{(1)}$, $\mathbf{y}_k = \mathbf{Q}_1 \mathbf{y}_k^{(1)}$
        compute N_k
        $\mathbf{x}_k = \mathbf{x}_k + \Delta\mathbf{x}(\mathbf{z}_k, \mathbf{y}_k, N_k)$
        check corrector stopping criteria†
    end
    if corrflag=1
        $k = k + 1$; stepcut=0; adjust Δs
        diagnose orthogonal blocks $\mathbf{Q}_\mu^T \mathbf{J}_k \mathbf{Q}_\mu$ for bifurcations
    else
        $\Delta s = \Delta s/2$; stepcut=stepcut+1
    end
    check continuation stopping criteria‡
end

† (corrflag=1) convergence attained $||\mathbf{f}_k|| < \epsilon$
(corrflag=2) divergence detected
(corrflag=3) maximum number of iterations exceeded

‡ (contflag=1) $k >$ maximum number of allowable steps
(contflag=2) stepcut $>$ maximum number of allowable step cuts
(contflag=3) $||\lambda_k|| >$ maximum allowable load factor

Figure 8: Pseudo code for `contcm.m`

periodic, *i.e.,* C_n-symmetric, structures have been discussed in [49]. The block diagonalization of the Jacobian (stiffness) and mass matrices in [49] appears to be a generalization of the well known de-coupling which occurs while solving linear axisymmetric problems with non-symmetric loads using a semi-analytical (Fourier discretization) technique ([50], pp.195-202). A more systematic and general approach to the linear analysis of symmetric structures, within the context of group representation theory, can be found in [25] (linear static analysis) and [13] (linear vibration analysis).

A very important distinction between what was done in [49] and what is done in this work is that the techniques discussed in [49] are limited to the **linear** analysis of **only** rotationally periodic structures. The techniques used in this work, based on a rigorous group theoretic analysis, are valid for the **global nonlinear** analysis of **any** type of symmetric structure, *e.g.,* spherical shells, cubic structures, *etc.*

5 Numerical Example: Axially-Loaded Cylindrical Shell

The numerical results discussed in this section are based upon a nonlinear bifurcation analysis of an axially-compressed cylindrical shell. The example of a cylindrical shell was chosen for a number of reasons: it is a historically important problem; it is a numerically challenging problem and; the group analysis for the cylinder can with little effort be extended to a wide range of other engineering problems. Thus, it is a good representation of the types of problems which will benefit from a group theoretic approach. The intent of this section is not to do a complete analysis of the cylindrical shell problem, rather it is to demonstrate and discuss different aspects of such an analysis using MMcontcm.m.

5.1 Group Analysis of the Discrete FE Model of the Cylindrical Shell

The cylindrical shell was modeled with four-node nonlinear shell elements with six degrees of freedom per node. The finite element discretization of the cylinder considered in this analysis consisted of N nodes about the circumference and M nodes along the length — resulting in a model with $6MN$ degrees of freedom, (DOF), *i.e.,* $V \in \mathbb{R}^{6MN}$. To demonstrate the performance of MMcontcm.m four different FE meshes ($M \times N$) will be looked at: 21×80 (10080 DOF), 41×80 (19680 DOF), 41×128 (31488 DOF) and 51×144 (44064 DOF).

For a uniform ($M \times N$) mesh, the system of nonlinear equations which model the discretized cylindrical shell are invariant with respect to a representation of the dihedral symmetry group D_N.[4] In [32] a group analysis of the FE model of the cylindrical shell was carried out, the results of which are displayed in Tables 2 and 3. Tables 2 and 3 can be used to determine both the number and sizes of orthogonal subspaces $|V^{(\mu)}|$ for D_n and C_n symmetric solution paths respectively once the FE mesh is chosen and the symmetry of a solution path is identified.

For example, suppose the cylindrical shell was modeled with a uniform FE mesh which consisted of $M = 41$ nodes along the length and $N = 128$ nodes around the circumference, leading to a FE model with D_{128}-symmetry. The isotropy subgroup of primary solution path would be D_{128}, *i.e.,* $n = 128$. Thus, $\frac{N}{n} = 1$ is odd, $n = 128$ is even and Table 2 shows that $|V^{(1)}| = 164$, $|V^{(2)}| = 82$, $|V^{(3)}| = 164$, $|V^{(4)}| = 88$, $|V_1^{(j)}| = 246$, $(j = 1, 2, \ldots, 63)$. Thus, the Jacobian block $\mathbf{J}^{(1)}$ is a 164×164 matrix with an average bandwidth of 6, and a complete orthogonal bifurcation analysis would require assembling 66 orthogonal blocks of dimensions no larger than 246×246

[4] Actually, due to the reflection symmetry about the mid-section the PDE's which model the cylindrical shell possess $D_N \oplus \mathbb{Z}_2$ symmetry. However, we will restrict our attention to the D_N symmetry.

Start with $\mathbf{x}_0 = (\mathbf{u}_0, \lambda_0)$ and Δs
define NP and $\mu(\mathrm{ID}), (\mathrm{ID} = 1, 2, \ldots, \mathrm{NP}-1)$
stepcut=0; contflag=0; $k = 1$
if ID $= 0$ (Continuation Processor ID=0)
  while contflag $= 0$
      compute initial prediction $\mathbf{x}_k$
      corrflag $= 0$
      while corrflag $= 0$
            transform $\mathbf{J}_k^{(1)} = \mathbf{Q}_1^T \mathbf{J}_k \mathbf{Q}_1$, $\mathbf{f}_k^{(1)} = \mathbf{Q}_1^T \mathbf{f}_k$, $(\mathbf{f}_\lambda^{(1)})_k = \mathbf{Q}_1^T (\mathbf{f}_\lambda)_k$
            solve $\mathbf{J}_k^{(1)} \mathbf{z}_k^{(1)} = \mathbf{f}_k^{(1)}$, $\mathbf{J}_k^{(1)} \mathbf{y}_k^{(1)} = (\mathbf{f}_\lambda^{(1)})_k$
            transform $\mathbf{z}_k = \mathbf{Q}_1 \mathbf{z}_k^{(1)}$, $\mathbf{y}_k = \mathbf{Q}_1 \mathbf{y}_k^{(1)}$
            compute N_k
            $\mathbf{x}_k = \mathbf{x}_k + \Delta \mathbf{x}(\mathbf{z}_k, \mathbf{y}_k, N_k)$
            check corrector stopping criteria†
      end
      if corrflag=1
        $k = k + 1$; stepcut=0; adjust Δs
        Send(1:NP-1,'$\mathbf{u}_k$'); Send(1:NP-1,'contflag')
      else
        $\Delta s = \Delta s/2$; stepcut=stepcut+1
      end
      check continuation stopping criteria‡
  end
else (Bifurcation Processor ID=1:NP-1)
  $\mathbf{u}_k$ = Recv; contflag = Recv
  while contflag $= 0$
      diagnose orthogonal blocks $\mathbf{Q}_{\mu(\mathrm{ID})}^T \mathbf{J}_k \mathbf{Q}_{\mu(\mathrm{ID})}$ for bifurcations
  end
end
if ID $= 0$
  Send(1:NP-1,'$\mathbf{u}_k$'); Send(1:NP-1,'contflag')
end

† (corrflag=1) convergence attained $||\mathbf{f}_k|| < \epsilon$
(corrflag=2) divergence detected
(corrflag=3) maximum number of iterations exceeded

‡ (contflag=1) $k >$ maximum number of allowable steps
(contflag=2) stepcut $>$ maximum number of allowable step cuts
(contflag=3) $||\lambda_k|| >$ maximum allowable load factor

Figure 9: Pseudo code for MMcontcm.m using NP processors

with average bandwidths less than or equal to 9. Besides leading to a well-conditioned numerical procedure, the *a priori* information given by Tables 2 and 3 about the size and number of blocks are of great benefit in evenly distributing work among the bifurcation processors for a load-balanced code.

A group theoretic analysis also provides the following generic "bifurcation hierarchy" which govern the types of bifurcations which can occur along a D_n-symmetric solution branch:

- if n is even, the symmetry of the bifurcation branches can be: C_n, $D_{n/2}$ or $D_{n/m}$ where $m \in [3, n]$ and $\frac{n}{m}$ is an integer;
- if n is odd, the symmetry of the bifurcation branches can be: C_n or $D_{n/m}$ where $m \in [3, n]$ and $\frac{n}{m}$ is an integer.

For example, the expected isotropy subgroups of bifurcation branches along a D_{128}-symmetric branch are D_{64}, D_{32}, D_{16}, D_8, D_4, D_2, D_1 and C_{128}. Of course, which symmetries *actually* occur in the analysis is dependent upon the physics in the problem. Table 4 displays the block structure for all generic solution branches with dihedral symmetry expected with a 41×128 mesh. As is evident in Table 4, the less symmetry a solution branch has the less benefit a group theoretic approach offers as far as allowing computation to be done in dimensionally-reduced subspaces. However, the group theoretic approach still allows one to avoid numerical ill-conditioning due to closely space bifurcation points which is crucial in some problems [32].

		continuation processor	bifurcation processor(s)				
$\frac{N}{n}$	n	$\mid V^1 \mid$	$\mid V^2 \mid$	$\mid V^3 \mid$	$\mid V^4 \mid$	$\mid V_1^j \mid$	$j = 1, 2, \ldots$
even	even	$M\left(\frac{3N}{n} + 2\right)$	$M\left(\frac{3N}{n} - 2\right)$	$\frac{3MN}{n}$	$\frac{3MN}{n}$	$\frac{6MN}{n}$	$\frac{n-2}{2}$
even	odd	$M\left(\frac{3N}{n} + 2\right)$	$M\left(\frac{3N}{n} - 2\right)$	—	—	$\frac{6MN}{n}$	$\frac{n-1}{2}$
odd	even	$M\left(\frac{3N}{n} + 1\right)$	$M\left(\frac{3N}{n} - 1\right)$	$M\left(\frac{3N}{n} + 1\right)$	$M\left(\frac{3N}{n} - 1\right)$	$\frac{6MN}{n}$	$\frac{n-2}{2}$
odd	odd	$M\left(\frac{3N}{n} + 1\right)$	$M\left(\frac{3N}{n} - 1\right)$	—	—	$\frac{6MN}{n}$	$\frac{n-1}{2}$

Table 2: Stiffness block information for D_n-symmetric solution paths for the axially-compressed cylindrical shell with N four-node nonlinear shell elements used about the circumference and $M - 1$ elements along the length. Average bandwidth for each block is $\overline{bw} \approx \frac{3|V^i|}{2M}$.

		continuation processor	bifurcation processor(s)		
$\frac{N}{n}$	n	$\mid V^1 \mid$	$\mid V^2 \mid$	$\mid V^j \mid$	$j = 1, 2, \ldots$
odd/even	even	$\frac{6MN}{n}$	$\frac{6MN}{n}$	$\frac{12MN}{n}$	$\frac{n-2}{2}$
odd/even	odd	$\frac{6MN}{n}$	—	$\frac{12MN}{n}$	$\frac{n-1}{2}$

Table 3: Stiffness block information for C_n-symmetric solution paths for the axially-compressed cylindrical shell with N four-node nonlinear shell elements used about the circumference and $M-1$ elements along the length. Average bandwidth for each block is $\overline{bw} \approx \frac{3|V^i|}{2M}$.

5.2 Computing the Jacobian matrix

The first major computational task for both the continuation and bifurcation processors is to compute the Jacobian matrix. As is typical of medium-to-large scale FE models in nonlinear structural mechanics, the Jacobian matrix $\mathbf{J}$ for the cylindrical shell is symmetric, indefinite and sparse. The number of non-zero components goes as $\sim 330MN$. In both contcm.m and MMcontcm.m, only the lower triangular part of the Jacobian matrix is computed and stored. Table 5 displays the size and sparsity of $\mathbf{J}$ as well as the approximate CPU time to compute it. As was expected, the CPU time to compute the Jacobian for the cases shown here is directly proportional to the DOF, *i.e.*, assembly time $\approx$ DOF/1570.

In this implementation of MMcontcm.m the Jacobian matrix is computed on a single processor with most of the computation being done by fortran MEX-files. It would not be difficult in future versions of the code to have a parallel assembly procedure which might lead to a speed-up for the larger DOF problems.

5.3 Transformation to Symmetry Coordinates

Once the Jacobian matrix has been assembled, the orthogonal blocks $\mathbf{J}^{(\mu)}$, $(\mu = 1, 2, \ldots, p)$ can be computed via similarity transformations as discussed in §3.4. For the overall group theoretic approach to be feasible, these transformations must done quickly and efficiently.

The structure and sparsity of the transformation matrices $\mathbf{Q}_\mu \in \mathbb{R}^{|V|\times|V^\mu|}$ varies from problem to problem. However, experience has shown that the number of nonzero components in $\mathbf{Q}_\mu$ is usually relatively small. For the cylindrical shell problem it can be shown that the maximum number of nonzeros in $\mathbf{Q}_\mu$ is bounded by $20MN$. It is interesting to note that the number $20MN$ is independent of the symmetry of the solution path which is being computed, even though the number of columns in $\mathbf{Q}_\mu$ is equal to the dimension of the symmetry subspace V^μ, whose dimension depends on the symmetry of the solution path.

The relatively small number of nonzeros in $\mathbf{Q}_\mu$ and the sparsity of the Jacobian matrix $\mathbf{J}$ leads to very efficient transformation routines. Table 6 displays the timing results for computing the orthogonal block $\mathbf{J}^{(\mu)}$, with $\mu = 5$, for the four different meshes along the primary solution

	continuation processor	bifurcation processor(s)				
isotropy group	$\mid V^1 \mid$	$\mid V^2 \mid$	$\mid V^3 \mid$	$\mid V^4 \mid$	$\mid V^j \mid$	$j = 1, 2, \ldots$
D_{128}	204	102	204	102	306	63
D_{64}	408	204	306	306	612	31
D_{32}	714	510	612	612	1224	15
D_{16}	1326	1122	1224	1224	2448	7
D_8	2050	1886	1968	1968	3936	3
D_4	4018	3854	3936	3936	7872	1
D_2	7954	7790	7872	7872	—	—
D_1	15826	15662	—	—	—	—

Table 4: Expected block structure for D_n-symmetric primary and higher order bifurcation branches off and including a D_{128}-symmetric solution path. Axially compressed cylindrical shell with: N=128, M=41. Average bandwidth for each block is $\overline{bw} \approx \frac{3|V^i|}{100}$.

mesh size	DOF	sparsity **J**	assembly time (sec)
21×80	10080	0.0054	6.3
41×80	19680	0.0027	12.5
41×128	31488	0.0018	20.3
51×144	44064	0.0012	28.3

Table 5: Jacobian size, typical sparsity and approximate CPU assembly time.

branch. The case $\mu = 5$ was chosen because it represents an upper bound with regard to size and computation for the orthogonal blocks. As is clearly demonstrated, along the primary solution branch, the time needed to compute $\mathbf{J}^{(5)}$ is an order of magnitude less than the time needed to compute $\mathbf{J}$. The matrices $\mathbf{Q}_\mu$ are computed using fortran MEX-files, but the actual similarity transformation is perform with MATLAB's sparse multiplication capabilities.

Table 7 is more detailed study of the CPU time needed to perform symmetry transformations along all the possible D_n-symmetric solution branches for the 41×128 mesh. As expected, the time needed to perform the transformations increases as the symmetry of the branch, (D_n), decreases.

mesh size	D_n	$\mathbf{J}$ (sec)	$\lvert V^{(5)} \rvert$	$\mathbf{J}^{(5)}$ (sec)
21×80	80	6.3	126	0.6
41×80	80	12.5	246	1.2
41×128	128	20.3	246	1.8
51×144	144	28.3	306	2.8

Table 6: Typical block Jacobian size ($\mu = 5$) and approximate CPU time to compute $\mathbf{J}$ and $\mathbf{J}^{(\mu)} = \mathbf{Q}_\mu^T \mathbf{J} \mathbf{Q}_\mu$ along the primary D_n-symmetric solution branch. Note that the reported CPU time for $\mathbf{J}^{(5)}$ includes the time to compute $\mathbf{Q}_5$ as well as the time needed to carry out the matrix multiplication.

	continuation processor		bifurcation processors $5 \le \mu \le \frac{n-2}{2}$								
D_n	$\lvert V^{(1)} \rvert$	$\mathbf{J}^{(1)}$ (sec)	$\lvert V^{(2)} \rvert$	$\mathbf{J}^{(2)}$ (sec)	$\lvert V^{(3)} \rvert$	$\mathbf{J}^{(3)}$ (sec)	$\lvert V^{(4)} \rvert$	$\mathbf{J}^{(4)}$ (sec)	$\lvert V^{(\mu)} \rvert$	$\mathbf{J}^{(\mu)}$ (sec)	Number of Blocks
128	164	1.3	82	1.0	164	1.3	82	1.0	246	1.8	66
64	328	1.9	164	1.2	246	1.6	246	1.6	492	2.9	34
32	574	2.8	410	2.2	492	2.6	492	2.6	984	5.3	18
16	1066	3.5	902	3.4	984	3.5	984	3.5	1968	7.9	10
8	2050	4.2	1886	4.1	1968	4.2	1968	4.2	3936	10.7	6
4	4018	5.2	3854	5.2	3936	5.2	3936	5.2	7872	15.0	4
2	7954	7.0	7790	7.0	7872	6.8	7872	6.8	—	—	3
1	15826	10.5	15662	10.5	—	—	—	—	—	—	1

Table 7: Block Jacobian size and approximate CPU time to transform the full Jacobian matrix $\mathbf{J}$ to symmetry coordinates: 41×128 mesh.

5.3.1 Direct Computation of $\mathbf{J}^{(\mu)}$ via Automatic Differentiation

While using similarity transformations to compute the orthogonal blocks $\mathbf{J}^{(\mu)}$ has worked well in this and previous studies, the procedure does require that the full Jacobian matrix $\mathbf{J}$ be assembled first. When $m_\mu = |V^{(\mu)}| \ll n$ this is somewhat of a brute force approach. For example, in the case of the 51×144 mesh in Table 6, computing the 306×306 Jacobian block $\mathbf{J}^{(5)}$, (average bandwidth of 10), required that we first assemble the 44064×44064 Jacobian matrix $\mathbf{J}$.

A very promising topic for future research is to make use of recent advances in using automatic differentiation (AD) [51], [52], to compute the symmetry-adapted Jacobian blocks $\mathbf{J}^{(\mu)}$ in a much more direct manner. It can be shown that the orthogonal blocks $\mathbf{J}^{(\mu)}$ can be "directly" computed as

$$\mathbf{J}^{(\mu)} = \mathbf{Q}_\mu^T \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{u}}_\mu} \tag{49}$$

where $\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{u}}_\mu}$ is an $n \times m_\mu$ matrix whose columns are defined as

$$\left(\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{u}}_\mu}\right)_i = \lim_{h\to 0} \frac{\mathbf{f}(\mathbf{u} + h\mathbf{Q}_\mu(:,i)) - \mathbf{f}(\mathbf{u})}{h} \quad (i = 1, 2, \ldots, m_\mu). \tag{50}$$

Note that eq. (50) only requires that $\mathbf{f}$ be computed in the standard coordinate system. Equations (49) and (50) can be used for the basis of a numerical differentiation procedure, as discussed in [51] and [52], to efficiently compute the orthogonal blocks $\mathbf{J}^{(\mu)}$.

5.4 Symmetry Boundary Conditions

As discussed above, the symmetry-adapted Jacobian matrices are found by first computing the Jacobian of the full matrix and then performing similarity transformations. For large problems, computing the full Jacobian can be relatively expensive and it may be necessary to compute it several times per solution step. In an effort to speed-up the continuation portion of the procedure, it is often possible to define a separate problem for the continuation process which can lessen the overall amount of work.

To understand this, consider that if the discrete model of a physical system has symmetry, the model is by definition composed of a finite number of "repeating" substructures. A well known technique to simplify the computations in an analysis of a structure with symmetry is to model a single substructure or an assemblage of several substructures. The analysis on the substructure(s) leads to a dimensionally reduced problem which lessens computational effort. As part of the solution process special linear constraints, known as *symmetry boundary conditions*, must be applied to the reduced problem. These constraints ensure that the reduced problem captures solutions which "correspond" to solutions of the full problem and generally take the form $c_1u_1 + c_2u_2 + \ldots + c_nu_n = c_0$.

It is often the case that the solutions which can be obtained by using symmetry boundary conditions are in 1-1 correspondence with solutions obtained by restricting the analysis of the "whole" structure to one or more of the symmetry-adapted subspaces [32]. For example, the three-bar truss in Fig. 4 can be thought of as an assemblage of three substructures: bars $\mathrm{A_0A_1}$, $\mathrm{A_0A_2}$ and $\mathrm{A_0A_3}$. When the load is first applied the deformed truss will maintain its original equilateral triangular shape, *i.e.,* D_3-symmetry. Using a group theoretic approach the D_3-symmetric primary solution path would be computed by first assembling the global 3×3 Jacobian matrix associated with all three bars and all three degrees-of-freedom (u_1, u_2, u_3) and transforming to the one-dimensional D_3-invariant subspace which is spanned by the vector $(0, 0, 1)$ [53]. On the other hand, the exact same D_3-symmetric primary solution path could be computed by performing an analysis on the *single* bar $\mathrm{A_0A_1}$ by enforcing the boundary conditions $u_1 = u_2 = 0$.

The advantage of using symmetry boundary conditions as opposed to similarity transformations is that a dimensionally reduced Jacobian matrix used in the continuation process can be computed directly without having to first compute the Jacobian of the full problem and then transforming to symmetry coordinates — for problems associated with a "large" symmetry group this can lead to substantial savings. On the other hand, this method does entail extra work for the user because the proper symmetry boundary conditions must be determined.

As an example of the potential speed-up, Table 8 compares the time needed to compute a Newton step using the standard symmetry transformations and using a constrained system for the cylindrical shell problem. A 41 $\times$ 128 mesh mesh was used in both cases and the linear constraints are exactly enforced using the method of Lagrange multipliers. While both methods give identical numerical results, the procedure with the constrained system is quicker than the procedure with the symmetry transformations in almost all cases. The speed-up along branches with D_{128}, D_{64}, D_{32} and D_{16}-symmetry is substantial. The results for the D_{128}-symmetric branch is very significant because this represents an analysis along the *primary solution branch* which provides important information needed to understand the axially compressed cylindrical shell. In fact, it is often the case in the technical literature that a *partial* analysis of the primary solution branch represents a significant portion of the numerical work.

	symmetry transformations				constrained system		
D_n	$\mathbf{J}$ (sec)	$\mathbf{J}^{(1)}$ (sec)	$\mathbf{J}^{(1)} \setminus \mathbf{f}^{(1)}$ (sec)	Newton Step (sec)	$(\mathbf{J}_c)$ (sec)	$(\mathbf{J}_c) \setminus \mathbf{f}_c$ (sec)	Newton Step (sec)
128	20.3	1.3	0.03	21.6	0.3	0.3	0.6
64	20.3	1.9	0.1	22.3	0.2	0.3	0.5
32	20.3	2.8	0.4	23.5	0.4	0.8	1.2
16	20.3	3.5	1.6	25.4	0.7	4.4	5.2
8	20.3	4.2	5.4	29.9	1.4	17.7	19.2
4	20.3	5.2	14.3	39.8	2.9	67.0	69.9
2	20.3	7.0	64.0	91.3	5.2	66.0	71.2
1	20.3	10.5	—	—	10.3	—	—

Table 8: Comparison of CPU time needed to compute a Newton step for the continuation process using symmetry transformations versus using a constrained system of equations: 41 $\times$ 128 mesh.

5.5 Example Run: Primary Solution Branch

To demonstrate the efficiency of MMcontcm.m we will present results from a typical nonlinear bifurcation analysis of the primary solution branch of an axially compressed cylindrical shell. The cylinder considered in this analysis was modeled with simple-support (SS-3) boundary conditions on both ends and the following geometric and material parameters: L=200 in., R=100 in., t=1 in., E=10^6 psi and $\nu = 0.3$. Four different meshes were used: 21×80, 41×80, 41×128 and 51×144.

The bifurcation analyses were done for a normalized load range of $\lambda = 0.8\lambda_{\text{crit}}$ and $\lambda = \lambda_{\text{crit}}$. where λ_{crit} is the "classical" critical buckling load

$$\lambda_{\text{crit}} = \frac{2\pi \mathrm{E}t^2}{\sqrt{3(1-\nu^2)}} = 114 \cdot 10^6 \text{ lbs.}$$

This range of the loading was chosen because it is well known that this is where most of the important symmetry breaking bifurcations along the primary solution branch will occur.

MMcontcm.m has several features which were designed to help the user make efficient use of the available processors:

- For the continuation part of the process, MMcontcm.m allows the user to choose either symmetry transformations or a constrained system of equations as discussed in §5.4. By choosing the fastest of the two approaches for a given branch one can minimize or even eliminate the amount of time the bifurcation processors are idle.
- The user can choose the number of processors which will be used for bifurcation analysis.
- In physical problems the analyst can sometimes rule out certain blocks as a potential source for bifurcation based upon physics and the symmetry associated with a given block. MMcontcm.m lets the analysts choose which blocks will be assembled and diagnosed for bifurcation.
- MMcontcm.m lets the user choose which blocks will be assembled on what processors. Given the *a priori* information about the size of individual blocks, *e.g.,* Tables 2, 3 and 4, the capability to assign individual processors specific blocks allows for a load-balanced process.

For a given analysis, the number of processors used and how the work is distributed among the available processors is problem dependent. For each mesh analyzed in this example, the continuation processor used a constrained system of equations to compute the primary solution branch. Twenty-five solution points were computed between $\lambda = 0.8\lambda_{\mathrm{crit}}$ and $\lambda = \lambda_{\mathrm{crit}}$, giving a relatively detailed look at the solution curve. For the bifurcation analysis, six processors were used for assembling and diagnosing blocks $\mathbf{J}^{(5)} - \mathbf{J}^{(16)}$ with two blocks per processor. The approximate size and assembly time for blocks $\mathbf{J}^{(5)} - \mathbf{J}^{(16)}$ for each mesh can be found in Table 6.

Table 9 and Fig. 10 display important timing and load balancing information from the four analyses. In Table 9 it is clear that for this example, the continuation part of the procedure is significantly faster than the bifurcation analysis — the end result being that the bifurcation processors are idle for an insignificant amount of time. The percent idle time in Table 9 is primarily a measure of the communication time between the continuation processor and the bifurcation processors. Figure 10 is a plot of the CPU time needed by each bifurcation processor to complete its particular analysis for the four cases run. Clearly, there was very good load balancing achieved in all four analyses.

mesh size	continuation processor CPU time (sec)	average bifurcation processor CPU time (sec)	Maximum % Idle Time
21×80	8.8	235	0.02
41×80	24.4	593	0.04
41×128	24.2	833	0.01
51×144	31.4	1263	0.08

Table 9: Timing results for the nonlinear bifurcation analysis of the cylindrical shell for four different meshes.

Finally, to demonstrate why problems such as the axially loaded cylindrical shell are numerically intractable without a group theoretic approach, the forty-two smallest eigenvalues for blocks $\mathbf{J}^{(5)} - \mathbf{J}^{(16)}$ as function of the normalized axial load for the 144×51 mesh are plotted in Fig. 11. Each crossing of the y-axis in Fig. 11 represents a bifurcation point and a singularity in the full

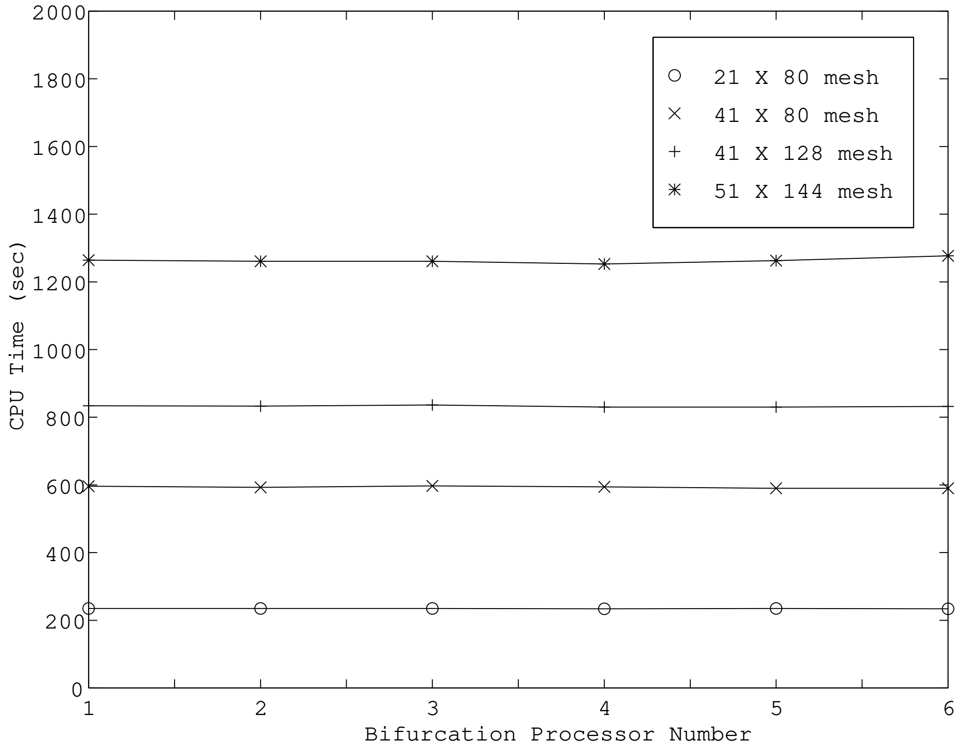

Figure 10: CPU time for each bifurcation processor for four different meshes.

Jacobian. With a group theoretic approach the continuation processor never sees any of the bifurcation points and has no difficulty is computing the primary solution branch.

6 Conclusions

In this work, a parallel group theoretic nonlinear continuation/bifurcation analysis code MMcontcm has been discussed. With MMcontcm, we have demonstrated that a parallel computational environment is a natural setting for the bifurcation analysis of a very important class of nonlinear equilibrium equations which display group invariance due to symmetry. The natural parallelism stems from a global de-coupling of the governing equilibrium equations made possible by the proper choice of a symmetry-adapted basis — which is systematically derived using well known results from group representation theory.

A key element in a group theoretic approach to a nonlinear continuation/bifurcation analysis is the reduced problem, (eq. (39)). The reduced problem is defined by restricting the full problem onto a proper subspace $V^{(1)}$ which contains exact global solutions of a specific symmetry type. This restriction leads to a well-conditioned continuation procedure which generically only needs to be able to compute solution branches through limit points. The bifurcation points which could cause severe ill-conditioning, *e.g.,* Fig. 11, are associated with the orthogonal $\mathbf{J}^{(\mu)}$, $(\mu = 2, 3, \ldots, p)$, which are never used by the continuation processor.

The fact that the continuation procedure is independent of the bifurcation analysis and that the bifurcation analyses of the individual orthogonal blocks $\mathbf{J}^{(\mu)}$ are independent of each other led to the parallel algorithm MMcontcm which is summarized in Fig. 9. Some important observations about the general group theoretic approach and its implementation in a parallel computational

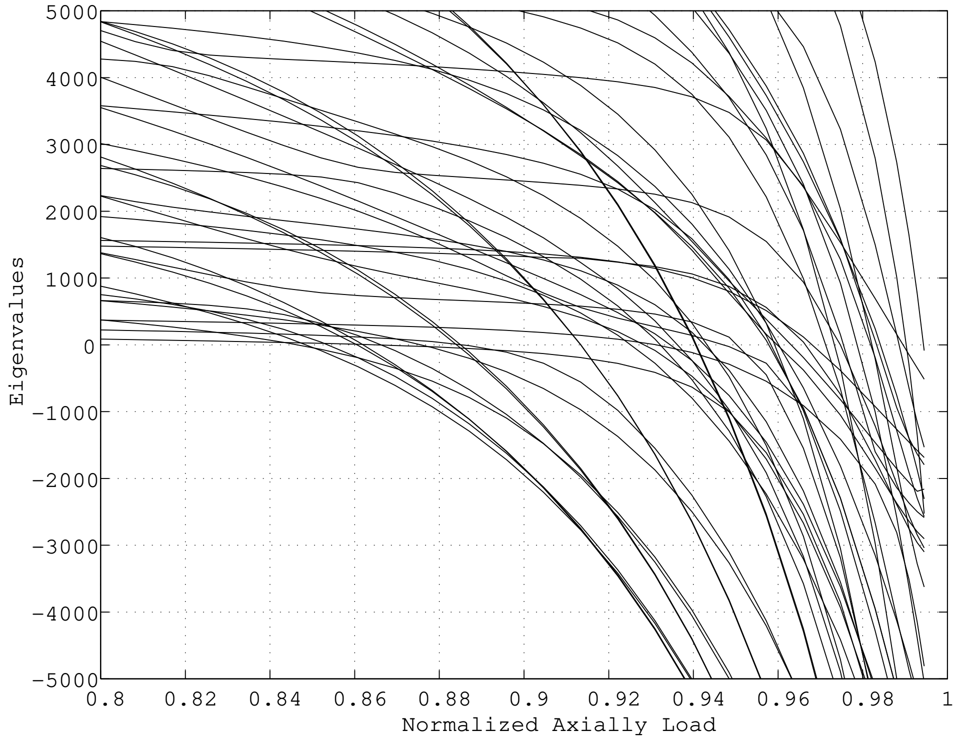

Figure 11: Forty-two smallest eigenvalues for blocks $\mathbf{J}^{(5)} - \mathbf{J}^{(16)}$ as a function of the normalized axial load. (144 $\times$ 51 mesh)

environment are:

- The only significant communication between processors occurs when the continuation processor converges to a solution point $\mathbf{x}_k$ and sends it to the bifurcation processors. The minimal communication guarantees a coarse-grained application.

- A group analysis of the problem being studied provides complete information on the number and sizes of the Jacobian blocks $\mathbf{J}^{(\mu)}$, *e.g.,* Tables 2-4. This *a priori* information allows the analyst to assign individual processors specific blocks in order to achieve a load-balanced process, (Fig. 10).

- It is sometimes possible to define an "equivalent" constrained set of dimensionally-reduced equations, as opposed to using the symmetry transformations, which can lead to dramatic speed-up in computing the Newton step for the continuation procedure, (Table 8). This helps to insure that the bifurcation processors are idle for a minimum amount of time.

In MMcontcm, we have laid a solid foundation upon which further research topics can be pursued. One very promising topic for future research, discussed in §5.3.1, is the use of automatic differentiation to "directly" compute the orthogonal blocks $\mathbf{J}^{(\mu)}$. Another promising area of research is developing *symmetry-motivated preconditioners* for iterative solution methods for problems with little or no symmetry. As discussed earlier, the more symmetry that a solution path has the greater the benefits of a group analysis. On the other hand, in a structural mechanics context the static/dynamic response of a structure with very little symmetry is often dominated by terms which come from a larger symmetry group. For example, while the underlying symmetry of an aircraft fuselage is that of a body of revolution, which is associated with the "large" symmetry group O(2), the interior and exterior components such as decks and wings reduce the symmetry of the whole structure to perhaps a simple reflection, which represents a "small" symmetry group. However, the dominant terms in the Jacobian matrix would be strongly tied to the underlying O(2)-symmetry. Therefore, it is reasonable to expect that the dimensionally reduced Jacobian blocks associated with the O(2)-symmetry, or some small subset of them, could serve as a physically meaningful basis for assembling an efficient preconditioner for a conjugate gradient method.

7 Acknowledgements

I am indebted to Thomas F. Coleman for his input and encouragement as well for providing the opportunity and facilities to perform this work. I also wish to thank Anne Trefethen and Vijay Menon for their generous assistance with MultiMATLAB.

References

[1] T. J. Healey. *Symmetry, Bifurcation and Computational Methods in Nonlinear Structural Mechanics.* PhD thesis, University of Illinois, Champaign, IL, 1985.

[2] H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P. Rabinowitz, editor, *Applications of Bifurcation Theory*, pages 359–384. Academic Press, 1977.

[3] E. Riks. Some computational aspects of the stability analysis of nonlinear structures. *Comp. Meth. Appl. Mech. Engin.*, 47:219–259, 1984.

[4] R. Seydel. Tutorial on continuation. *Int. J. Bifur. and Chaos*, 1:3–11, 1991.

[5] D. H. Sattinger. *Group Theoretic Methods in Bifurcation Theory.* Springer-Verlag, 1979.

[6] A. Fässler and E. Stiefel. *Group Theoretical Methods and Their Applications.* Birkhäuser, Boston, 1992.

[7] M. Golubitsky and D. G. Schaeffer. *Singularities and Groups in Bifurcation Theory Vol. I.* Springer-Verlag, 1984.

[8] M. Golubitsky, I. Stewart, and D. G. Schaeffer. *Singularities and Groups in Bifurcation Theory Vol. II.* Springer-Verlag, 1985.

[9] A. Vanderbauwhede. *Local Bifurcation and Symmetry.* Pitman, 1982.

[10] T. J. Healey. Numerical bifurcation with symmetry: Diagnosis and computation of singular points. In M. Golubitsky L. Kaitai, J. Marsden and G. Iooss, editors, *Proceedings of the International Conference on Bifurcation Theory and its Numerical Analysis*, pages 218–227, Xi'an China, 1988.

[11] T. J. Healey. A group theoretic approach to computational bifurcation problems with symmetry. *Comp. Meth. Appl. Mech. Engin.*, 67:257–295, 1988.

[12] T. J. Healey. Global bifurcation and continuation in the presence of symmetry with an application to solid mechanics. *Siam J. Math. Anal.*, 19(4):824–840, 1988.

[13] T. J. Healey and J. A. Treacy. Exact block diagonalization of large eigenvalue problems for structures with symmetry. *Int. J. Numer. Meths. Engin.*, 31:265–285, 1991.

[14] K. Ikeda and K. Murota. Bifurcation analysis of symmetric structures using block diagonalization. *Comp. Meth. Appl. Mech. Engin.*, 86:215–243, 1991.

[15] K. Ikeda, K. Murota, and H. Fujii. Bifurcation hierarchy of symmetric structures. *Int. J. Sol. Struct.*, 27(12):1551–1573, 1991.

[16] K. Murota and K. Ikeda. Computational use of group theory in bifurcation hierarchy analysis of symmetric structures. *Siam Journal of Scientific and Statistical Computing*, 12(2):273–297, 1991.

[17] P. Stork and B. Werner. Symmetry adapted block diagonalisation in equivariant steady state bifurcation problems and its numerical application. *Advances in Mathematics*, 20(4):455–488, 1991.

[18] K. Gatermann and A. Hohmann. Symbolic exploitation of symmetry in numerical pathfollowing. *Impact of Computing in Science and Engineering*, 3:330–365, 1991.

[19] B. Werner and A. Spence. The computation of symmetry-breaking bifurcation points. *SIAM Journal of Numerical Analysis*, 21(2):388–399, 1984.

[20] M. Dellnitz and B. Werner. Computational methods for bifurcation problems with symmetries, with special attention to steady state and hopf bifurcation points. *Journal of Computational and Applied Mathematics*, 26:97–123, 1989.

[21] T. J. Healey and H. Kielhöfer. Symmetry and nodal properties in the global bifurcation analysis of quasi-linear elliptic equations. *Arch. Rat. Mech. Anal.*, 113:299–311, 1991.

[22] H. J. Weinitschke. On the calculation of limit and bifurcation points in stability problems of elastic shells. *Int. J. Sol. Struct.*, 21(1):79–95, 1985.

[23] P. J. Aston. Analysis and computation of symmetry-breaking bifurcation and scaling laws using group theoretic methods. *Siam J. Math. Anal.*, 22(1):181–212, 1991.

[24] J. C. Wohlever and T. J. Healey. A group theoretic approach to the global bifurcation analysis of an axially-compressed cylindrical shell. *Comp. Meth. Appl. Mech. Engin.*, 122:315–349, 1995.

[25] A. Bossavit. Symmetry, groups and boundary value problems. a progressive introduction to noncommutative harmonic analysis of partial differential equations in domans with geometrical symmetry. *Comp. Meth. Appl. Mech. Engin.*, 56:167–215, 1986.

[26] T. M. Whalen. *On the Existence of Periodic Solutions to Nonlinear Elastodynamic Systems with Symmetry.* PhD thesis, Cornell University, Ithaca, N.Y., 1993.

[27] E.L. Allgower, K.Böhmer, K. Georg, and R. Miranda. Exploiting symmetry in boundary element methods. *Siam J. Num. Anal.*, 29:534–552, 1992.

[28] G. G. Hall. *Applied Group Theory.* Longmans, 1967.

[29] V. E. Hill. *Groups, Representations and Characters.* Hafner Press, 1975.

[30] W. Miller. *Symmetry, Groups and Their Applications.* Academic Press, 1972.

[31] J. Serre. *Linear Representations of Finite Group.* Springer-Verlag, 1977.

[32] J. C. Wohlever. A group theoretic approach to the global bifurcation analysis of symmetric structures using a standard finite element package. *Comp. Meth. Appl. Mech. Engin.*, 1996. submitted.

[33] B. H. Kröplin. Postbuckling instability analysis of shells using the mixed method. In E. Ramm, editor, *Buckling of Shells: Proceedings of a State-of-the-Art Colloquium, Universität Stuttgart, Germany*, pages 175–200. Springer-Verlag, 1982.

[34] W. K. Liu, E. S. Law, D. Lam, and T. Belytschko. Resultant-stress degenerated-shell element. *Comp. Meth. Appl. Mech. Engin.*, 55:259–300, 1986.

[35] A. Maewal and W. Nachbar. Stable postbuckling equilibria of axially compressed, elastic circular cylindrical shells: A finite element analysis and comparison with experiments. *J. Appl. Mech.*, 7:475–481, 1977.

[36] J. C. Simo, D. D. Fox, and M. S. Rifai. Geometrically exact stress resultant shell models: Formulation and computational aspects of the nonlinear theory. In *State of the Art Surveys on Computational Mechanics*, pages 161–190. ASME Publications, 1989.

[37] J. C. Simo, D. D. Fox, and M. S. Rifai. On a stress resultant geometrically exact shell model. part ii: The linear theory; computational aspects. *Comp. Meth. Appl. Mech. Engin.*, 73:53–92, 1989.

[38] D. W. Lloyd. The mechanics of drape. In E.L. Axelrad and F.A. Emmerling, editors, *Flexible Shells: Theory and Applications*, pages 271–282, Berlin, 1984. Springer-Verlag.

[39] J. T. Jenkins. Static equilibrium configurations of a model red blood cell. *J. Math. Bio.*, 4:149–169, 1977.

[40] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, 1983.

[41] E. L. Allgower and K. Georg. *Numerical Continuation Methods: An Introduction.* Springer-Verlag, Heidelberg, 1990.

[42] C. Den Heijer and W.C. Rheinboldt. On steplength algorithms for a class of continuation methods. *Siam J. Num. Anal.*, 18:925–948, 1981.

[43] H. B. Keller. Practical procedures in path following near limit points. In R. Glowinski and J.L. Lions, editors, *Computing Methods in Applied Sciences and Engineering*, volume 5, pages 359–384. North-Holland Publishing Co., 1982.

[44] D. Bushnell. *Computerized Buckling Analysis of Shells.* Martinus Nijhoff, 1985.

[45] W. Kaplan. *Advanced Calculus.* Addison-Wesley, Reading, MA, 1984.

[46] E. Kreyszig. *Differential Geometry.* Dover, Mineola, NY, 1991.

[47] K. Bathe and E. Wilson. *Numerical Methods in Finite Element Analysis.* Prentice Hall, 1976.

[48] T. J. Healey. Symmetry and equivariance in nonlinear elastostatics: Part i. *Arch. Rat. Mech. Anal.*, 105(3):205–228, 1989.

[49] J. Liu and G. Wu. An efficient finite element solution method for analysing large-scale symmetric structures. In O. O. Storaasli and J. M. Housner, editors, *Computer Systems in Engineering: Parallel Computational Methods for Large-Scale Structural Analysis and Desgn*, volume 4(4-6), pages 373–380, Great Britain, 1994. Pergamon.

[50] O. C. Zienkiewicz. *The Finite Element Method Fourth Edition: Volume 2 Solids and Fluid Mechanics, Dynamics and Non-Linearity.* McGraw-Hill, Great Britain, 1991.

[51] T. F. Coleman and A. Verma. The efficient computation of sparse jacobian matrices using automatic differentiation. Tech. Report TR95-1557, Computer Science Department, Cornell University, November 1995.

[52] T. F. Coleman and A. Verma. Structure and efficient jacobian calculation. In M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors, *Computational Differentiation: Techniques, Applications and Tools*, pages 149–159, Philadelphia, 1996. SIAM.

[53] J. C. Wohlever. *A Group Theoretic Approach to the Nonlinear Bifurcation Analysis of Shells of Revolution.* PhD thesis, Cornell University, Ithaca, NY, 1993.