23

Detecting and Traversing Critical Points

TABLE OF CONTENTS

§23.1. Introduction

Critical points were defined and mathematically characterized in Chapter 5. Chapter 18 explains how solution procedures of purely incremental type can be adjusted to traverse limit points, and difficulties in traversing bifurcation points are noted. Chapters 24 and follwing discuss critical points in the context of stability analysis.

The present Chapter deals with the determination or detection of critical points in statically nonlinear analysis and with procedures for traversal of such points with incremental/corrective techniques. Emphasis is given to handling isolated bifurcation points, which offer moderate degree of difficulty: tougher than limit points, easier than multiple bifurcation. Much of the material that follows is taken from Chapter 5 of Crivelli's thesis.[1]

There are two basic approaches to the problem of detecting critical points. We may attempt to *find* critical points by solving an algebraic equation which has those point as a root, or we can try to *detect* critical points as we are marching along an equilibrium path.

The first approach embodies what are collectively called *direct methods*. As the name implies, these methods look for critical points without being concerned, at least theoretically, with tracing equilibrium paths up to those points.

The second approach embodies the so-called *indirect methods*. In these methods the detection of a critical point is intimately related to the continuation procedure.

Remark 23.1. The simplest example of a direct method is the Linearized PreBuckling (LPB) analysis discussed in Chapters 24 and 25. This procedure, when applicable, sets up directly an eigenproblem in the reference configuration and thus avoids tracing the nonlinear response.

§23.2. Direct Methods and Test Functions

A *direct method* for calculating bifurcation points consists of formulating a suitable set of equations which has the critical points as solutions. In the context of general nonlinear analysis (that is, when no *a priori* simplifying assumptions are made as in the LPB analysis) this set of equations should include the equilibrium equation because only critical points on the equilibrium path are of interest.

It follows that direct formulations are achieved by augmenting the residual force equilibrium equation with a set of constraints that characterize the critical point. The main characteristic of this procedure is that it does not require, at least theoretically, equilibrium path tracing. Thus, ideally it should be possible to obtain all the critical points first and then join them with equilibrium paths. Except for certain specialized situations this is not possible, however, because the solution of the nonlinear equations resulting from this approach requires a good initial guess. Since these equations are nonlinear, they cannot be solved directly, and linearization coupled with an iterative procedure is required. This linearization accounts for most of the cost of direct methods.

As mentioned above, the procedure consists of augmenting the original residual force equation by a set of constraints that define a critical point. This characterization of critical points is intimately

[1] L. A. Crivelli, A Total-Lagrangian Beam Element for the Analysis of Nonlinear Space Structures, Ph. D. Dissertation, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO, April 1991.

related to the idea of constructing a *critical point test function* or CPTF. By definition, a CPTF is a function whose zeros are the critical points of the response.[2]

§23.2.1. Determinant as CPTF

Since the tangent stiffness matrix is singular at critical points, the most obvious test function is the determinant of **K**. To be able to use this function, however, in the context of a Newton-Raphson iterative procedure we need expressions for the partial derivatives of the determinant with respect to the state parameters. This approach is generally impractical for several reasons:

1. Analytical expressions for the determinant are hopelessly complicated for any but trivial problems.

2. The estimation of these derivatives by one-sided finite differences would be enormously expensive, requiring the assembly and factorization of $N+1$ tangent stiffness factorizations for N degrees of freedom[3]

3. The detrminant is a notoriously ill-behaved function.

§23.2.2. Lowest Eigenvalue as CPTF

A critical point can also be characterized by checking for a null eigenvalue of **K**. Thus a prototype of direct methods, as described in Seydel[4] may be expressed generally as

$$\begin{bmatrix} \mathbf{r}(\mathbf{u},\lambda) \\ \mathbf{K}(\mathbf{u},\lambda)\mathbf{z} \\ \phi(\mathbf{z}) \end{bmatrix} = \mathbf{0}. \tag{23.1}$$

The last equation, $\phi(\mathbf{z}) = 0$, has been added as a constraint on $\mathbf{z}$ to rule out the trivial solution $\mathbf{z} = \mathbf{0}$ and to make the system determinate. There are several possible choices for the normalization equation, the simplest being to require $\mathbf{z}$ to have unit length: $\mathbf{z}^T\mathbf{z} = 1$.

To solve this equation iteratively, define the augmented incremental vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{d} \\ \eta \\ \mathbf{h} \end{bmatrix} \tag{23.2}$$

where **d** and η are the corrective changes in **u** and λ, respectively, defined in Chapter 17, and and $\mathbf{h} = \mathbf{z}^{k+1} - \mathbf{z}^k$. is the corrective change in **z**. Assuming that **K** does not depend on λ and that the derivative of **K** with respect to **u**, denoted by $\mathbf{K}_v$, is commutative with respect to its two outermost indices, the resulting Newton system is

$$\begin{bmatrix} \mathbf{K} & \mathbf{q} & \mathbf{0} \\ \mathbf{K}_v\mathbf{z} & \mathbf{0} & \mathbf{K} \\ \mathbf{0} & \mathbf{0} & \mathbf{a}^T \end{bmatrix} \mathbf{y} = -\begin{bmatrix} \mathbf{r} \\ \mathbf{r}_\lambda \\ \phi \end{bmatrix} \tag{23.3}$$

[2] The name “branching test function” is sometimes restricted to test functions that catch only bifurcation points. Here we shall use the generic term “critical test function” to embody *all* critical points.

[3] That of the tangent stiffness **K**, plus N tangent stiffnesses corresponding to the perturbation of each state vector component.

[4] R. Seydel, *From Equilibrium to Chaos*, Elsevier, New York, 1988.

where $\mathbf{r}$ is the residual from the equilibrium equations, $\mathbf{r}_\lambda$ is the residual from the eigenvalue problem and ϕ measures the violation of the normalization condition.

This approach has been explored by Wriggers and Simo[5] in the context of nonlinear structural analysis. As can be seen from the previous equation the Jacobian of this augmented system is neither sparse nor symmetric, and has large storage requirements. In addition, the coefficient matrix becomes singular at bifurcation points, and hence ill-conditioned in their neighborhood.

§23.2.3. Solving the Minimum-Eigenvalue CPTF System

It is desirable to solve (21.3) while taking advantage of the symmetry of $\mathbf{K}$. This can be done by using the concept of auxiliary systems. Assuming $\mathbf{K}$ is nonsingular, consider the five auxiliary systems:

$$\mathbf{K}\mathbf{d}_r = -\mathbf{r}, \quad \mathbf{K}\mathbf{d}_q = \mathbf{q}, \quad \mathbf{K}\mathbf{d}_h = -\mathbf{r}_\lambda, \tag{23.4}$$

$$\mathbf{K}\mathbf{d}_{hr} = \mathbf{K}_v\mathbf{z}\mathbf{d}_r, \quad \mathbf{K}\mathbf{d}_{hq} = \mathbf{K}_v\mathbf{z}\mathbf{d}_q. \tag{23.5}$$

Then the solution to the system can be written as:

$$\eta = -\frac{\phi - \mathbf{a}^T\mathbf{d}_{hr}}{\mathbf{a}^T\mathbf{d}_{hq}}, \quad \mathbf{d} = \mathbf{d}_r + \eta\mathbf{d}_q, \quad \mathbf{h} = \mathbf{d}_h - \mathbf{d}_{hr} + \eta\mathbf{d}_{hq}. \tag{23.6}$$

The solution of these five systems at each corrective iteration is expensive. However, the main disadvantage of this approach is having to compute the $\mathbf{u}$-derivative of the stiffness matrix. Since an exact derivation of $\mathbf{K}_v$ is only practically feasible for simple nonlinear problems, some investigators (*e.g.* Seydel and Wriggers-Simo, *loc. cit.*) propose using numerical differentiation to compute $\mathbf{K}_v$. In this context it should be observed that only the product $\mathbf{K}_v\mathbf{z}$ (a square matrix) is required.

Numerical experiments reported in Seydel's book show that the convergence rate of this approach depends on whether the critical point is a limit or a bifurcation point. At limit points, the Jacobian of the augmented equation is nonsingular, which guarantees fast local convergence. At a bifurcation point, the vector $[\,\mathbf{z}^T \quad \mathbf{0}^T \quad 0\,]^T$ is a left null eigenvector, which shows that the Jacobian becomes singular. Thus, slower convergence rates may be expected at bifurcation points. However, experience shows that for some problems the convergence rate is good for the λ component when a not very stringent accuracy is required, as can be seen from the examples presented in Wriggers and Simo.

Remark 23.2. If the critical point is known to be a limit point, the penalty spring augmentation method described below is much simpler and inexpensive. And even the simplistic random perturbation scheme described there often works. Thus it appears that the direct method is advantageous in special instances, for example when direct determination of the CP locations as stability envelope, is desirable, *and* a good estimation of that location is available. This occurs in some applications such as optimization with stability constraints.

[5] P. Wriggers and J.C. Simo, A General Procedure for the Direct Computation of Turning and Bifurcation Points, *Int. J. Numer. Meth. Engrg.*, **30**, 155–176, 1990.

§23.3. Indirect Methods

The underlying idea behind indirect methods is to recognize the occurrence of a critical point while following an equilibrium path by continuation. The key to success is having a good CPTF τ. We defer the discussion on how to construct such a CPTF to §21.4, and for the moment assume that such a function is available.

These methods benefit from data collected from the continuation procedure. A critical point is characterized by being a zero of the CPTF. Because hitting an exact zero is highly unlikely, we check for a critical point by invoking the condition

$$\tau(\mathbf{u}^{k+1},\lambda^{k+1})\ \tau(\mathbf{u}^{k},\lambda^{k}) < 0 \tag{23.7}$$

This is called a *bracketing* or *straddling* condition. It means that a critical point occurs in the interval or bracket $[\lambda^k \le \lambda \le \lambda^{k+1}]$. The next task is to "zero in" this point by reducing the bracket size.

§23.3.1. Refined Determination of a Critical Point

Assume a critical point has been located by verifying bracketing condition (21.7) on the CPTF. Further, assume that the test function is a continuous function of λ and has only one isolated zero in the interval. This condition requires using a sensible incremental step — which is not known *a priori*. However, the problem of determining an appropriate step may be solved by a combination of intuition and an adaptive incremental control strategy, which monitors the convergence behavior of the solution.

Once the algorithm finds that a critical point is bracketed or straddled by two equilibrium solutions, we may obtain a first approximation to such point by using linear interpolation:

$$\lambda_c \simeq \lambda^k + \frac{\lambda^{k+1}-\lambda^k}{\tau^k-\tau^{k+1}}\tau^k \tag{23.8}$$

We present below an adaptive method developed in Crivelli's thesis (*loc. cit.* to improve the initial approximation (21.8). It makes use of iterative inverse interpolation and divided differences.[6] We have computed a set of m values of the CPTF at m unequally spaced values of the control parameter λ. We seek a polynomial interpolation for τ of the form

$$\tau(\lambda) = c_0 + c_1(\lambda-\lambda_0) + \cdots + c_m(\lambda-\lambda_0)\ldots(\lambda-\lambda_{m-1}) \tag{23.9}$$

On defining the sequence of *divided differences*

$$\begin{aligned}
\tau[\lambda_0;\lambda] &= \frac{\tau(\lambda)-\tau(\lambda_0)}{\lambda-\lambda_0}\\
\vdots\qquad &\qquad\vdots\\
\tau[\lambda_0,\ldots,\lambda_{k-1};\lambda] &= \frac{\tau[\lambda_0,\ldots,\lambda_{k-2};\lambda]-\tau[\lambda_0,\ldots,\lambda_{k-2};\lambda_{k-1}]}{\lambda-\lambda_{k-1}}
\end{aligned} \tag{23.10}$$

[6] For background in such techniques, see for example G. Dahlquist and A. Björck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, 1974.

the coefficients in equation (8) can be computed as

$$\begin{aligned} c_0 &= \tau(\lambda_0) \\ c_1 &= \tau[\lambda_0; \lambda_1] \\ \vdots \quad & \quad \vdots \\ c_k &= \tau[\lambda_0, \ldots, \lambda_{k-1}; \lambda_k] \end{aligned} \tag{23.11}$$

Observe that this formula is recursive and a new point can be added without great difficulty. Now the problem of locating a critical point is reduced to finding a zero of (21.9).

Remark 23.3. Observe that we may as well use equation to interpolate λ as a function of τ and obtain λ_c just by setting $\tau = 0$. However, both procedures are fundamentally different and may give different results. For instance, for very sharp limit points and for bifurcation points the test function could stay almost constant for most of the continuation and change to a steep slope while very close to the limit point. This type of behavior can be regarded as an exponential relation between τ and λ. Such an exponential can be well approximated by a polynomial, whereas its inverse —a logarithmic function— may pose severe difficulties. Thus we may expect faster convergence from the former procedure.

Based on the foregoing remark we choose inverse interpolation for finding a zero of (21.9). For this we recast that equation into the form

$$\lambda_c = \psi(\lambda_c) \tag{23.12}$$

or, more explicitly

$$\lambda_c = \lambda_0 - \frac{1}{c_1}\left[c_0 + \cdots + c_m\,(\lambda_c - \lambda_0)\ldots(\lambda_c - \lambda_{m-1})\right] \tag{23.13}$$

The equation is then solved by iteration, with $\lambda_c^{(0)}$ equal to the approximated λ_c obtained form the last interpolation, starting with the linear interpolation given by (21.8). Convergence is usually fast because the error is inversely proportional to the first CPTF derivative, $\tau' = \partial\tau/d\lambda$, which is expected to be large near isolated critical points. Furthermore, should τ' be small, we can expect a good prediction from the linear interpolation formula.

The main advantages of the foregoing procedure is that it is self adaptive and easily programmable. We can increment the number of points used in the interpolation equation with little extra effort. We can keep the number of points under a given maximum by discarding the outermost points when we add a new one. Thus the procedure is flexible and opens up several implementation alternatives. Furthermore, the storage requirements are kept modest and we can expect good convergence to reasonable accuracy.

After a new approximation $\overline{\lambda}_c$ has been established, we have to compute the corresponding solution $\overline{\mathbf{u}}_c$. However, some care has to be exercised since the Jacobian matrix can be very ill-conditioned in the critical point neighborhood. There are several alternatives to circumventing this ill-conditioning problem, which merge with the general topic of *traversing* a critical point. This subject is taken up in the following subsections.

§23.4. Traversing Critical Points: Difficulties

In the remainder of this section we consider the problem of *traversing* critical points using incremental/corrective methods. Corrective solution methods, of which Newton-Raphson is the prototype, may run into difficulties at or near critical points. Essentially those are caused by the fact that the tangent stiffness matrix becomes singular at such points and consequently ill-conditioned in their neighborhood. This ill-conditioning can introduce *noise* which may render the solution procedure unstable.

Sometimes the problems associated with noise are easily circumvented, whereas in some cases the problems may become computationally overwhelming and may require either specialized techniques or intensive human intervention to proceed. Generally speaking, the degree of difficulty is influenced by the following factors:

1. Bifurcation points are more difficult to handle than limit points.
2. Isolated critical points (at which the rank deficiency is one) are easier to handle than multiple or compound critical points.
3. Dynamic critical points of flutter type are more difficult to handle than static ones. Ultimate in nastiness (sort of like the triple jump in ice skating) are problems of *fracture* or *localization* where an equililibrium path suddenly terminates, and dynamic methods are required for proceeding further.

In what follows we focus on the characteristics of Newton's method near an isolated critical point of static type. Even under those relatively benign conditions the topic is still an area of active research.

§23.5. Newton-Raphson near Critical Points

The behavior of Newton-Raphson methods at critical points have received much attention from the numerical analysis community.[7] Convergence rates have been established and accelerators have been proposed in a general framework. Under certain conditions, it can be established that the convergence of the solution component in the null space is linear while that in the range space converges superlinearly. This property can be exploited in the limit point case.

The purpose of a Newton-Raphson critical point accelerator, or NRCPA, is to recover quadratic rate of convergence. To achieve this goal on a general setting NRCPAs need information provided by second order (and ocassionally higher) rate equations. This information is very expensive to obtain because the rate $\dot{\mathbf{K}}$ of the tangent stiffness matrix, which is a "cubic array" discussed further in Chapter 24, makes its appearance. Because of such expense, NRCPAs based on higher order rate equations are rarely used in nonlinear structural mechanics.

The procedures described below represent more practical approaches to the traversal problem. They attempt to live with the first-order rate information, such as provided by **K** and **q**, which is normally available in finite element programs.

§23.5.1. Injecting Perturbations

By far the simplest technique to attempt critical point traversal is the equivalent of the "Hail Mary" football play. If at a certain step of an incremental/iterative process the equation solver returns a "singular stiffness" diagnostic, perturb the state **u** by a minute random amount and try again. This may be repeated up to a certain number of "downs" (usually 2 or 3).

This simplistic method sometimes works surprisingly well for crossing "nice" limit points in conjunction with the arclength increment control method. But at sharp limit points, bifurcation points or regions where the tangent stiffness has high rank deficiency this technique often fails.

§23.5.2. Penalty Spring Stabilization

As emphasized several times before, the essential difficulty lies in the ill-conditioning of **K** near critical points. To stabilize this matrix, Felippa[8] has proposed to combine the solution of *three* linear systems. The coefficient matrix is rendered nonsingular by adding a fictitious *penalty* spring stiffness s to the i^{th} equation:

$$(\mathbf{K}+s\mathbf{E}_i)\,\mathbf{d}^s_r=-\mathbf{r},\qquad (\mathbf{K}+s\mathbf{E}_i)\,\mathbf{d}^s_q=\mathbf{q},\qquad (\mathbf{K}+s\mathbf{E}_i)\,\mathbf{d}^s_e=s\mathbf{e}_i \tag{23.14}$$

[7] See, for example, the recent publications:

D. W. Decker, H. B. Keller and C. T. Kelly, Convergence Rates for Newton's Method at Singular Points, *Siam J. Numer. Anal.*, **20**, 296–314, 1983.

C. T. Kelly and R. Suresh, A New Acceleration Method for Newton's Method at Singular Points, *Siam J. Numer. Anal.*, **20**, 1001–1009, 1983.

G. Reddien, On Newton's Method for Singular Problems, *Siam J. Numer. Anal.*, **15**, 993–996, 1978.

W. Rheinboldt, Numerical Analysis of Continuation Methods for Nonlinear Structural Problems, *Computer & Structures*, **15**, 1–11, 1978.

[8] C. A. Felippa, Traversing critical points by penalty springs, Contrib. C2/1, Proc. NUMETA'87 Conf., M. Nijhoff Pubs., Dordrecht, 1987.

Here $\mathbf{e}_i$ is the elementary vector of order N all of whose entries are zero except the i^{th} entry which is one, and $\mathbf{E}_i = \mathbf{e}_i \mathbf{e}_i^T$. Some possible choices for the index i are discussed in that article (handed out in class). One advantage to this approach is that the symmetry and sparseness of **K** is not affected by this diagonal correction. The solution of the original Newton-Raphson system (17.13), reproduced here for convenience:

$$\begin{bmatrix} \mathbf{K} & -\mathbf{q} \\ \mathbf{a}^T & g \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \eta \end{bmatrix} = - \begin{bmatrix} \mathbf{r} \\ c \end{bmatrix}, \tag{23.15}$$

can be expressed as a linear combination of the solutions of the three auxiliary problems defined in (21.14):

$$\begin{aligned} \mathbf{d} &= \mathbf{d}_r^s + \sigma_q \mathbf{d}_q^s + \sigma_e \mathbf{d}_e^s, \\ \eta &= -\left(c + \mathbf{a}^T \mathbf{d}\right)/g, \end{aligned} \quad . \tag{23.16}$$

Here the coefficients σ_q and σ_e are obtained by requiring that **d** and η solve the original problem. The following 2 × 2 unsymmetric system results:

$$\begin{bmatrix} g + \mathbf{a}^T \mathbf{d}_q^s & \mathbf{a}^T \mathbf{d}_e^s \\ -\mathbf{e}_i^T \mathbf{d}_q^s & 1 - \mathbf{e}_i^T \mathbf{d}_e^s \end{bmatrix} \begin{bmatrix} \sigma_q \\ \sigma_e \end{bmatrix} = \begin{bmatrix} -c - \mathbf{a}^T \mathbf{d}_r^s \\ \mathbf{e}_i^T \mathbf{d}_r^s \end{bmatrix} \tag{23.17}$$

The computation of η from this equation breaks down if $g = 0$. In such a case, it is recommended that η be recovered from the equivalent equation:

$$\eta = \sigma_q + s \frac{\sigma_e d_i - d_i^2}{\mathbf{d}^T \mathbf{q}} \tag{23.18}$$

The value of s in equation is irrelevant as long as it is sufficiently large to stabilize **K**.

An advantage of this procedure is that it avoids the need for partitioning the stiffness matrix proposed by Rheinboldt (*loc. cit.*), which requires special treatment of the elements in the i^{th} row and column of **K**. Furthermore, none of the right-hand sides of the auxiliary systems in requires access to elements of **K**.

Disadvantages of this procedure are as follows:

1. It breaks down at bifurcation points, where the coefficient matrix of the 2 × 2 system becomes singular.
2. Some overhead is necessary to trace the rate of change of the components of the state vectors, which is used as a criterion in the selection of the index i.
3. The equation solver should be able to *undo* part of the reduction, from the current value of the row index to the selected value of i.

§23.5.3. Thurston's Equivalence Transformation

A different approach has been proposed by Thurston, Brogan and Stehlin. [9] Here an equivalence transformation is effected on the stiffness matrix to obtain a partitioning such that the singularity

[9] G. A. Thurston, F. A. Brogan and P. Stehlin, Postbuckling Analysis Using a General Purpose Code, *AIAA Journal*, **24**, 1013–1020, 1986.

is confined to a small block-submatrix. To achieve this transformation it is necessary to compute the eigenvectors corresponding to the smallest eigenvalues of **K**. These eigenvectors enter the first columns of the transformation matrix, which is then completed with a permutation matrix in such a way that renders the full matrix non-singular. The solution vector is then split accordingly and the components corresponding to the permutation part of the transformation matrix are condensed and solved in terms of the modal variables. Although this procedure works adequately for almost singular matrices, it fails for truly singular matrices where the modal variables remain undefined.

§23.6. Crivelli's Procedure

We present here an alternative procedure developed in Crivelli's thesis (cited in footnote 1) which does not involve partitioning but requires access to elements of **K**. Furthermore, it avoids the need to estimate penalty coefficients and the choice of the index i is straightforward. It also requires minor modifications to the solver routine. It can be implemented without even modifying the solver, by forcing the solver to return when it finds a diagonal term smaller than a given threshold, performing the required transformations outside the solver and then having the solver continue from the point of interruption. In addition, the proposed procedure furnishes an approximation to the null eigenvector and provides a measure of the distance to the critical point.

Define the permutation matrix $\mathbf{P}_i$:

$$\mathbf{P}_i = \begin{array}{r} \\ \\ \\ i\rightarrow \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & \dots & 0 & \overset{\overset{i}{\downarrow}}{0} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \tag{23.19}$$

Observe that

$$\mathbf{P}_i\mathbf{P}_i^T = \begin{bmatrix} \mathbf{I}_{i-1} & & \\ & 0 & \\ & & \mathbf{I}_{n-i} \end{bmatrix} = \mathbf{I}_n - \mathbf{e}_i\mathbf{e}_i^T, \tag{23.20}$$

where N is the dimension of $\mathbf{P}_i$ and $\mathbf{I}_k$ is the $k \times k$ identity matrix. Define

$$\hat{\mathbf{K}} = \mathbf{P}_i^T\mathbf{K}\mathbf{P} + \mathbf{e}_n\mathbf{e}_n^T, \qquad \hat{\mathbf{h}} = \mathbf{P}_i^T\mathbf{K}\mathbf{e}_i = \mathbf{P}_i^T\mathbf{k}_i, \tag{23.21}$$

and denote $\hat{\mathbf{z}}$ to be the solution of

$$\hat{\mathbf{K}}\hat{\mathbf{z}} = \hat{\mathbf{h}}, \qquad \bar{\mathbf{z}} = \mathbf{P}\hat{\mathbf{z}} - \mathbf{e}_i. \tag{23.22}$$

The vector $\bar{\mathbf{z}}$ so defined satisfies

$$\mathbf{K}\bar{\mathbf{z}} = \mathbf{k}_i^T\bar{\mathbf{z}}\mathbf{e}_i, \qquad \mathbf{k}_i = \mathbf{P}_i\hat{\mathbf{K}}\mathbf{P}_i^T\bar{\mathbf{z}} + K_{ii}\mathbf{e}_i. \tag{23.23}$$

We show below that if **K** is singular with rank deficiency of one, and the index i is chosen properly, then $\bar{\mathbf{z}}$ is the null eigenvector of **K**. If **K** is singular, there is a subset of the columns of **K** that are linearly dependent. Denote by $S = \{i_1, \ldots, i_k\}$ a set of indices such that $S \subset \{1, \ldots, n\}$; this set S is a subset of the column indices of $\mathbf{K}$. This set is chosen so that there is a linear combination of columns of $\mathbf{K}$ satisfying

$$\sum_{i_k \subset S} \beta_{i_k} \mathbf{k}_{i_k} = \mathbf{0} \tag{23.24}$$

where $\mathbf{k}_i$ is the i^{th} column of $\mathbf{K}$. Such a set exists because $\mathbf{K}$ is singular. Choose an index i from such that $\beta_i \neq 0$. Column $\mathbf{k}_i$ can be then expressed as a linear combination of the other columns of $\mathbf{K}$. Apply now the procedure with the index i chosen as described. It is immediately seen that $\mathbf{k}_i^T \overline{\mathbf{z}} = 0$; therefore $\overline{\mathbf{z}}$ is the null eigenvector of $\mathbf{K}$.

If $\mathbf{K}$ is *almost* singular, then $\overline{\mathbf{z}}$ is an approximation to the null eigenvector. It can be shown that if $\mathbf{K}$ is continuously differentiable and $\hat{\mathbf{K}}$ is nonsingular, then $\overline{\mathbf{z}}$ approaches $\mathbf{z}$ as $(\mathbf{u}, \lambda)$ approach $(\mathbf{u}_c, \lambda_c)$.

To solve the original problem, define the following auxiliary systems:

$$\hat{\mathbf{K}}\hat{\mathbf{d}}_r = \mathbf{P}_i^T \mathbf{r}, \quad \overline{\mathbf{d}}_r = \mathbf{P}_i \hat{\mathbf{d}}_r, \qquad \hat{\mathbf{K}}\hat{\mathbf{d}}_q = \mathbf{P}_i^T \mathbf{q}, \quad \overline{\mathbf{d}}_q = \mathbf{P}_i \hat{\mathbf{d}}_q \tag{23.25}$$

Observe that $\mathbf{d}_r^T \mathbf{e}_i = \mathbf{d}_q^T \mathbf{e}_i = 0$. Define

$$\mathbf{d} = \overline{\mathbf{d}}_r + \sigma_q \overline{\mathbf{d}}_q + \sigma_z \overline{\mathbf{z}} \tag{23.26}$$

To obtain the coefficients σ_q and σ_e we use a similar procedure as before. We have

$$\begin{aligned} \mathbf{K}\mathbf{d}_r &= \mathbf{P}_i \mathbf{P}_i^T \mathbf{K} \mathbf{P}_i \hat{\mathbf{d}}_r + \mathbf{k}_i^T \mathbf{d}_r \mathbf{e}_i = \mathbf{P}_i \hat{\mathbf{K}} \hat{\mathbf{d}}_r + \mathbf{k}_i^T \mathbf{d}_r \mathbf{e}_i \\ &= -(\mathbf{I} - \mathbf{e}_i \mathbf{e}_i^T)\mathbf{r} + \mathbf{k}_i^T \mathbf{d}_r \mathbf{e}_i = -\mathbf{r} + (\mathbf{k}_i^T \mathbf{d}_r + r_i)\mathbf{e}_i \end{aligned} \tag{23.27}$$

and

$$\mathbf{K}\mathbf{d}_q = \mathbf{q} + (\mathbf{k}_i^T \mathbf{d}_q - q_i)\mathbf{e}_i \tag{23.28}$$

Also observe that

$$\begin{aligned} \mathbf{k}_i^T \mathbf{d}_r &= \mathbf{d}_r^T (\mathbf{P}_i \hat{\mathbf{K}} \hat{\mathbf{z}} + K_{ii} \mathbf{e}_i) = \hat{\mathbf{d}}_r^T \hat{\mathbf{K}} \hat{\mathbf{z}} \\ &= -\mathbf{r}^T \mathbf{P}_i \hat{\mathbf{z}} = -\mathbf{r}^T (\overline{\mathbf{z}} + \mathbf{e}_i) = -\mathbf{r}^T \overline{\mathbf{z}} - r_i \end{aligned} \tag{23.29}$$

Thus

$$\mathbf{k}_i^T \mathbf{d}_r + r_i = -\mathbf{r}^T \overline{\mathbf{z}} \tag{23.30}$$

Similarly

$$\mathbf{k}_i^T \mathbf{d}_q - q_i = -\mathbf{q}^T \mathbf{z} \tag{23.31}$$

From previous equations it is immediately seen that

$$\mathbf{K}\mathbf{d} - \mathbf{q}\eta = -\mathbf{r} + \left[\sigma_q + \frac{c + \mathbf{a}^T \mathbf{d}}{g} \right] \mathbf{q} + \left[\mathbf{k}_i^T \overline{\mathbf{z}} \sigma_z + \mathbf{q}^T \overline{\mathbf{z}} \sigma_q - \mathbf{r}^T \overline{\mathbf{z}} \right] \mathbf{e}_i \tag{23.32}$$

To recover the original system, we require the terms inside squared brackets to vanish, which leads to the auxiliary system

$$\begin{bmatrix} g + \mathbf{a}^T \mathbf{d}_q & \mathbf{a}^T \overline{\mathbf{z}} \\ \mathbf{q}^T \overline{\mathbf{z}} & \mathbf{k}_i^T \overline{\mathbf{z}} \end{bmatrix} \begin{bmatrix} \sigma_q \\ \sigma_z \end{bmatrix} = \begin{bmatrix} -c - \mathbf{a}^T \mathbf{d}_r \\ \mathbf{r}^T \overline{\mathbf{z}} \end{bmatrix} \tag{23.33}$$

Finally, we have to recover η. Instead of using We can obtain η directly by requiring

$$\eta = \sigma_q \tag{23.34}$$

This approach has some advantages over the one pertaining to the penalty spring method. First, if $\sigma_z = 0$ it reduces to the one used by classical continuation. Second, it enforces prior equations. Closer inspection of tells us that $\sigma_e = d_i$ if exact arithmetic is used. Thus, the second term vanishes. When finite precision arithmetic is used, rounding errors may prevent this term from vanishing. If in addition, a sufficiently large value for the penalty spring s is chosen, could produce some artificially large value of η with the consequence that the corrector step may move the solution away from convergence.

Several remarks may be made in regard to the preceding derivation. At a limit point, neither $\mathbf{q}^T\mathbf{z}$ nor $\mathbf{a}^T\mathbf{z}$ vanish and the equations are well behaved even though $\mathbf{k}_i^T\mathbf{z}$ is zero. Thus this procedure can satisfactory traverse limit points. At a bifurcation point the last row of equation is identically null, thus the value of σ_z is not defined. This is consistent with the definition of bifurcation point, since at these points more than one solution is possible and we cannot continue tracing any branch without introducing additional information. Thus when a bifurcation point is detected, this procedure breaks down and we have to resort to a branch switching algorithm as described in the following sections.

§23.7. Predictor Stabilization

A similar procedure can be employed to stabilize the predictor, a subject that has not received much attention in the literature. If $\mathbf{K}$ is almost singular, the predictor may be poor. Using $\hat{\mathbf{d}}_q$ defined in equation (11) we define the predictor step $\Delta\mathbf{u}^{(0)}$ as

$$\Delta\mathbf{u}^{(0)} = \hat{\mathbf{d}}_q \Delta\lambda^{(0)} + \sigma_z\bar{\mathbf{z}} \tag{23.35}$$

Thus

$$\mathbf{K}\Delta\mathbf{u}^{(0)} - \mathbf{q}\Delta\lambda^{(0)} = \left[\Delta\lambda^{(0)}(\mathbf{k}_i^T\mathbf{q} - q_i) + \sigma_z\mathbf{k}_i^T\bar{\mathbf{z}}\right]\mathbf{e}_i \tag{23.36}$$

Requiring the term in brackets to vanish and using equations (17) the load increment is obtained as:

$$\Delta\lambda^{(0)} = \frac{\mathbf{k}_i^T\bar{\mathbf{z}}}{\mathbf{q}^T\bar{\mathbf{z}}}\sigma_z \tag{23.37}$$

Assuming we use the global hyperelliptic constraint we can obtain σ_q as

$$\sigma_z = \frac{\ell}{\pm\sqrt{\dfrac{a^2}{v^2}\left(\left(\dfrac{\mathbf{k}_i^T\bar{\mathbf{z}}}{\mathbf{q}^T\bar{\mathbf{z}}}\right)^2\mathbf{d}_q^T\mathbf{S}\mathbf{d}_q + \bar{\mathbf{z}}^T\mathbf{S}\bar{\mathbf{z}}\right) + \left(\dfrac{\mathbf{k}_i^T\bar{\mathbf{z}}}{\mathbf{q}^T\bar{\mathbf{z}}}\right)^2 b^2}} \tag{23.38}$$

since by construction $\mathbf{d}_q^T\bar{\mathbf{z}} = 0$.

Observe that at a limit point, $\mathbf{k}_i^T\mathbf{z} = 0$, $\mathbf{q}^T\mathbf{z} \neq 0$, equation (38) gives $\sigma_z = \ell$, $\Delta\lambda^{(0)} = 0$, allowing the solution to move away from the limit point. At a bifurcation point $\mathbf{q}^T\mathbf{z} = 0$, and this equation does not give any solution for σ_z, as expected, since any solution with a nonzero value of σ_z will lie on the emanating branch. In this case, we simply set $\sigma_z = 0$ and use prior relations compute $\Delta\lambda^{(0)}$.

§23.8. Critical-Point Test Functions

In this section we investigate several candidates for CPTF. A CPTF is a function that allows monitoring the continuation procedure for occurrence of critical points. By definition, critical points should be the only zeros of a CPTF. As mentioned before, critical points are characterized by the tangent stiffness **K** being singular. Thus, test functions essentially monitor **K** to obtain an estimate of its proximity to a critical point.

A highly accurate estimator of *how* singular is **K** is its smallest eigenvalue. Since **K** is real and symmetric, its eigenvalues are all real; in addition, if **K** varies continuously, its eigenvalues also vary continuously and singularity is simply detected by checking for a null eigenvalue. Thus, the smallest eigenvalue of **K** makes a very attractive choice for a CPTF. However, an eigenvalue analysis at each increment is prohibitively expensive.

Another possible test function is the determinant of **K**. As noticed by Abbott[10] the order of magnitude of the determinant may be computationally inconvenient, especially for large stiff systems, when the solution is computed far away from bifurcation. Scaling **K** by 10 changes the determinant by 10^N! This difficulty can be overcome by computing the determinant as a pair of numbers, a characteristic normalized between -1 and 1 and a mantissa. This decomposition is available in several direct solvers. A drawback of this approach occurs, however, if the null eigenvalue has an even multiplicity, since in this case the determinant will not change sign and the tester will fail to detect a critical point.

Two additional choices are considered in the following. Using previous equations the following relation results:

$$\frac{\bar{\mathbf{z}}^T\mathbf{K}\bar{\mathbf{z}}}{\bar{\mathbf{z}}^T\bar{\mathbf{z}}} = \frac{\mathbf{k}_i^T\bar{\mathbf{z}}}{\bar{\mathbf{z}}^T\bar{\mathbf{z}}} \tag{23.39}$$

If we regard the left-hand side of this equation as a Rayleigh quotient, and taking into account that $\bar{\mathbf{z}}$ is a close approximation to the smallest eigenvector of **K** when the solution is sufficiently close to the limit point, we may use the right-hand side of equation to monitor the smallest eigenvalue of **K**. Thus, $\mathbf{k}_i\bar{\mathbf{z}}$ is a candidate for a CPTF. A similar test function is proposed by Seydel (*loc.cit.*). This approach is expensive if it is to be carried out at every increment, because the decomposition is only computed near critical points. However, if by some other means we have detected the proximity of a critical point, this computation may be obtained as a by-product of the stabilization procedure described in the previous Chapter. In such case, it becomes an attractive CPTF. We therefore look for a cheaper alternative that can provide a good estimate of the singularity of **K**.

Since we use full Newton we generally have a decomposition of **K** of the form:

$$\mathbf{K} = \mathbf{L}^T\mathbf{D}\mathbf{L} \tag{23.40}$$

where **L** is an upper triangular matrix and **D** is diagonal. It is well known that

$$\tau = \min_i \mathbf{D}(i) \tag{23.41}$$

[10] J. P. Abbott, An Efficient Algorithm for the Determination of Certain Bifurcation Points, *J. Comput. Appl. Math.*, **4**, 19–27, 1978.

also monitors the smallest eigenvalue of $\mathbf{K}$. Moreover, $\tau = 0$ when $\mathbf{K}$ is singular and $\tau < 0$ if $\mathbf{K}$ is indefinite. Thus, τ defined is readily available without any overhead and is the test function adopted here. Once it has been determined that there is a critical point lying between two solutions we may switch to the test function.

§23.9. Branch switching

In this Chapter we consider the computational treatment of bifurcation points. Such points are characterized by the fact that the augmented stiffness becomes singular there. The vector $\langle \mathbf{z}^T \quad 0 \rangle$ is a left singular vector of the augmented stiffness, as can be seen by premultiplying the augmented stiffness matrix by the vector $\langle \mathbf{z}^T \quad 0 \rangle$ and taking into account that $\mathbf{q}$ is orthogonal to $\mathbf{z}$. However, this relation shows that the problem is consistent and as such, a solution exits; more precisely, an infinite set of solutions exits with any two of them differing by a vector lying in the null space of $\mathbf{K}$.

The coefficient σ_z cannot be obtained because the second row of this equation vanishes identically. Thus, we cannot use here the same procedure used to traverse limit points. As previously mentioned, at a bifurcation point more than one branch coexist. If σ_z is kept equal to zero, we may continue along the current branch, while if another value of σ_z is chosen appropriately, we may switch to a different branch. It is seen that the problem is indeterminate and we need an additional condition to specify on which branch of the solution the continuation will proceed. While continuing on the known branch can be achieved fairly simply, switching to a crossing branch is not so obvious and requires a more elaborate procedure.

We will assume that $\mathbf{K}$ has a rank deficiency of one and the bifurcation point is *isolated* or *simple*. Under these conditions, there are only two intersecting branches at the bifurcation point. The process of branch switching requires two steps. First we have to find some point sufficiently close to the crossing branch. Second, we have to stay on that branch; this requires to avoid any iteration sending us back to the known branch. This Chapter will be concerned with the first step, calculating one solution point on the emanating branch.

Suppose we have detected a critical point, and that we have classified it as a bifurcation point. The scenario is as follows: we have some solutions $\mathbf{u}(\lambda)$ such that $\mathbf{r}(\mathbf{u}, \lambda) = \mathbf{0}$. If the bifurcation point has been located by using the indirect method, we may assume that we have one solution, say $(\overline{\mathbf{u}}, \overline{\lambda})$, that approximates the bifurcation point $(\mathbf{u}_c, \lambda_c)$, and in addition we have a good estimate of the buckling mode vector $\mathbf{z}$. Denote a solution on the crossing branch by $(\mathbf{w}, \lambda)$, i.e. $\mathbf{r}(\mathbf{w}, \lambda) = \mathbf{0}$. Our main goal in this section is to find one such solution. This first solution is then used as a starting point to trace the entire branch.

In general, any method for switching branches consists of two parts, first an approximation to a point on the crossing branch is *guessed* by means of a predictor; then an iteration converging to that branch must be established. In the following subsections we will describe different types of predictors.

§23.9.1. Tangent Predictors

Possibly the most accurate predictor is the one based on the tangent to the crossing branch. However, this procedure requires a rather accurate calculation of the bifurcation point $(\mathbf{u}_c, \lambda_c)$ and of the second derivatives of the residual vector, which is given in terms of the rate matrices $\dot{\mathbf{K}}$ and $\dot{\mathbf{q}}$. Since we

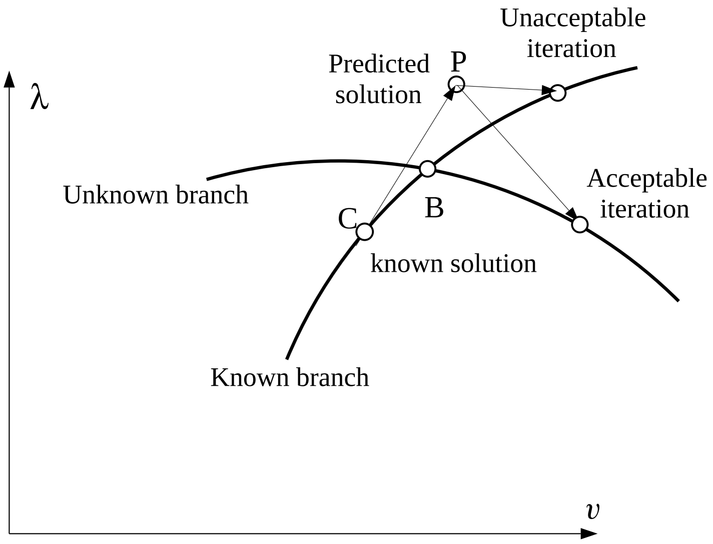

Figure 23.1. Schematic representation of a branch switching procedure.

have assumed that the bifurcation point is simple, there can be at most *two* intersecting branches. Since $\mathbf{q}$ is in the range of $\mathbf{K}_c$, the system $\mathbf{Kw} = \mathbf{q}$ is consistent at a bifurcation point. Hence, there is a unique solution $\mathbf{w}$ lying in the range of $\mathbf{K}(\mathbf{u}_c, \lambda_c)$ that is, satisfying the relation $\mathbf{w}^T\mathbf{z} = 0$. Now the state variation rate $\dot{\mathbf{u}}$ at a bifurcation point can be decomposed into a *homogeneous solution* component $\sigma\mathbf{z}$ in the buckling mode direction, and a *particular solution* component, orthogonal to $\mathbf{z}$, which is furnished by $\mathbf{w}$:

$$\dot{\mathbf{u}} = (\mathbf{w} + \sigma\mathbf{z})\,\dot{\lambda} \tag{23.42}$$

The next step is to find σ. It was pointed out that the first-order rate equations do not provided enough information for computing σ. To obtain the required information we must resort to the second-order system. Premultiplying equation by $\mathbf{z}^T$ and taking into account the bifurcation condition 3 in Definition 2 we get the scalar equation

$$\mathbf{z}^T\dot{\mathbf{K}}\dot{\mathbf{u}} - \mathbf{z}^T\dot{\mathbf{q}}\dot{\lambda} = 0 \tag{23.43}$$

The rate matrices $\dot{\mathbf{K}}$ and $\dot{\mathbf{q}}$ may be written as linear combinations of $\dot{\mathbf{u}}$ and $\dot{\lambda}$ as:

$$\dot{\mathbf{K}} = \mathbf{L}\dot{\mathbf{u}} + \mathbf{N}\dot{\lambda}, \qquad \dot{\mathbf{q}} = -\left(\mathbf{N}\dot{\mathbf{u}} + \mathbf{j}\dot{\lambda}\right) \tag{23.44}$$

Then

$$\mathbf{z}^T\left(\mathbf{L}\dot{\mathbf{u}} + \mathbf{N}\dot{\lambda}\right)\dot{\mathbf{u}} + \mathbf{z}^T\left(\mathbf{N}\dot{\mathbf{u}} + \mathbf{j}\dot{\lambda}\right)\dot{\lambda} = 0 \tag{23.45}$$

The replacement of $\dot{\mathbf{u}}$ by its decomposition gives

$$\mathbf{z}^T\left(\mathbf{L}\,(\mathbf{w} + \sigma\mathbf{z})\,\dot{\lambda} + \mathbf{N}\dot{\lambda}\right)(\mathbf{w} + \sigma\mathbf{z})\,\dot{\lambda} + \mathbf{z}^T\left(\mathbf{N}\,(\mathbf{w} + \sigma\mathbf{z})\,\dot{\lambda} + \mathbf{j}\dot{\lambda}\right)\dot{\lambda} = 0 \tag{23.46}$$

Insertion of the relations

$$a = \mathbf{z}^T\mathbf{L}\mathbf{z}\mathbf{z}, \qquad b = \mathbf{z}^T\,(\mathbf{L}\mathbf{z}\mathbf{w} + \mathbf{N}\mathbf{z})\,, \qquad c = \mathbf{z}^T\,(\mathbf{L}\mathbf{w}\mathbf{w} + 2\mathbf{N}\mathbf{w} + \mathbf{j}) \tag{23.47}$$

yields the quadratic equation

$$a\sigma^2 + 2b\sigma + c = 0 \tag{23.48}$$

If at least one coefficient is nonzero, this quadratic equation provides two roots: σ_1 and σ_2 —root $\sigma = \infty$ is acceptable. Since by hypothesis a branch reaches the bifurcation point, one of the roots must be real. Consequently, the other root must also be real. The condition that at least one coefficient be nonzero serves as a criterion for a simple bifurcation point because it guarantees a transversal intersection of two branches. It is worth noting that $a = 0$ characterizes a *pitchfork* or symmetric bifurcation while $a \neq 0$ characterizes a *transcritical* or asymmetric bifurcation.

Although this method is systematic and reliable, it suffers from the disadvantage of requiring second derivatives of the residual in addition to requiring an accurate approximation of the bifurcation point. Both procedures are rather expensive, particularly the former, since it is not always possible to obtain an analytical expression for the rate of the stiffness matrix in a reasonably simple way. This is obviously the case when the formulation used corresponds to elaborate mathematical models. These problems motivated the search for other alternatives. In general, these alternatives are based on *buckling mode injection*.

§23.9.2. Buckling Mode Injection

The purpose of these predictors is to obtain a sufficiently close approximation to the emanating branches without resorting to the computation of the exact tangents to such branches. The underlying idea is to obtain an increment vector **d** representing the difference between one solution in the known branch and a solution on the unknown branch for a given value of the load parameter λ,

$$\mathbf{d} = \mathbf{w} - \mathbf{u}, \qquad \lambda = \overline{\lambda} \tag{23.49}$$

Observe that the two tangents at the bifurcation point obtained by replacing the two roots span a plane. In a sufficiently small neighborhood of the bifurcation point $(\mathbf{u}_c, \lambda_c)$ both branches lie on that plane. If $\overline{\lambda}$ is sufficiently close to λ_c it would be possible to approximate the vector **d** by a vector almost parallel to this tangent plane. If **d** could be obtained exactly it would be possible to switch branches in a straightforward manner. However, in most practical problems only an approximation to **d** is possible. Such approximation is generally split into two steps

1. *Find a direction pointing to the other branch*
2. *Find a step along this direction so that the distance between the two solutions is minimized in a given norm.*

The problem is then reduced to finding **d** and δ such that

$$|\mathbf{u} + \delta\mathbf{d} - \mathbf{w}| = 0, \qquad \mathbf{r}(\mathbf{u}, \overline{\lambda}) = \mathbf{r}(\mathbf{w}, \overline{\lambda}) = \mathbf{0} \tag{23.50}$$

The first condition can be viewed in the more general context of a constraint relation between the solutions on both branches, thus generalizing constraints.

The first subproblem can be solved by taking **d** proportional to **z** since **z** lies on the plane defined by the two tangents. This can easily be seen just by taking the difference between the two tangents and observing that this difference is a multiple of **z**. This procedure can be viewed as an *injection* of the buckling mode into the known solution.

The solution to the second subproblem is more difficult. It does not seem to be possible to find a δ that satisfies the previous equation for a given value of $\overline{\lambda}$ without resorting to information provided by second or higher-order rate equations. We cope with this problem by shifting it to the corrector, which will adjust λ instead of δ in such a way that a constraint condition between the solutions on both branches is satisfied. The problem is so restated because the corrector is better equipped to deal with a problem posed in terms of changes in the state vector and control parameters, because an incremental equation for δ is not available. Thus, at the predictor level, we will be content with an estimation of δ. Some values dictated by experience are given in Seydel (*loc.cit.*).

Note that two predictors are in fact available:

$$\overline{\mathbf{w}}_{1,2} = \mathbf{u} \pm \delta \mathbf{z} \tag{23.51}$$

For perfect structures undergoing symmetric —pitchfork— bifurcation, the sign choice is irrelevant, but for asymmetric —transcritical— bifurcation, one branch will be *energy preferred*. This difficulty can be overcome by calculating solutions on both half-branchs emanating from the bifurcation point and choosing the one with least energy.

The success of the whole procedure strongly depends on an appropriate corrector. Correctors should be carefully designed in order to display *selective properties*. This is meant as the capability of the corrector to avoid converging towards the known branch. The design of such correctors is dealt with in the next section.

§23.10. Correctors

One alternative is to *enrich* the displacement increment with the buckling mode. Felippa (*loc.cit.*) modifies equations so that

$$\mathbf{d} = \mathbf{d}_r^s + \sigma_q \mathbf{d}_q^s + \sigma_e \mathbf{d}_e^s \pm \sigma_z \mathbf{z} \tag{23.52}$$

To provide sufficient equations a length constraint in injected, $\mathbf{z}^T \mathbf{d} = \delta$. Again, the unknown coefficients are obtained by requiring that $\mathbf{d}$ verifies the incremental equation which leads to the following 3 × 3 unsymmetric auxiliary system:

$$\begin{bmatrix} g + a_q & a_e & a_z \\ -d_{qi} & 1 - d_{ei} & l \\ s_q & s_e & 1 \end{bmatrix} \begin{bmatrix} \sigma_q \\ \sigma_e \\ \pm\sigma_z \end{bmatrix} = \begin{bmatrix} -c - a_r \\ d_{ri} \\ \delta - s_r \end{bmatrix} \tag{23.53}$$

in which

$$a_z = \mathbf{z}^T \mathbf{a}, \quad d_z = \mathbf{z}^T \mathbf{d}, \quad s_r = \mathbf{z}^T \mathbf{d}_r^s, \quad s_q = \mathbf{z}^T \mathbf{d}_q^s, \quad s_e = \mathbf{z} \mathbf{d}_e^s \tag{23.54}$$

The selective properties of this corrector are not clearly established.

To obtain a corrector with enhanced selective properties, Skeie and Felippa[11] propose using a constraint that has greater affinity for the emanating branch than for the known branch. For this purpose, they look at the limit behavior of the local hyperelliptic constraint. They show that this constraint reduces to a narrow cylinder around the known branch when $a/\ell \gg b/\ell$. To obtain a robust corrector it is necessary to inject information from both branches into the constraint equation.

[11] G. Skeie and C.A. Felippa, Detecting and Traversing Bifurcation Points in Nonlinear Structural Analysis, Report CU-CSSC-89-23, Center for Space Structures and Controls, University of Colorado, Boulder, 1989.

This can easily be achieved if the constraint is expressed in terms of the difference between the solutions on two different branches, as shown below.

In the correction step, can be generalized to

$$c(\mathbf{w}-\mathbf{u},\lambda-\overline{\lambda})=0 \tag{23.55}$$

Increments to both $\mathbf{w}$ and $\mathbf{u}$ and to λ are computed so that this constraint is enforced. What we are trying to avoid is an iteration converging towards the known branch. Experience shows that this frequently happens when $\overline{\lambda}=\lambda_c$ is kept fixed while trying to obtain a state vector $\mathbf{u}$ satisfying the residual equation. For example, consider the case of symmetric —pitchfork— bifurcation. In this case, the crossing branch exists for values of λ either greater or smaller than λ_c. Since we do not know beforehand on which side of the bifurcation point the crossing branch lies, there is a 50% chance that no emanating branch exist for the chosen value of λ. Besides, it is possible that the emanating branch could not be parametrized by λ close to the bifurcation point and a local parametrization may be required.

Once again we can expand the solution space. A point in this expanded space is defined by $(\mathbf{u},\mathbf{w},\lambda)$ and a parametrization for the solution in this new space is sought. In the case of prior equations the parameter was the arc-length ℓ; here δ plays the same role. The only difference is that we replace $\Delta\mathbf{u}$, which represents an increment of the solution on the known branch, by $\mathbf{w}-\mathbf{u}$ which measures the difference between two solutions lying on different branches. For example, displacement control results in

$$w_k-u_k=\delta \tag{23.56}$$

In summary, we look for two state corrections $\mathbf{d}_w$ and $\mathbf{d}_v$ and a load correction η that simultaneously satisfy

$$\mathbf{r}(\mathbf{u}^k+\mathbf{d}_v^k,\lambda^k+\eta^k)=\mathbf{r}(\mathbf{w}^k+\mathbf{d}_w^k,\lambda^k+\eta^k)=\mathbf{0} \tag{23.57}$$

and the constraint condition

$$c(\Delta\mathbf{u}^k+\mathbf{d}_v^k,\,\Delta\mathbf{w}^k+\mathbf{d}_w^k;\,\Delta\lambda^k+\eta^k)=0 \tag{23.58}$$

These corrections are obtained from

$$\begin{bmatrix}\mathbf{K}_v & \mathbf{0} & -\mathbf{q}\\ \mathbf{0} & \mathbf{K}_w & -\mathbf{q}\\ \mathbf{a}^T & -\mathbf{a}^T & g\end{bmatrix}\begin{bmatrix}\mathbf{d}_v\\ \mathbf{d}_w\\ \eta\end{bmatrix}=-\begin{bmatrix}\mathbf{r}_v\\ \mathbf{r}_w\\ c\end{bmatrix} \tag{23.59}$$

where $\mathbf{K}_v=\mathbf{K}(\mathbf{u}^k,\lambda^k)$ and $\mathbf{K}_w=\mathbf{K}(\mathbf{w}^k,\lambda^k)$.

Again it is possible to solve the previous equation while preserving the symmetry and sparseness of $\mathbf{K}$. If $\mathbf{K}$ is nonsingular we may write

$$\mathbf{d}_{ru}=-\mathbf{K}_v^{-1}\mathbf{r}_v,\quad \mathbf{d}_{rw}=-\mathbf{K}_w^{-1}\mathbf{r}_w,\quad \mathbf{d}_{qu}=\mathbf{K}_v^{-1}\mathbf{q},\quad \mathbf{d}_{qw}=\mathbf{K}_w^{-1}\mathbf{q} \tag{23.60}$$

Thus, the two incremental displacements and the load increment may obtained as:

$$\begin{aligned}\mathbf{d}_v&=\mathbf{d}_{ru}+\eta\mathbf{d}_{qu}\\ \mathbf{d}_w&=\mathbf{d}_{rw}+\eta\mathbf{d}_{qw}\\ \eta&=-\frac{c+\mathbf{a}^T(\mathbf{d}_{ru}-\mathbf{d}_{rw})}{g+\mathbf{a}^T(\mathbf{d}_{qu}-\mathbf{d}_{qw})}\end{aligned} \tag{23.61}$$

By using the procedure described in previous equations we require much less storage than when using the original system, because $\mathbf{K}_v$ and $\mathbf{K}_w$ can overwrite each other. The extra storage requirement is for four N-dimensional auxiliary vectors. However, it is still demanding in terms of computer time, because two linear systems of equations with different coefficient matrices must be solved at each iteration.

As an alternative, we may keep the solution on the known branch fixed and search for another equilibrium solution that also verifies the constraint equation. That is, solve

$$\begin{bmatrix} \mathbf{r}(\mathbf{w},\lambda) \\ c(\mathbf{u};\,\mathbf{d}_w,\lambda) \end{bmatrix} = \mathbf{0} \tag{23.62}$$

This equation resembles the original continuation equation; the main difference being a slight modification of the constraint condition. However, there is no guarantee that a solution of this equation will lie on the emanating branch; observe that $\mathbf{u}$ and $\mathbf{w}$ correspond to two different values of the load parameter λ and it is possible that two solutions on the same branch satisfy the constraint equation.

In the examples presented in Crivelli's thesis the first approach is used, since we are willing to pay the additional price to guarantee the success of the branch switching. Furthermore this cost is modest compared to that of the entire continuation process.

A remark is in order here. The stiffness matrices $\mathbf{K}_v$ and $\mathbf{K}_w$ may become ill-conditioned and a procedure similar to that leading to equations may be developed. However, note that we are not looking for a highly accurate solution on the known branch, since we are using it just to *repel* an iteration from the crossing branch back to the known branch. Such a process is not required to provide an highly accurate solution on the crossing branch either, because that solution is just used to obtain a starting point on the new branch. We can thus apply the transformation to $\mathbf{K}$ to avoid numerical instability, but regarding $\overline{\mathbf{d}}_{ru}$ and $\overline{\mathbf{d}}_{rw}$ as sufficiently good approximations to $\mathbf{d}_{ru}$ and $\mathbf{d}_{rw}$, respectively. Once a point on the new branch is obtained, the solution can be further improved by switching back to the continuation procedure used for regular points.

§23.11. Treating Bifurcation by Perturbation

The last approach to be mentioned here is treating bifurcation by perturbation. The idea is to perturb the residual equation $\mathbf{r}(\mathbf{u},\lambda)$ in such a way that the underlying regularity intrinsic to a bifurcation point is destroyed. This is accomplished by introducing physical or numerical *imperfections*. In the mathematical literature, this approach is referred to as *unfolding*. The perturbed system displays limit point behavior rather than bifurcation point behavior. The key difficulty with this approach is to find a perturbation parameter ϵ and an imperfection distribution that is critical to the underlying regularity. For simple structures, this parameter ϵ may be guessed from physical considerations. Once an appropriate value is found, the branch can be traced by the standard continuation procedure and the problem is reduced to limit point traversal.

Homework Exercises for Chapter 23

Detecting and Traversing Critical Points

EXERCISE 23.1

[C:25] For the steep 2-bar arch with $S = 2$ and arbitrary H and subject to a vertical load $-\lambda q$, find the locus of all critical points directly by setting up and solving the $\mathbf{r} = 0$ system augmented by det($\mathbf{K}$) $= 0$ as CPTF.

Hint: you may try the effect of

```
ClearAll[Em,A0,S,vX,vY,H,lambda];
Em=1; A0=1; S=2;
Do [
  rX = (4*A0*Em*vX*(S^2 + 2*vX^2 + 4*H*vY + 2*vY^2))/(4*H^2 + S^2)^(3/2);
  rY = (8*A0*Em*(H + vY)*(vX^2 + 2*H*vY + vY^2))/(4*H^2 + S^2)^(3/2)-lambda;
  K={{(4*A0*Em*(S^2 + 6*vX^2 + 4*H*vY + 2*vY^2))/(4*H^2 + S^2)^(3/2),
      (16*A0*Em*vX*(H + vY))/(4*H^2 + S^2)^(3/2)},
     {(16*A0*Em*vX*(H + vY))/(4*H^2 + S^2)^(3/2),
      (8*A0*Em*(2*H^2 + vX^2 + 6*H*vY + 3*vY^2))/(4*H^2 + S^2)^(3/2)}};
 sol=NSolve[{rX==0,rY==0,Det[K]==0},{lambda,vX,vY}];
 Print["solution for H=",H,"  is ",({lambda,vX,vY}/.sol)//InputForm],
   {H,1/2,3,1/2} ];
```

EXERCISE 23.2 [A+C:25] The residual equation

$$r(\theta,\lambda) = \theta - \lambda \sin\theta = 0, \tag{E23.1}$$

where $v \equiv \theta$ is the only degree of freedom, provides what is perhaps the simplest example of bifurcation. This has two solution paths, $\theta = 0$ (the fundamental path) and $\lambda = \theta/\sin\theta$ (the post-buckling path), which intersect at B ($\lambda = 1, \theta = 0$.

Solve this equation using the purely incremental arclength control method programmed in Chapter 18 and devise a procedure by which the program, starting from $\lambda = \theta = 0$ and moving up the fundamental path, detects B and (without cheating) follows one of the post-buckling paths. Probably the simplest one to implement is perturbation.

EXERCISE 23.3 [C:25] Solve the steep arch $H = 2S$ under a vertical load using the program of Exercise 20.3 and observe whether the program senses the bifurcation point or just marches along the fundamental path. Experiment with perturbing the problem by inserting a small horizontal load to see whether you can coax it into the secondary path.