Computational techniques for the non-linear analysis of structures

Reijo Kouhia

April 2004, corrected version May 2009

Abstract

These lecture notes gives an introduction to computational strategies for non-linear structural analysis. Some techniques, based on Newton's iteration to solve the global equilibrium equations are explaned. Continuation or path-following methods to solve the parametrized non-linear equations are presented. Special emphasis is given to the determination of critical points along equilibrium paths. Asymptotic techniques in structural stability analysis are also briefly discussed. As supplementary material, some videly used algorithms in solving algebraic eigenvalue problems and linear equation systems are presented.

Contents

Chapter 1

Solution methods for non-linear equations

1.1 Some historical notes

Numerical solution of non-linear equations has a long history. A common iterative procedure bears the name of Newton or Newton-Raphson [27], but there exist many names which could be credited either before Newton's time or later [198]. The general idea of solving an equation by improving an estimate of a solution by adding a correction term had been in use in many cultures millenia prior to this time [198]. Certain ancient Greek and Babylonian methods for extracting roots have this form, as do some methods of Arabic algebraist from at least the time of al-Khayy $\bar{a}$m (1048-1131) [198].

French algebraist Francois Viète published in 1600 in Paris a work concerning the numerical solution of non-linear algebraic equations: *De numerosa potestatum*. Viète restricted his attention to monic polynomial equations and can, in some sense, be viewed as a forerunner in using the finite-difference scheme of the Newton-Raphson method.

Newton's tract *De analysi per aequationes numero terminorum infinitas* [1] (On analysis by equations unlimited in the number of their term), probably dating from the mid 1669, is the first recorded discussion by Newton of what can be recognized as an instance of the Newton-Raphson method. It seems to be that the tract is a reworking of old material of Viète and Nicolaus Mercator's *Logarithmotechnia*, published in London in September 1668 [84], [198]. No calculus is used in the presentation and references to fluxional derivatives first appear later in that tract, suggesting that Newton regarded his method as a purely algebraic procedure [198]. The first published use by Newton of his method applied to a nonpolynomial equation appears in the second edition of his treatise *Philosophiae Naturalis Principia Mathematica*[2]

[1] The tract remained semi-secret a long time until William Jones printed it, with other early mathematical essays by Newton, in *Analysis per quantitatum series, fluxiones, ac differentias* in 1711 [84].

[2] First edition of the *Principia* was published in London in 1687.

Joseph Raphson (1648-1712?) published in 1690 a tract *Analysis aequationum universalis* in which he presented a method for solving polynomial equations. Newton's and Raphson's methods were long regarded as distinct, until in 1798 J.-L. Lagrange observed that the difference is only due to the presentation and not due to the underlying method and credited Raphson's method as being simpler. It is also interesting to note, that the formulation of the method using the now familiar calculus notation is also due to Lagrange.

Thomas Simpson (1710-1761) seems to be the first to give the method a general formulation, in terms of fluxional calculus, applicable to nonpolynomial equations. Simpson published his work in London in 1740 and describes "A new method for the solution of equations in numbers" without making reference to the work of any predecessors. In his work the technique is also described for a system of equations, however, restricted to the case of two equations [198].

As expressed by Ypma [198], the Newton-Raphson-Simpson method would be a designation which represents the facts of history in a more appropriate way rather than calling the method simply by Newton's name. This major lack of recognition is probably due to Lagrange and especially due to Fourier, who did not mention either Raphson or Simpson in his influental book *Analyse des Équations Determinées* published in 1831 [198].

The modern literature on the solution of non-linear algebraic equations is vast. The bibliography of the classical monograph by James Ortega and Werner Rheinboldt [130] published in 1970 is 35 pages long and contains approximately 850-900 references. Path following, continuation, embedding or homotopy methods, as they are also called, are constantly used for a wide range of scientific applications to solve *parametrized* nonlinear equation systems. One reason for their success is their versatility and robustness. Recent books dealing with continuation are written e.g. by Allgower and Georg [6], Keller [102], Rheinboldt [148], Seydel [169].

Even though the idea of continuation dates back to the last century, [3] the earliest application of techniques for the numerical solution of parametrized equations appears to have been made by E. Lahaye in 1934 for a single equation, using Newton's method to move along the solution curve [130]. Later Lahaye also considered systems of equations (1948) [130].

In structural mechanics, interest towards continuation rose after the invention of the finite element method and the advent of digital computers during the 1960's. The continuation was first realized by incremental loading without any equilibrium iterations, i.e. Euler-forward approach by Turner *et al.* in 1960 [186] and Argyris in 1965 [9]. Later Newton's iteration was adapted by Oden [129] and Mallet and Marcal [120]. Early work involving limit point instabilities was due Sharifi and Popov [171] and Sabir and Lock [157].

Finally, it is noted that the solution of non-linear equations have a close relation to unconstrained optimization, see refs. [10], [52], [127]. The first application of Newton's method to the problem of multivariate unconstrained optimization seems to be due to

[3] For a historical summary of continuation see Ficken [67].

Simpson in his *A New Treatise of Fluxions*, published in 1737 [198].

1.2 Introduction

Discretization of the non-linear equations of static equilibrium result in a system of the form

$$\boldsymbol{f}(\boldsymbol{q}) \equiv \boldsymbol{r}(\boldsymbol{q}) - \boldsymbol{p}(\boldsymbol{q}) = \boldsymbol{0}. \tag{1.1}$$

The unbalanced or residual force is denoted by $\boldsymbol{f}$, $\boldsymbol{q}$ is the state variable vector which, in the displacement based FE-formulation, is a nodal point displacement vector. External loads and internal resistance forces are denoted by $\boldsymbol{p}$ and $\boldsymbol{r}$, respectively.

In dynamics, the relation of the equations of motion is transformed using d'Alembert's principle to a problem of finding dynamic equilibrium

$$\boldsymbol{f}(\boldsymbol{q},t) \equiv \boldsymbol{r}(\boldsymbol{q}) - \boldsymbol{p}(\boldsymbol{q},t) + \boldsymbol{M}\,\ddot{\boldsymbol{q}} = \boldsymbol{0}. \tag{1.2}$$

This ordinary differential equation system can be solved with either explicit or implicit time integration schemes. Explicit integration algorithms provide the most straightforward solution method, but since they are almost allways conditionally stable the limitation of maximum stepsize puts severe restrictions on the practical use of these schemes. They are mainly used in analyses where the high frequency content of the structure contributes significantly to the response, as is the case in transients induced by shocks, blast or any type of loading with a broad frequency range. Implicit schemes benefit the fact that the step length is not so severely limited by stability considerations and they are efficiently used in transient problems with frequency content in the lower range, in which the behaviour of the structure is mainly inertial [77].

A multistep (k-step) method to integrate the time dependency of (1.2) can be expressed in the form

$$\sum_{i=0}^{k} a_i \boldsymbol{q}_{n-i} = h \sum_{i=0}^{k} b_i \dot{\boldsymbol{q}}_{n-i}, \qquad \sum_{i=0}^{k} c_i \boldsymbol{q}_{n-i} = h^2 \sum_{i=0}^{k} g_i \ddot{\boldsymbol{q}}_{n-i}, \tag{1.3}$$

where a_i, b_i, c_i and g_i are coefficients and h is the latest step-length. Solving $\ddot{\boldsymbol{q}}_n$ from these equations and substituting it into (1.2) gives an algebraic equation in $\boldsymbol{q}_n$

$$\boldsymbol{f}(\boldsymbol{q}_n,t_n) = \boldsymbol{r}(\boldsymbol{q}_n) - \boldsymbol{p}(\boldsymbol{q}_n,t_n) + (g_0 h^2)^{-1}\boldsymbol{M}\Big(\sum_{i=0}^{k} c_i \boldsymbol{q}_{n-i} - h^2 \sum_{i=1}^{k} g_i \ddot{\boldsymbol{q}}_{n-i}\Big) = \boldsymbol{0}. \tag{1.4}$$

Denoting the effective load vector by

$$\boldsymbol{p}_{\text{eff}}(\boldsymbol{q}_n,t_n) = \boldsymbol{p}(\boldsymbol{q}_n,t_n) - (g_0 h^2)^{-1}\boldsymbol{M}\Big(\sum_{i=0}^{k} c_i \boldsymbol{q}_{n-i} - h^2 \sum_{i=1}^{k} g_i \ddot{\boldsymbol{q}}_{n-i}\Big) \tag{1.5}$$

the equation of dynamic equilibrium

$$\boldsymbol{f}(\boldsymbol{q}_n, t_n) \equiv \boldsymbol{r}(\boldsymbol{q}_n) - \boldsymbol{p}_{\text{eff}}(\boldsymbol{q}_n, t_n) = \boldsymbol{0} \tag{1.6}$$

is of the same form as the equation of static equilibrium (1.1).

Solution of the non-linear set of equations (1.1) is usually done in an stepwise manner, by incrementin the external load $\boldsymbol{p}$ from un unloaded state to a spesific value. Considering a certain increment n, the application of Taylor's series expansion on the vector of unbalanced forces at state $\boldsymbol{q}_n^i$ results in

$$\boldsymbol{f}(\boldsymbol{q}_n^{i+1}) \approx \boldsymbol{f}(\boldsymbol{q}_n^i) + \boldsymbol{f}'(\boldsymbol{q}_n^i)\delta\boldsymbol{q}_n^i = \boldsymbol{0} \tag{1.7}$$

where quadratic and higher order terms are neglected and $\boldsymbol{f}'$ denotes the Jacobian matrix

$$\boldsymbol{f}' = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} - \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} = \boldsymbol{K}_r - \boldsymbol{K}_p = \boldsymbol{K} \tag{1.8}$$

which becomes the tangent stiffness matrix at an equilibrium point. The Newton-Raphson iteration formula is then

$$\begin{aligned} \boldsymbol{q}_n^{i+1} &= \boldsymbol{q}_n^i - [\boldsymbol{f}'(\boldsymbol{q}_n^i)]^{-1}\boldsymbol{f}(\boldsymbol{q}_n^i) \\ &= \boldsymbol{q}_n^i + \delta\boldsymbol{q}_n^i = \boldsymbol{q}_{n-1} + \Delta\boldsymbol{q}_n^i + \delta\boldsymbol{q}_n^i = \boldsymbol{q}_{n-1} + \Delta\boldsymbol{q}_n^{i+1}, \end{aligned} \tag{1.9}$$

where the superscript denotes the iteration count and the subscript the step number which usually will be omitted when reference is made to quantities of the same step.

The load stiffness matrix $\boldsymbol{K}_p$ is symmetric provided the load is conservative. The lack of symmetry in constitutive equations, e.g. in non-associative plasticity models, can also produce an unsymmetric stiffness matrix. In addition, some co-rotational formulations lead to unsymmetric Jacobian matrices when evaluated at a non-equilibrium point even if the loading is conservative and the material model possesses symmetry properties.

1.3 Local convergence

Local convergence of the iteration scheme (1.9) can be proved if the following standard assumptions hold [130], [52]:

1. $\boldsymbol{f}$ is continuously differentiable in an open convex domain $D \in \mathbb{R}^N$
2. there exists $\boldsymbol{q}^*$ and $r > 0$ such that $\mathcal{B}(\boldsymbol{q}^*, r) \in D$ and $\boldsymbol{f}(\boldsymbol{q}^*) = \boldsymbol{0}$
3. the Jacobian matrix $\boldsymbol{f}'$ is invertible at $\boldsymbol{q}^*$ and $\| [\boldsymbol{f}'(\boldsymbol{q}^*)]^{-1} \| \leq \beta$
4. the Jacobian matrix is Lipschitz continuous in $\mathcal{B}(\boldsymbol{q}^*, r)$, i.e.

$$\|\boldsymbol{f}'(\boldsymbol{q}) - \boldsymbol{f}'(\boldsymbol{y})\| \leq \gamma\|\boldsymbol{q} - \boldsymbol{y}\| \qquad \forall \boldsymbol{q}, \boldsymbol{y} \in \mathcal{B}(\boldsymbol{q}^*, r). \tag{1.10}$$

Then there exist $\epsilon > 0$ such that for all $\boldsymbol{q}^0 \in \mathcal{B}(\boldsymbol{q}^0, \epsilon)$ the sequence $\boldsymbol{q}^1, \boldsymbol{q}^2, \ldots$ generated by the Newton's iteration (1.9) converges to $\boldsymbol{q}^*$ and obeys

$$\|\boldsymbol{q}^{k+1} - \boldsymbol{q}^*\| \leq \beta\gamma \|\boldsymbol{q}^k - \boldsymbol{q}^*\|^2. \tag{1.11}$$

Practically, this asymptotic result can be interpreted as doubling of the number of significant digits in $\boldsymbol{q}^k$ as an approximation to $\boldsymbol{q}^*$.

The Newton attraction theorem also expresses the existence of a domain of attraction, which implies that if the Newton iterates ever land in this domain then they will remain there and eventually converge to $\boldsymbol{q}^*$; a result which insures some measure of stability for the iteration process [51].

A well known convergence result for Newton's method is due to Kantorovich. It differs from the theorem presented mainly in that it makes no assumption about the existence of the solution $\boldsymbol{q}^*$. It assumes only that the Jacobian is nonsingular at the initial point $\boldsymbol{q}^0$, $\boldsymbol{f}'$ is Lipschitz continuous in a region containing $\boldsymbol{q}^0$, and the first step of Newton's method is sufficiently small. Under these assumptions the Kantorovich theorem shows that there exists a unique solution in the region. Formally stated; assuming that

1. $\boldsymbol{f}$ is continuously differentiable in a ball $\mathcal{B}(\boldsymbol{q}^0, r), r > 0$,
2. the Jacobian matrix $\boldsymbol{f}'$ is nonsingular at $\boldsymbol{q}^0$ and $\| [\boldsymbol{f}'(\boldsymbol{q}^0)]^{-1} \| \leq \beta$
3. the Jacobian matrix is Lipschitz continuous in $\mathcal{B}(\boldsymbol{q}^0, r)$, see eq. (1.10), with Lipschitz constant γ,
4. the first Newton step is sufficently small: $\|[\boldsymbol{f}'(\boldsymbol{q}^0)]^{-1}\boldsymbol{f}(\boldsymbol{q}^0)\| \leq \eta$

then if $h_0 = \beta\gamma\eta < \frac{1}{2}$ the Newton sequence (1.9) converges to a unique solution in $\mathcal{B}(\boldsymbol{q}^0, r_1)$, where $r_1 = \min(r, r_0)$

$$r_0 \equiv \frac{1 - \sqrt{1 - 2h_0}}{\beta\gamma}. \tag{1.12}$$

and

$$\|\boldsymbol{q}^k - \boldsymbol{q}^*\| \leq (2h_0)^{2^k} \frac{\eta}{h_0}, \qquad k = 0, 1, 2, \ldots \tag{1.13}$$

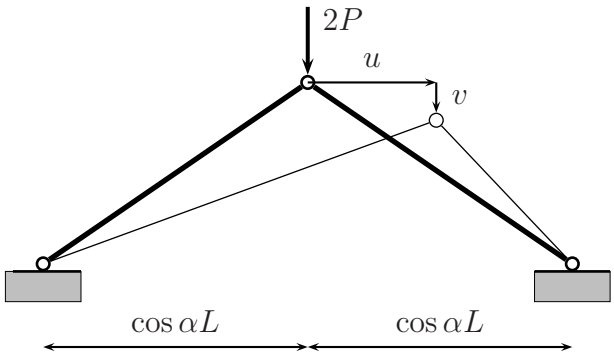
It is also worth noticing, that the Newton's method is self correcting, which means that $\boldsymbol{q}^{k+1}$ only depends upon $\boldsymbol{f}$ and $\boldsymbol{q}^k$ implying that the bad effects from previous iterations are not carried along, an advantage which is not shared by quasi-Newton methods [51].

However, there are some serious drawbacks in the Newton's method. In large nonlinear structural finite element analysis the need to evaluate the Jacobian of $\boldsymbol{f}$ and its possile factorization for each iteration can be extremely costly. Another disadvantage is that the domain of attraction for a particular problem can be very small thus requiring a very good initial approximation to $\boldsymbol{q}^*$ in order to get convergence of the iteration process. In structural analyses this is usually not a problem, since choosing sufficiently small time or load

steps, the previously known equilibrium configuration provides a good initial estimate to the next step. Nevertheless, if a good initial approximation to $\boldsymbol{q}^*$ is not available, special techniques have been developed to circumvent this problem [143].

Several modifications to the basic Newton's method have been introduced in order to avoid the formation and factorization of the Jacobian. The simplest possible choice is to hold the Jacobian fixed for a certain period, for instance, during one load or time increment. Especially in engineering literature this scheme is ambiguously named "the modified Newton-Raphson iteration". This technique is useful when the Jacobian is changing slowly, however, it is very difficult to decide how long the Jacobian should be held fixed. Evidently, the rate of convergence is decreased, but the overall efficiency in some particular problems may increase.

Example 1.3.1. *A Mises truss will be considered. Length and the initial angle of the bars at the initial state are L and α, respectively. and the axial stiffness equals to EA. The bars are assumed to be absolutely rigid in bending. Determine the equilibrium equations and solve with some value of the load and investigate the convergence of the Newton's process.*



Length of the bars in the deformed configuration is

$$\begin{aligned} L_{1,\mathrm{def}} &= \sqrt{(L\cos\alpha + u)^2 + (L\sin\alpha - v)^2} \\ &= L\sqrt{1 + 2q_1\cos\alpha + q_1^2 - 2q_2\sin\alpha + q_2^2} \qquad (1.14) \\ L_{2,\mathrm{def}} &= \sqrt{(L\cos\alpha - u)^2 + (L\sin\alpha - v)^2} \\ &= L\sqrt{1 - 2q_1\cos\alpha + q_1^2 - 2q_2\sin\alpha + q_2^2} \qquad (1.15) \end{aligned}$$

where $q_1 = u/L$ and $q_2 = v/L$. Using the Green-Lagrange definition for the strain

$$\epsilon_i = \frac{1}{2}\frac{L_{i,\mathrm{def}}^2 - L^2}{L^2} \qquad i = 1, 2 \qquad (1.16)$$

gives

$$\epsilon_1 = q_1\cos\alpha + \tfrac{1}{2}q_1^2 - q_2\sin\alpha + \tfrac{1}{2}q_2^2, \qquad (1.17)$$

$$\epsilon_2 = -q_1\cos\alpha + \tfrac{1}{2}q_1^2 - q_2\sin\alpha + \tfrac{1}{2}q_2^2. \qquad (1.18)$$

The principle of virtual work is

$$\int_0^L N_1 \delta\epsilon_1 \mathrm{d}x + \int_0^L N_2 \delta\epsilon_2 \mathrm{d}x = 2P\delta v \tag{1.19}$$

where the axial force is defined as $N_i = EA\epsilon_i$ and the virtual strains have the expressions:

$$\delta\epsilon_1 = (\cos\alpha + q_1)\delta q_1 + (-\sin\alpha + q_2)\delta q_2, \tag{1.20}$$
$$\delta\epsilon_2 = (-\cos\alpha + q_1)\delta q_1 + (-\sin\alpha + q_2)\delta q_2, \tag{1.21}$$

where $\delta q_1 = \delta u/L$ and $\delta q_2 = \delta v/L$. The expression of the virtual work is thus

$$[EA(cq_1 + \tfrac{1}{2}q_1^2 - sq_2 + \tfrac{1}{2}q_2)(c + q_1) + (cq_1 - \tfrac{1}{2}q_1^2 + sq_2 - \tfrac{1}{2}q_2)(c - q_1)]\delta u + [EA(q_1^2 - 2sq_2 + q_2^2)(q_2 - s) - 2P]\delta v = 0, \tag{1.22}$$

where $s = \sin\alpha$ and $c = \cos\alpha$. Since the variations δu and δv are arbitrary, the the equilibrium equations must satisfy

$$\boldsymbol{f}(\boldsymbol{q}) = \left\{ \begin{array}{ll} f_1 = 2c^2 q_1 + q_1^3 - 2sq_1q_2 + q_1q_2^2 & = 0 \\ f_2 = -sq_1^2 + q_1^2 q_2 + 2s^2 q_2 - 3sq_2^2 + q_2^3 - 2\lambda & = 0 \end{array} \right. , \tag{1.23}$$

where $\lambda = P/EA$.

Elements of the Jacobian matrix are

$$\frac{\partial f_1}{\partial q_1} = 2c^2 + 3q_1^2 - 2sq_2 + q_2^2, \tag{1.24}$$
$$\frac{\partial f_1}{\partial q_2} = 2q_1(q_2 - s) = \frac{\partial f_2}{\partial q_1} \tag{1.25}$$
$$\frac{\partial f_2}{\partial q_2} = 2s^2 + q_1^2 - 6sq_2 + 3q_2^2 \tag{1.26}$$

Solution for the given load is symmetric (prior bifurcation), thus $q_1 = 0$, and the Jacobian matrix is

$$\boldsymbol{K} = \left[\begin{array}{cc} 2c^2 - 2sq_2 + q_2^2 & 0 \\ 0 & 2s^2 - 6sq_2 + 3q_2^2 \end{array} \right]. \tag{1.27}$$

It is clearly seen that the Jacobian is positive definite if

$$2c^2 - 2sq_2 + q_2^2 > 0 \tag{1.28}$$
$$2s^2 - 6sq_2 + 3q_2^2 > 0. \tag{1.29}$$

The first inequality is always satisfied if $s = \tan\alpha < \sqrt{2}$, i.e. for initial angles $\alpha < 54.74°$. The second inequality is satisfied for dispacaments $q_2 < (1 - \sqrt{3}/3)\sin\alpha$.

Solution of the symmetric deformation is equivalent in solving a single non-linear equation

$$f(q_2, \lambda) = 2s^2 q_2 - 3sq_2^2 + q_2^3 - 2\lambda = 0. \tag{1.30}$$

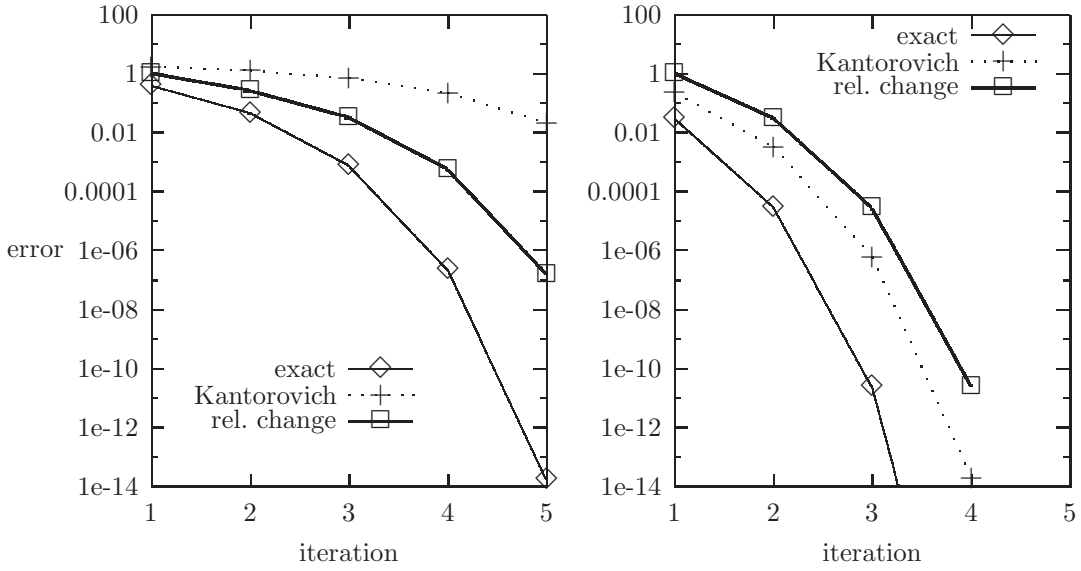


Figure 1.1: Mises truss; convergence of Newton's method, large step on the lhs and small step on the rhs.

For this system, estimation of the Lipschitz constant γ can be done from the expression of the second derivative

$$\gamma < \max |f''(q_2)| \qquad q_2 \in (0, r_0). \tag{1.31}$$

Using the value $\alpha = 30^\circ$ gives the following values

$$\gamma = 3, \quad \beta = |f'(0)^{-1}| = 2 \tag{1.32}$$

In fig.1.1 convergence of the Newton's method on the first step is shown using two different step sizes. In real computations convergence is checked with some estimate of the error, here in the figure a relative norm

$$\text{estimated error} = e_{\text{rel}} = \frac{|\delta q^k|}{|\delta q^0|} = \frac{|\delta q^k|}{\eta} \tag{1.33}$$

In fig.1.1 the estimated error (1.33) is compated to the relative true error $|q^k - q_*|/\eta$ and the relative Kantorovich estimate, i.e. sequence (1.13) divided by η. The load is chosen such that the exact solution is either 0.1 or 0.01.

It is clearly seen, that the Kantorovich estimate is far too concervative for the large step case, where $\eta = 0.072, h_0 = 0.432$ thus giving the radius of the convergence domain $r_0 = 0.1052$. In the small step case the corresponding values are: $\eta = 0.009702, h_0 = 0.0582, r_0 = 0.010002$.

1.4 Convergence near singularity

Another, purely numerical, problem is also present near singular points. It is well known that the convergence rate of Newton's method downgrades from quadratic to linear when the solution of the equation system corresponds to a singular point [50]. Many techniques to speed up the convergence have been presented in the mathematical literature, which can in theory give a superlinear rate of convergence.

1.5 Quasi-Newton Iterations

1.5.1 Basic properties

A class of algorithms called quasi-Newton (or variable metric, variance, secant, update or modification methods) have been developed in order to speed up the convergence of the modified Newton method, but which could still be more efficient than the true Newton-Raphson scheme. The very essence of these methods lies in an update formula of the Jacobian matrix, performed in such a way which avoids the reforming and factorization of the global matrix.

The problem is now how to develop a good approximation to the Jacobian at the state $\boldsymbol{q}_i$ based on information at the iterations i and $i-1$.[4]

Available data are: the Jacobian at iteration step $i-1$ (or an earlier approximation of it), the unbalanced forces and the state variables at iterations $i, i-1$. It seems natural to require that the approximation $\bar{\boldsymbol{H}}_i$ to $\boldsymbol{f}'(\boldsymbol{q}_i)$ satisfies the *secant relationship*

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{q}_i) &= \boldsymbol{f}(\boldsymbol{q}_{i-1}) + \bar{\boldsymbol{H}}_i(\boldsymbol{q}_i - \boldsymbol{q}_{i-1}), \\ \Rightarrow \quad \bar{\boldsymbol{H}}_i \delta\boldsymbol{q}_{i-1} &= \delta\boldsymbol{f}_{i-1}, \end{aligned} \tag{1.34}$$

where

$$\delta\boldsymbol{q}_{i-1} = \boldsymbol{q}_i - \boldsymbol{q}_{i-1} \quad \delta\boldsymbol{f}_{i-1} = \boldsymbol{f}_i - \boldsymbol{f}_{i-1}.$$

This equation is central to the development of the quasi-Newton methods and it is therefore called the quasi-Newton, or secant equation.

In the case of a scalar equation, the secant relationship (1.34) completely determines $\bar{\boldsymbol{H}}_i$, but for a system of equations, additional requirements have to be imposed. It is reasonable to require that the updated matrix $\bar{\boldsymbol{H}}_i$ is close to the previous matrix $\boldsymbol{H}_{i-1}$. This nearness is measured by matrix norms, and the requirement can be given as follows: find $\bar{\boldsymbol{H}}_i$ such that

$$\min \left\{ \|\bar{\boldsymbol{H}}_i - \boldsymbol{H}_{i-1}\| : \bar{\boldsymbol{H}}_i \delta\boldsymbol{q}_{i-1} = \delta\boldsymbol{f}_{i-1} \right\}. \tag{1.35}$$

Usually, in connection to quasi-Newton updates, the Frobenius norm or its weighted form are used

$$\|\boldsymbol{H}\| = \|\boldsymbol{H}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{H}^T\boldsymbol{H})}, \qquad \|\boldsymbol{H}\|_{W,F} = \|\boldsymbol{W}\boldsymbol{H}\boldsymbol{W}\|_F,$$

[4] The position for the symbol i showing the iteration count is now placed at the lower right corner of the quantity and the incremental step counter is not shown.

in which $\boldsymbol{W}$ is a positive definite symmetric matrix. Note that the Frobenius norm does not satisfy the submultiplicative property which is usually satisfied by matrix norms.

It is also desirable that the updated matrix should inherit some properties which are characteristic to the system. In structural finite element applications such properties usually are symmetry and positive definiteness of the stiffness matrix. So, the update $\bar{\boldsymbol{H}}_i$ should also satisfy

$$\begin{aligned} \boldsymbol{H}_{i-1} = \boldsymbol{H}_{i-1}^T \quad &\longrightarrow \quad \bar{\boldsymbol{H}}_i = \bar{\boldsymbol{H}}_i^T \\ \boldsymbol{x}^T \boldsymbol{H}_{i-1} \boldsymbol{x} > 0 \quad &\longrightarrow \quad \boldsymbol{x}^T \bar{\boldsymbol{H}}_i \boldsymbol{x} > 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0}. \end{aligned}$$

However, it should be remembered that the new iterative change $\delta \boldsymbol{q}_i$ has to be easily and cost effectively computed, otherwise the benefit of this kind of update is lost since the price which is paid for omitting the full Newton step is the degradation of the convergence rate.

The quasi-Newton techniques are closely related to the conjugate-Newton methods, see Refs. [26], [92], [136].

1.5.2 Rank one updates

Derivation

A single rank update to the stiffness matrix is a correction of the form

$$\bar{\boldsymbol{H}} = \boldsymbol{H} + \alpha \hat{\boldsymbol{y}} \hat{\boldsymbol{z}}^T, \tag{1.36}$$

where the unit vectors $\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$ and the scalar α are to be determined. Substituting this expression into the quasi-Newton equation (1.34) gives

$$\boldsymbol{H} \delta \boldsymbol{q} - \delta \boldsymbol{f} = -\alpha \hat{\boldsymbol{y}} \hat{\boldsymbol{z}}^T \delta \boldsymbol{q},$$

where the superscripts, indicating the iteration count, are omitted. Denoting the Euclidian vector norm by $\| \cdot \|_2$, it is easily seen that by choosing

$$\boldsymbol{y} = \delta \boldsymbol{f} - \boldsymbol{H} \delta \boldsymbol{q}, \quad \hat{\boldsymbol{y}} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2}, \quad \text{and} \quad \alpha = \frac{\|\boldsymbol{y}\|_2}{\hat{\boldsymbol{z}}^T \delta \boldsymbol{q}}$$

the secant relationship is fulfilled for all vectors $\hat{\boldsymbol{z}}$ which are not orthogonal to $\delta \boldsymbol{q}$. Thus, the single rank update is expressed as

$$\bar{\boldsymbol{H}} = \boldsymbol{H} + \frac{1}{\hat{\boldsymbol{z}}^T \delta \boldsymbol{q}} (\delta \boldsymbol{f} - \boldsymbol{H} \delta \boldsymbol{q}) \hat{\boldsymbol{z}}^T, \quad \forall \hat{\boldsymbol{z}}, \quad \hat{\boldsymbol{z}}^T \delta \boldsymbol{q} \neq 0, \quad \|\hat{\boldsymbol{z}}\|_2 = 1. \tag{1.37}$$

The vector $\hat{\boldsymbol{z}}$ can now be determined from the closeness requirement (1.35)

$$\begin{aligned} \min \|\bar{\boldsymbol{H}} - \boldsymbol{H}\|_F = \min \|\alpha \hat{\boldsymbol{y}} \hat{\boldsymbol{z}}^T\|_F &= \min \left[\operatorname{tr}(\alpha^2 \hat{\boldsymbol{z}} \hat{\boldsymbol{y}}^T \hat{\boldsymbol{y}} \hat{\boldsymbol{z}}^T) \right]^{\frac{1}{2}} \\ &= \|\boldsymbol{y}\|_2 \min \sqrt{\frac{\hat{\boldsymbol{z}}^T \hat{\boldsymbol{z}}}{\hat{\boldsymbol{z}}^T \delta \boldsymbol{q}}} = \min \frac{1}{\sqrt{\hat{\boldsymbol{z}}^T \delta \boldsymbol{q}}}. \end{aligned} \tag{1.38}$$

It is clear that the minimum is obtained when the vectors $\hat{\boldsymbol{z}}$ and $\delta\boldsymbol{q}$ are parallel, i.e. by choosing $\hat{\boldsymbol{z}} = \delta\boldsymbol{q}/\|\delta\boldsymbol{q}\|$, and the resulting update formula is

$$\bar{\boldsymbol{H}} = \boldsymbol{H} + \frac{(\delta\boldsymbol{f} - \boldsymbol{H}\delta\boldsymbol{q})\delta\boldsymbol{q}^T}{\delta\boldsymbol{q}^T\delta\boldsymbol{q}}. \tag{1.39}$$

Broyden [33] derived this approximation basing the consideration on somewhat different reasoning. It was supposed that $\bar{\boldsymbol{H}}$ and $\boldsymbol{H}$ operate identically on a vector belonging to the orthogonal complement of $\delta\boldsymbol{q}$, i.e.

$$\bar{\boldsymbol{H}}\boldsymbol{w} = (\boldsymbol{H} + \boldsymbol{y}\boldsymbol{z}^T)\boldsymbol{w} = \boldsymbol{H}\boldsymbol{w}, \quad \text{if} \quad \boldsymbol{w}^T\delta\boldsymbol{q} = 0. \tag{1.40}$$

It yields immediately $\boldsymbol{z} = \delta\boldsymbol{q}$ and substitution into the secant equation (1.34) gives for $\boldsymbol{y}$ the same expression as earlier.

Broyden's update formula does not have the property of hereditary symmetry and positive definiteness, but its simplicity provides an easy introduction to the quasi-Newton methods. However, it is interesting to note, that a symmetric rank one update is obtained from (1.37) by choosing $\boldsymbol{z} = \boldsymbol{y} = \delta\boldsymbol{f} - \boldsymbol{H}\delta\boldsymbol{q}$. Obviously in this case the closeness property (1.35) is not satisfied.

A greater variety of suitable symmetric update formulas can be derived if the correction is made by a matrix of rank two. These methods are examined in the following sections.

Implementation

Expression (1.39) for the Broyden's update formula is not suitable for practical computation. Direct use of (1.39) would destroy the specific sparsity pattern of the Jacobian and needs the factorization of the updated matrix, which therefore would be even more costly than application of Newton's method. The following derivation follows closely the one given by Kelley [104].

It is easy to see that the formula (1.39) can be expressed as

$$\bar{\boldsymbol{H}} = \boldsymbol{H} + \frac{\boldsymbol{f}_i\delta\boldsymbol{q}^T}{\delta\boldsymbol{q}^T\delta\boldsymbol{q}}. \tag{1.41}$$

Applying the Sherman-Morrison-Woodbury formula to the Broyden's update (1.41) [5] gives the update formula for the inverse matrix

$$(\boldsymbol{H} + \boldsymbol{u}\boldsymbol{v}^T)^{-1} = \left(\boldsymbol{I} - \frac{(\boldsymbol{H}^{-1}\boldsymbol{u})\boldsymbol{v}^T}{1 + \boldsymbol{v}^T\boldsymbol{H}^{-1}\boldsymbol{u}}\right)\boldsymbol{H}^{-1} \tag{1.42}$$

[5] The Sherman-Morrison-Woodbury formula gives a convenient expression for the inverse of $(\boldsymbol{S} + \boldsymbol{U}\boldsymbol{V}^T)$ where $\boldsymbol{S}$ is a nonsingular $n \times n$ matrix and $\boldsymbol{U}$, $\boldsymbol{V}$ are both $n \times k$ matrices: $(\boldsymbol{S} + \boldsymbol{U}\boldsymbol{V}^T)^{-1} = \boldsymbol{S}^{-1} - \boldsymbol{S}^{-1}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{V}^T\boldsymbol{S}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^T\boldsymbol{S}^{-1}$. Also $\boldsymbol{I} + \boldsymbol{V}^T\boldsymbol{S}^{-1}\boldsymbol{U}$ has to be nonsingular [79].

For a sequence of Broyden updates, it can be written

$$\boldsymbol{H}_i = \boldsymbol{H}_{i-1} + \boldsymbol{u}_{i-1}\boldsymbol{v}_{i-1}^T \tag{1.43}$$

where

$$\boldsymbol{u}_{i-1} = \boldsymbol{f}_i/\|\delta\boldsymbol{q}_{i-1}\| \qquad \text{and} \qquad \boldsymbol{v}_{i-1} = \delta\boldsymbol{q}_{i-1}/\|\delta\boldsymbol{q}_{i-1}\|. \tag{1.44}$$

Defining

$$\boldsymbol{w}_{i-1} = \frac{\boldsymbol{H}_{i-1}^{-1}\boldsymbol{u}_{i-1}}{1 + \boldsymbol{v}_{i-1}^T\boldsymbol{H}_{i-1}^{-1}\boldsymbol{u}_{i-1}} \tag{1.45}$$

then

$$\begin{aligned}\boldsymbol{H}_i^{-1} &= (\boldsymbol{I} - \boldsymbol{w}_{i-1}\boldsymbol{v}_{i-1}^T)(\boldsymbol{I} - \boldsymbol{w}_{i-2}\boldsymbol{v}_{i-2}^T)\cdots(\boldsymbol{I} - \boldsymbol{w}_0\boldsymbol{v}_0^T)\boldsymbol{H}_0^{-1} \\ &= \left[\prod_{j=0}^{i-1}(\boldsymbol{I} - \boldsymbol{w}_j\boldsymbol{v}_j^T)\right]\boldsymbol{H}_0^{-1},\end{aligned} \tag{1.46}$$

and the iterative step $\delta\boldsymbol{q}_i$ can be computed with $\boldsymbol{H}_0^{-1}$ and the $2i$ vectors $w_j, v_j, j = 0, \ldots i-1$ as

$$\delta\boldsymbol{q}_i = -\left[\prod_{j=0}^{i-1}(\boldsymbol{I} - \boldsymbol{w}_j\boldsymbol{v}_j^T)\right]\boldsymbol{H}_0^{-1}\boldsymbol{f}_i. \tag{1.47}$$

It can be shown that there is no need to store the sequence w_i [53]. To show that, let's first show that the computation of $\boldsymbol{w}_{i-1}$ and $\delta\boldsymbol{q}_i$ can be combined:

$$\delta\boldsymbol{q}_i = -\boldsymbol{H}_i^{-1}\boldsymbol{f}_i = -(\boldsymbol{I} - \boldsymbol{w}_{i-1}\boldsymbol{v}_{i-1}^T)\boldsymbol{H}_{i-1}^{-1}\boldsymbol{f}_i = -(\boldsymbol{I} - \boldsymbol{w}_{i-1}\boldsymbol{v}_{i-1}^T)\boldsymbol{z}, \tag{1.48}$$

where the auxiliary vector $\boldsymbol{z}$ is

$$\boldsymbol{z} = \boldsymbol{H}_{i-1}^{-1}\boldsymbol{f}_i = \left[\prod_{j=0}^{i-2}(\boldsymbol{I} - \boldsymbol{w}_j\boldsymbol{v}_j^T)\right]\boldsymbol{H}_0^{-1}\boldsymbol{f}_i. \tag{1.49}$$

Using the definition (1.45), gives

$$\boldsymbol{w}_{i-1} = \frac{\boldsymbol{H}_{i-1}^{-1}\boldsymbol{u}_{i-1}}{1 + \boldsymbol{v}_{i-1}^T\boldsymbol{H}_{i-1}^{-1}\boldsymbol{u}_{i-1}} = \frac{\boldsymbol{z}}{\|\delta\boldsymbol{q}_{i-1}\|(1 + \boldsymbol{v}_{i-1}^T\boldsymbol{z}/\|\delta\boldsymbol{q}_{i-1}\|)} = \alpha^{-1}\boldsymbol{z}, \tag{1.50}$$

where $\alpha = \|\delta\boldsymbol{q}_{i-1}\| + \boldsymbol{v}_{i-1}^T\boldsymbol{z}$. Hence

$$\begin{aligned}\delta\boldsymbol{q}_i &= -(\boldsymbol{I} - \boldsymbol{w}_{i-1}\boldsymbol{v}_{i-1}^T)\boldsymbol{z} = -\boldsymbol{z}(1 - \alpha^{-1}\boldsymbol{v}_{i-1}^T\boldsymbol{z}) \\ &= -\boldsymbol{z}(1 - \alpha^{-1}(\alpha - \|\delta\boldsymbol{q}_{i-1}\|)) = -\alpha^{-1}\|\delta\boldsymbol{q}_{i-1}\|\boldsymbol{z} = -\|\delta\boldsymbol{q}_{i-1}\|\boldsymbol{w}_{i-1}\end{aligned} \tag{1.51}$$

and the Broyden formula for the iterative change (1.48) can be written as

$$\delta\boldsymbol{q}_i = -\left[\prod_{j=0}^{i-1}\left(\boldsymbol{I} + \frac{\delta\boldsymbol{q}_{j+1}\delta\boldsymbol{q}_j^T}{\|\delta\boldsymbol{q}_j\|^2}\right)\right]\boldsymbol{H}_0^{-1}\boldsymbol{f}_i \tag{1.52}$$

However, this formula cannot be used directly, since $\delta \boldsymbol{q}_i$ appears on both sides of the equation

$$\begin{aligned} \delta \boldsymbol{q}_i &= -\left(\boldsymbol{I} + \frac{\delta \boldsymbol{q}_j \delta \boldsymbol{q}_{j-1}^T}{\|\delta \boldsymbol{q}_{j-1}\|^2} \right) \left[\prod_{j=0}^{i-2} \left(\boldsymbol{I} + \frac{\delta \boldsymbol{q}_{j+1} \delta \boldsymbol{q}_j^T}{\|\delta \boldsymbol{q}_j\|^2} \right) \right] \boldsymbol{H}_0^{-1} \boldsymbol{f}_i \\ &= -\left(\boldsymbol{I} + \frac{\delta \boldsymbol{q}_j \delta \boldsymbol{q}_{j-1}^T}{\|\delta \boldsymbol{q}_{j-1}\|^2} \right) \boldsymbol{H}_{i-1}^{-1} \boldsymbol{f}_i. \end{aligned} \tag{1.53}$$

Solving for $\delta \boldsymbol{q}_i$ gives

$$\delta \boldsymbol{q}_i = -\frac{\boldsymbol{H}_{i-1}^{-1} \boldsymbol{f}_i}{1 + \dfrac{\delta \boldsymbol{q}_{i-1}^T \boldsymbol{H}_{i-1}^{-1} \boldsymbol{f}_i}{\|\delta \boldsymbol{q}_{i-1}\|^2}}. \tag{1.54}$$

If there is no space to store the increasing number of vectors $\delta \boldsymbol{q}_i$ and their norms, one can restart the update process, i.e. clear the storage and start over. Another strategy is to replace the oldest of the stored vectors are replaced by the most resent. Such methods are called limited memory formulations in the optimizatation literature.

In the algorithm below, the index i is the iteration counter and the matrix update counter is denoted by k. The maximum values for iterates and matrix updates are $maxit$ and $kmax$, respectively.

Broyden's quasi-Newton algorithm: evaluate $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{q}_0)$, compute the initial residual $r_0 = \|\boldsymbol{f}_0\|$ and set $i = 0, k = -1$. Solve $\boldsymbol{H}_0 \delta \boldsymbol{q}_0 = -\boldsymbol{f}_0$. Iterate until convergence and $i < maxit$:

1. set $k = k + 1, i = i + 1$ and update $\boldsymbol{q}_i = \boldsymbol{q}_{i-1} + \delta \boldsymbol{q}_k$
2. evaluate $\boldsymbol{f}_i = \boldsymbol{f}(\boldsymbol{q}_i)$
3. if $k < kmax$ then
   (a) solve $\boldsymbol{H}_0 \boldsymbol{z} = -\boldsymbol{f}_i$
   (b) for $j = 0, k-1$ update $\boldsymbol{z} = \boldsymbol{z} + \delta \boldsymbol{q}_j \delta \boldsymbol{q}_{j-1}^T \boldsymbol{z} / \|\delta \boldsymbol{q}_{j-1}\|^2$
   (c) compute $\delta \boldsymbol{q}_k = \boldsymbol{z} / (1 + \delta \boldsymbol{q}_{k-1}^T \boldsymbol{z} / \|\delta \boldsymbol{q}_{k-1}\|^2)$
   (d) if $k = kmax$ then set $k = -1$ and solve $\delta \boldsymbol{q}_0 = -\boldsymbol{H}_0^{-1} \boldsymbol{f}_i$

Notice, that the initial matrix $\boldsymbol{H}_0$ need not to be the Jacobian of $\boldsymbol{f}$ at $\boldsymbol{q}_0$. It can be some approximation of it or even an identity matrix. This fact makes it an appealing alternative if the linear system is solved by iterative methods.

1.6 Line search

A line search procedure is often used in conjunction with quasi-Newton methods. It is meant as an inexpensive way to have an improved iterative direction. In a general finite element context it can be defined as a procedure to find a scalar multiplier η such that

$$G(\eta) = \delta \boldsymbol{q}^T \boldsymbol{f}(\boldsymbol{q} + \eta \delta \boldsymbol{q}) \approx 0. \tag{1.55}$$

The approximative sign in the above expression indicates that line search need not to be performed very accurately. Matthies and Strang [122] suggest the value $STOL = 0.5$ with the criteria

$$\left| \frac{G(\eta)}{G(0)} \right| < STOL. \tag{1.56}$$

If this tolerance is tightened, the number of internal force vector evaluations may increase drastically, thus requiring too much work with respect to the benefit obtained.

Algorithms for line search are presented in Refs. [122], [115], [49] .

1.7 Inexact Newton method

The inexact Newton method [57], [71]. [135] is a generalization of Newton's method. The idea is to find an iterative change $\delta \boldsymbol{q}$ and a scalar $\eta \in [0, 1)$ which satisfy

$$\|\boldsymbol{f}(\boldsymbol{q}) + \boldsymbol{f}'(\boldsymbol{q})\delta \boldsymbol{q}\| \leq \eta \|\boldsymbol{f}(\boldsymbol{q})\|. \tag{1.57}$$

In many implementations the *forcing term* η is specified first, and then $\delta \boldsymbol{q}$ is determined so that (1.57) holds. The purpose of choosing a proper value of the forcing term is to avoid oversolving the Newton step in the early phase of iteration. They are mostly used in connection with iterative linear solvers.

Exercises

1. Experiment the convergence behaviour of the Newton's method for the Mises truss when selecting the load factor as $\lambda = \sqrt{3}/9 \sin^3 \alpha$, which renders the Jacobian matrix singular at the root. Try also the chord Newton. What can be concluded.

2. Solve the diffusion-reaction problem (the Bratu problem) on a unit square $\Omega = [0, 1] \times [0, 1]$

$$-\Delta u = \lambda \exp(u) \quad \text{in} \quad \Omega, \qquad u = 0 \quad \text{on} \quad \partial\Omega. \tag{1.58}$$

   Experiment the convergence of Newton's method with the λ values $\lambda = 1, 4$ and 6.81.

Discretize the Laplacian operator by the five point difference scheme in a uniform grid

$$-\Delta u_{i,j} \approx h^{-2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}), \tag{1.59}$$

where $u_{i,j} = u(x_i, x_j)$ and $h = x_{i+1} - x_i = y_{i+1} - y_i$ is the grid spacing, $x_i = hi, y_i = hi, i = 0, \ldots m+1, h = 1/(m+1)$. The number of unknowns is thus $N = m^2$. Use at least two discretizations, e.g. $m = 10$ and $m = 20$.

Chapter 2

Parametrized non-linear equations

2.1 Continuation method

2.1.1 Basic algorithm

The load vector is usually parametrized by a single variable λ, the load parameter, defining the intensity of the load vector and the system (1.1) can be written as

$$\boldsymbol{f}(\boldsymbol{q}, \lambda) \equiv \boldsymbol{r}(\boldsymbol{q}) - \lambda \boldsymbol{p}(\boldsymbol{q}) = \boldsymbol{0}. \tag{2.1}$$

If the loads does not dependent on deformations, like in dead-weight loading, the system (1.1) reduces to

$$\boldsymbol{f}(\boldsymbol{q}, \lambda) = \boldsymbol{r}(\boldsymbol{q}) - \lambda \boldsymbol{p}_{\mathrm{r}} = \boldsymbol{0}, \tag{2.2}$$

where $\boldsymbol{p}_{\mathrm{r}}$ is the reference load vector. Equations (2.1) and (2.2) define a one dimensional equilibrium curve in a $N + 1$ dimensional displacement-load space. Procedures to trace such a one dimensional equilibrium path are called continuation or path following methods. They are incremental, step-wise algorithms. A typical continuation step includes the predictor and the corrector phases.

To traverse a solution path a proper parametrization is needed. Simple load control is the oldest type of parametrization. It is usually the most efficient one in the regular parts of a path. However, near the so called limit points, where the structure loses its load carrying capacity (at least locally), it breaks down. At the limit point the Jacobian matrix is singular and the load parameter is decreasing after such a point. A remedy is to change the control from the load parameter to some displacement component. Selecting the controlling displacement (or component from the scaled vector containing both displacements and the load parameter) to be the largest one from the last converged increment, results in a simple and reliable continuation procedure [60], [148]. Non-dimensionalizing of the variables is an essential point of this method.

A common setting of a continuation process is to augment the discrete equilibrium equations with a constraint. These constraints can be defined as path length measured in a

specified manner from the equilibrium point, or as a minimizing condition for the residual, or constraints in terms of incremental work, or constraints based on some orthogonality relation.

In many cases the displacement-load constraint can be defined by equation c in the following form:

$$\boldsymbol{h}(\boldsymbol{q},\lambda) = \left\{ \begin{array}{rcl} \boldsymbol{f}(\boldsymbol{q},\lambda) & = & \boldsymbol{0} \\ c(\boldsymbol{q},\lambda) & = & 0. \end{array} \right. \tag{2.3}$$

This kind of procedures are also commonly called arc-length methods. Using the Newton-Raphson linearization on the extended system (2.3) results in

$$\left\{ \begin{array}{rcl} \boldsymbol{f}'\delta\boldsymbol{q} + \dot{\boldsymbol{f}}\delta\lambda + \boldsymbol{f}(\boldsymbol{q},\lambda) & = & \boldsymbol{K}\delta\boldsymbol{q} - \boldsymbol{p}_r\delta\lambda + \boldsymbol{f} \;=\; \boldsymbol{0} \\ c'\delta\boldsymbol{q} + \dot{c}\delta\lambda + c(\boldsymbol{q},\lambda) & = & \boldsymbol{b}^T\delta\boldsymbol{q} + e\delta\lambda + c \;=\; 0 \end{array} \right. . \tag{2.4}$$

In order to utilize the spesific sparsity pattern of the tangent stiffness matrix $\boldsymbol{K}$, the solution of the augmented equations (2.4) is usually performed by using the following three phase block elimination method, also known as bordering algorithm [65, 101, 148, 154, 162, 174]:

1. solve $\boldsymbol{K}\delta\boldsymbol{q}_f = -\boldsymbol{f}$ and $\boldsymbol{K}\boldsymbol{q}_p = \boldsymbol{p}_r$,
2. compute $\delta\lambda = -(c + \boldsymbol{b}^T\delta\boldsymbol{q}_f)/(e + \boldsymbol{b}^T\boldsymbol{q}_p)$,
3. compute $\delta\boldsymbol{q} = \delta\boldsymbol{q}_f + \delta\lambda\boldsymbol{q}_p$.

In this format the solution of the linear equation system at phase 1 is performed by means of direct solvers. If iterative solvers are used, the nonsymmetric sparse format of the coefficient matrix in (2.4)

$$\boldsymbol{H}\delta\boldsymbol{y} = -\boldsymbol{h}, \quad \boldsymbol{H} = \left[\begin{array}{cc} \boldsymbol{K} & -\boldsymbol{p}_r \\ \boldsymbol{b}^T & e \end{array} \right], \quad \delta\boldsymbol{y} = \left\{ \begin{array}{c} \delta\boldsymbol{q} \\ \delta\lambda \end{array} \right\}, \quad \boldsymbol{h} = \left\{ \begin{array}{c} \boldsymbol{f} \\ c \end{array} \right\}, \tag{2.5}$$

is more appropriate. see refs. [5], [6], [7], [13], [42], [133], [134], [136].

Alternatively, the system (2.5) can be written as

$$(\boldsymbol{K} + e^{-1}\boldsymbol{p}_r\boldsymbol{b}^T)\delta\boldsymbol{q} = -\boldsymbol{f} - e^{-1}c\boldsymbol{p}_r \quad \text{and} \quad \delta\lambda = -e^{-1}(c + \boldsymbol{b}^T\delta\boldsymbol{q}). \tag{2.6}$$

Note that $(\boldsymbol{K} + e^{-1}\boldsymbol{p}_r\boldsymbol{b}^T)$ is a rank 1 modification of $\boldsymbol{K}$. Therefore, its inverse can easily be determined by the Sherman-Morrison formula, once the inverse of $\boldsymbol{K}$ is known. However, utilization of the Sherman-Morrison formula requires $e \neq 0$, while the block elimination strategy does not. The nonsingularity of $\boldsymbol{K}$ is required by both algorithms.

At regular points of the solution path the matrix $\boldsymbol{f}' = \boldsymbol{K}$ is nonsingular, and thus the solvability condition of the bordering algorihm as well as the nonsingulary of the augmented matrix $\boldsymbol{H}$ in (2.5) is guaranted if the Schur complement of $\boldsymbol{K}$ in $\boldsymbol{H}$ is nonzero:

$$e + \boldsymbol{b}^T\boldsymbol{q}_p = e + \boldsymbol{b}^T\boldsymbol{K}^{-1}\boldsymbol{p}_{\mathrm{r}} \neq 0. \tag{2.7}$$

At limit and bifurcation points the matrix $\boldsymbol{K}$ is singular. Nevertheless, the augmented matrix $\boldsymbol{H}$ is nonsingular at limit points. More precisely, if $\boldsymbol{K}$ is singular and $\mathrm{rank}(\boldsymbol{K}) = N-1$ then the augmented matrix $\boldsymbol{H}$ is nonsigular if and only if [102]

$$\boldsymbol{p}_{\mathrm{r}} \notin \mathrm{range}\boldsymbol{K} \quad \text{and} \quad \boldsymbol{b}^T \notin \mathrm{range}\boldsymbol{K}^T, \tag{2.8}$$

which are satisfied at limit points. These solvablity conditions (2.8) are equivalent to

$$\boldsymbol{p}_{\mathrm{r}}^T\boldsymbol{\psi} \neq 0, \quad \text{and} \quad \boldsymbol{b}^T\boldsymbol{\phi} \neq 0, \tag{2.9}$$

where $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ are the left, and right eigenvectors, respectively, i.e. satisfying $\boldsymbol{K}^T\boldsymbol{\psi} = \boldsymbol{0}$ and $\boldsymbol{K}\boldsymbol{\phi} = \boldsymbol{0}$. Conditions (2.9) can be easily verified by premultiplying the upper one of equations (2.4) by $\boldsymbol{\psi}^T$ and solving $\delta\lambda$ giving

$$\delta\lambda = \frac{\boldsymbol{\psi}^T\boldsymbol{f}}{\boldsymbol{\psi}^T\boldsymbol{p}_{\mathrm{r}}}. \tag{2.10}$$

and

$$\boldsymbol{K}\delta\boldsymbol{q} = -\boldsymbol{f} + \delta\lambda\boldsymbol{p}_{\mathrm{r}} = -\boldsymbol{f} + \frac{\boldsymbol{\psi}^T\boldsymbol{f}}{\boldsymbol{\psi}^T\boldsymbol{p}_{\mathrm{r}}}\boldsymbol{p}_{\mathrm{r}}. \tag{2.11}$$

At a limit point the matrix $\boldsymbol{K}$ is singular and the iterative change $\delta\boldsymbol{q}$ can thus be expressed as a sum of a vector $\boldsymbol{\phi}$ belonging to the nullspace of $\boldsymbol{K}$ and a particular solution $\tilde{\boldsymbol{q}}$, orthogonal to $\boldsymbol{\phi}$:

$$\delta\boldsymbol{q} = \tilde{\boldsymbol{q}} + \xi\boldsymbol{\phi}. \tag{2.12}$$

Substituting it to the lower one of equations (2.4), gives

$$\boldsymbol{b}^T(\tilde{\boldsymbol{q}} + \xi\boldsymbol{\phi}) + e\delta\lambda + c = 0 \quad \Rightarrow \quad \xi = -\frac{c + e\delta\lambda + \boldsymbol{b}^T\tilde{\boldsymbol{q}}}{\boldsymbol{b}^T\boldsymbol{\phi}}. \tag{2.13}$$

Continuation procedure with linear predictor and Newton like corrector iteration is also called the Euler-Newton method. Higher order correctors can also be used [7], [191].

2.1.2 Different constraint equations

Arc-length constraints

A large class of constraint equations can be written in the form

$$c(\boldsymbol{q}, \lambda) = \boldsymbol{t}^T\boldsymbol{C}\boldsymbol{n} - c_0 = 0 \tag{2.14}$$

where $\boldsymbol{t}$ and $\boldsymbol{n}$ are $n+1$ dimensional vectors and c_0 is a scalar. The weighting matrix $\boldsymbol{C}$ can be partitioned as

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{W} & \\ & \alpha^2 \end{bmatrix}, \tag{2.15}$$

where $\boldsymbol{W}$ is a positive definite or semidefinite diagonal matrix corresponding to displacements and α is a scaling factor. Updating the weight factors in $\boldsymbol{W}$ has proved to be beneficial for overall efficiency. Intuitively it can be understood easily, since then the process puts more weight on the most rapidly changing parts.

One of the first attempts to overcome limit points with augmented constraints is due to Haselgrove in 1961 [85], which remained for a long time undiscovered by structural engineers. Fried [74] presented again this procedure, which he called the orthogonal trajectory approach.

In structural mechanics the earliest developments are credited to Riks [150] and Wempner [195] in early 70's. They proposed a constraint in the form of a plane perpendicular to the prediction step.[1] This approach gained popularity only after a decade, when Ramm [145] and Crisfield [46] proposed the block elimination strategy. Ramm's procedure is a modification of the Riks-Wempner scheme, where the reference direction is updated to be the secant from the beginning of the present increment through the current point. However, fixing the reference direction, such as in the normal plane method, seems to stabilize the iteration process resulting in a more robust procedure.

Crisfield [46] uses a quadratic constraint, i.e. $\boldsymbol{t} = \boldsymbol{n}$ in (2.14), which means that iterations are constrained to the surface of an ellipsoid or a cylinder depending on the value of the scaling factor α ($\alpha \neq 0$ or $\alpha = 0$, respectively). Crisfield explicitely forces the cylindrical constraint at every iteration cycle, which results in a quadratic scalar equation for the solution of the load parameter change, in contrast to the linearized procedure of the block elimination phase 2, which can cause some ambiguity in the selection of the proper root. This feature was improved by Forde and Stiemer [72], who developed a scaling procedure for the consistently linearized version of the elliptical constraint to force the constraint at every iteration step.

An extension to the constraint equation (2.14) which combines the arc-length method with the pure displacement control is given by Runesson *et al.* [154]. In their approach the displacement vector in the constraint equation is decomposed into free and prescribed components.

If the Haselgrove-Fried orthogonal trajectory procedure is used with the chord Newton-Raphson scheme, it is identical with the Riks-Wempner-Ramm normal plane method. In the case of true Newton's method, the reference is made to a changing direction (as in the updated normal plane method), which might be a potential danger for oscillating behaviour. However, in computations made by the author oscillating behaviour was not observed when using the Haselgrove-Fried method with the true Newton iteration in analyzing elastic structures. [2]

Some of these variants can be expressed through a constraint of the form (2.14) and the formulas are given in Table 1, where references to more detailed descriptions are also given.

[1] The Haselgrove-Fried orthogonal trajectory method is also discussed in Riks's work [150].

[2] Some computations with elastic-plastic behaviour indicate that the normal plane method is more robust than the orthogonal residual procedure with respect to spurious unloading.

constraint	$\boldsymbol{t}^T$	$\boldsymbol{n}^T$	c_0	References
NP	$\boldsymbol{t}_1^T/\|\boldsymbol{t}_1\|_C$	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	Δs	[145], [150]
UNP	$\boldsymbol{t}_{i-1}^T/\|\boldsymbol{t}_{i-1}\|_C$	$[(\Delta\boldsymbol{q}_i)^T, \Delta\lambda_i]$	Δs	[145]
E	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	$[\Delta\boldsymbol{q}_i^T, \Delta\lambda_i]$	$(\Delta s)^2$	[46], [49]
VCP	$[(\Delta\boldsymbol{q}_i)^T, \Delta\lambda_i]$	$\boldsymbol{e}_k$	Δs	[60], [146], [148]
NP	normal plane			
UNP	updated normal plane			
E	elliptical			
VCP	variable control parameter			
$\boldsymbol{t}_j^T$	$[\Delta\boldsymbol{q}_j^T, \Delta\lambda_j]$			
Δ	incremental quantity			
δ	iterative change			
Δs	(pseudo) arc-length			
$\boldsymbol{e}_k$	a unit vector having a component 1 at position k			

Table 2.1: Expressions for different constraint types.

Allgower and Georg [6] have used a minimization condition

$$\min_{\boldsymbol{y}} \left\{ \|\boldsymbol{y} - \boldsymbol{y}_i\|_C \;\middle|\; \boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{0} \right\}, \tag{2.16}$$

where $\boldsymbol{y}_i = (\boldsymbol{q}_i{}^T, \lambda_i)^T$ is the current estimate of the solution, and $\|\cdot\|_C$ denotes the weighted Euclidean type vector seminorm $\|\boldsymbol{y}\| = \sqrt{\boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y}}$. The weight matrix $\boldsymbol{C}$ is defined in equation (2.15). The shortest distance from $\boldsymbol{y}_i$ to the equilibrium curve necessarily means that the tangent at the solution point $\boldsymbol{y}$ is orthogonal to the vector $\boldsymbol{y} - \boldsymbol{y}_i$, i.e. the solution $\boldsymbol{y}$ of (2.16) satisfies

$$\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{0}, \quad \text{with} \quad \boldsymbol{t}_i^T\boldsymbol{C}(\boldsymbol{y} - \boldsymbol{y}_i) = 0. \tag{2.17}$$

The tangent vector $\boldsymbol{t}^i$ is determined from equation

$$\boldsymbol{f}'(\boldsymbol{y}_i)\boldsymbol{t}_i = \boldsymbol{0}, \tag{2.18}$$

in which $\boldsymbol{f}' = \partial\boldsymbol{f}/\partial\boldsymbol{y}$. Note that here the tangent vector is not a unit vector as in ref. [6]. Linearization of (2.17) at $\boldsymbol{y}^i$ results in the system

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{y}_i) + \boldsymbol{f}'(\boldsymbol{y}_i)\delta\boldsymbol{y} &= \boldsymbol{0}, & (2.19)\\ \boldsymbol{t}_i^T\boldsymbol{C}\delta\boldsymbol{y} &= 0, & (2.20) \end{aligned}$$

which can also be solved with the block factorization strategy. This algorithm was proposed already by Haselgrove [85] in 1961 and later by Fried [74], who named it the

orthogonal trajectory method. It has proven to be very efficient in geometrically non-linear problems when used with the full Newton-Raphson iteration. When using the chord (modified) Newton's method it is identical with the Riks approach [150]. It should be mentioned that the system (2.19) cannot be written in the form of (2.4) or (2.5).

Special adaptation for iterative solvers

Walker [193] has proposed a strategy where the constraint is introduced within the iterates of the linear solution. If the constraint is defined as $c = \boldsymbol{t}^T \delta \boldsymbol{y} = 0$, the procedure is as follows [193]:

1. Find $\boldsymbol{Q} \in \mathbb{R}^{(N+1)\times N}$ such that range$(\boldsymbol{Q}) = \{\boldsymbol{t}\}^\perp$ and $\|\boldsymbol{Qq}\| = \|\boldsymbol{q}\|$ for all $\boldsymbol{q} \in \mathbb{R}^N$. Then $\bar{\boldsymbol{H}}\,\boldsymbol{Q} \in \mathbb{R}^{N\times N}$, where $\bar{\boldsymbol{H}} = [\boldsymbol{K}, -\boldsymbol{p}_r] = \boldsymbol{f}'(\boldsymbol{y})$.

2. Apply Krylov subspace method to solve approximately $\bar{\boldsymbol{H}}\,\boldsymbol{Q}\delta\boldsymbol{q} = -\boldsymbol{f}$.
Then set $\delta\boldsymbol{y} = \boldsymbol{Q}\delta\boldsymbol{q}$.

To specify the matrix $\boldsymbol{Q}$, the following scheme based on Householder transformations is proposed in [193]:

1. Choose $i, 1 \le i \le N+1$ and let $\boldsymbol{e}_i$ be the ith column of $\boldsymbol{I} \in \mathbb{R}^{(N+1)\times(N+1)}$. Determine the Householder transformation $\boldsymbol{P}$ such that $\boldsymbol{Pt} = \pm\boldsymbol{e}_i$.

2. Set $\boldsymbol{Q} = \boldsymbol{P}\hat{\boldsymbol{I}}_i$, where $\hat{\boldsymbol{I}}$ is obtained by deleting the ith column of $\boldsymbol{I}$.

As reported in [193] the most successful choice has been $\boldsymbol{Q} = \boldsymbol{P}\hat{\boldsymbol{I}}_{N+1}$, which is perhaps rather natural due to the special nature of the control parameter.

Incremental work and load constraints

Krenk [110] introduced another type of orthogonality constraint equation. His orthogonal residual approach does not need the block factorization scheme, only solution with one right-hand side per iteration step is required, and it is thus ideally suited with the use of iterative linear equation solvers. As argued by Krenk, the magnitude of the displacement increment is optimal when the orthogonality condition

$$\Delta\boldsymbol{q}_i^T \tilde{\boldsymbol{f}}_{i+1} = 0 \tag{2.21}$$

is satisfied. This linear condition is used to determine the current load parameter λ. The algorithm can be described briefly as:

1. compute: $\boldsymbol{r}_i = \boldsymbol{r}(\boldsymbol{q}_0 + \Delta\boldsymbol{q}_i),\ \Delta\boldsymbol{r}_i = \boldsymbol{r}_i - \lambda_0\boldsymbol{p}_r,\ \Delta\lambda_{i+1} = \Delta\boldsymbol{q}_i^T\Delta\boldsymbol{r}_i / \Delta\boldsymbol{q}_i{}^T\boldsymbol{p}_r$,

2. solve: $\boldsymbol{K}\delta\boldsymbol{q}_{i+1} = -\tilde{\boldsymbol{f}}_{i+1} = (\lambda_0 + \Delta\lambda_{i+1})\boldsymbol{p}_r - \boldsymbol{r}_i$,

3. compute: $\Delta\boldsymbol{q}_{i+1} = \Delta\boldsymbol{q}_i + \delta\boldsymbol{q}_{i+1}$.

λ^0 and $\boldsymbol{q}^0$ denote the load level and the displacement vector at the beginning of current increment. However, even if this elegant algorithm seems to be ideal with the use of an iterative linear equation solver, it has some drawbacks observed in numerical experiments. Since the size of the increment is not restricted during the iteration, the algorithm seems to have some tendency to increase the displacement increment near limit points. To remedy this deficiency Krenk imposed a maximum length for the iterative displacement increment. However, the rejection usually downgrades the convergence of the scheme. It might be a better choice to control the current iterate when already computing the value of the incremental load parameter in phase 1, if the load increment is beyond some safeguard values. In such situations, a standard Newton step could result in an acceptable displacement increment in phase 2.

Krenk and Hededal [111] combined a single cycle BFGS scheme, also called as memoryless BFGS, with the orthogonal residual procedure. They named this modification dual orthogonal residual method.

Bergan [24] has introduced a method in which the load step is adjusted by minimizing the norm of the unbalanced force:

$$\min_{\boldsymbol{y}} \|\boldsymbol{f}(\boldsymbol{y})\| = \min_{\boldsymbol{q},\lambda} \|\boldsymbol{f}(\boldsymbol{q},\lambda)\|, \tag{2.22}$$

in which the norm should be a weighted one.

Constant incremental work control method has been proposed by Karamanlidis *et al.* [96] and Bathe and Dvorkin [17]. The constant increment of work is fixed in the predictor step and during the corrector iteration phase the constraint is

$$\delta W = (\lambda + \tfrac{1}{2}\delta\lambda)\boldsymbol{p}_r^T \delta\boldsymbol{q} = 0. \tag{2.23}$$

Luckily this quadratic constraint will result in a double root for the iterative change of the load parameter

$$\delta\lambda = -\frac{\boldsymbol{p}_r^T \delta\boldsymbol{q}_f}{\boldsymbol{p}_r^T \boldsymbol{q}_p}. \tag{2.24}$$

It should also be noted, that the constant incremental work constraint will result in a symmetric augmented matrix $\boldsymbol{H}$ in eq. (2.5). It is easily seen that in the case of single load component in the reference load vector the work constraint method controls only single displacement. Therefore it can fail in certain snap-back situations where the denominator of (2.24) is nearly zero. Chen and Blandford [44] claimed the work control method to be superior to other published solution strategies. In the opinion of present author, this claim is largely exaggerated.

A number of other modifications of these basic procedures have been introduced in the literature. However, in the opinion of the author, the very specific form of the constraint equation, if properly posed, is of secondary importance. Efficiency of the solution algorithm depends mainly on the selection of proper step-size and the updating strategy of the tangent stiffness matrix.

2.1.3 Some computational aspects

Step length adaptation

Step length selection is one of the most crucial parts of continuation in view of efficiency. Several methods exist, see e.g. [6], [86], [145], [161], [163]. In engineering literature Ramm's simple method seems to be the most popular. The new step size is scaled by relating the number of iterations used in the previous step (I_n) to a desired value I_d:

$$\Delta s_{n+1} = (I_d/I_n)^p \Delta s_n, \tag{2.25}$$

where the scaling exponent $p = 0.5$ is usually adopted. If the desired number of corrector iterations is properly chosen this simple arc-length control will result in a rather robust procedure. However, it can produce too small step-sizes which are kept unchanged for unnecessarily long times. Some safeguard limit values for the step-length changes should also be used together with (2.25).

Alternative choices are presented by Den Heijer and Rheinboldt [86] where the steplength adaptation strategy is based on error models obtained by analyzing the Newton-Kantorovich theory and by Georg [76] which is based upon asymptotic estimates. In the latter approach the steplength is continually adapted so that a nominal prescribed contraction rate is maintained. An approach with similar characteristics is also proposed in ref. [64].

Orientation of the curve tracing is also determined in the predictor phase. The sign of the load increment can be determined by monitoring either the inertia of the tangent stiffness matrix or the angle between the predictor step and the previous increment. The latter approach is suitable for continuation with iterative linear solvers.

The predictor step proceeds from a known equilibrium configuration, i.e. a point of the equilibrium path, towards the next point on the path. A common practice is to use an Euler predictor, a predictor step in the direction of the tangent of the path. Higher order predictors are also possible. Performance studies of different improved predictor schemes can be found e.g. in refs. [5],[62], [191]. However, simple Euler predictor is usually the most effective.

Stiffness matrix update strategy

For moderate size problems computation of the stiffness matrix is the most time consuming part in the continuation process. On the other hand, the solution of the linearized equation system dominates the computational cost for large problems. If direct linear equation solvers are used, the factorization is the dominating phase, which has to be performed after every stiffness matrix update. Therefore, proper update strategy has a pronounced effect on the computational cost.

In the present lecture notes, two different strategies are considered. The first one is based simply on higher order Newton schemes. If the number of chord Newton steps following a full Newton step is increasing with the iteration number, it will result in a robust,

automatic and rather efficient procedure without any user given adjustment parameters. The higher order method, also called Shamanski method in ref. [104], is worthwhile to apply when direct linear equation solvers are used. The efficiency is due to the fact that the effort used in the stiffness matrix updates near the equilibrium point is usually wasted since the convergence could be obtained with few chord Newton steps, which are much cheaper than one full Newton step.

Another strategy is to monitor the convergence rate:

$$q = \frac{\|\delta \boldsymbol{x}_i\|_C}{\|\delta \boldsymbol{x}_{i-1}\|_C}.$$

If it is small enough, then the chord Newton is used and in the opposite case the stiffness matrix is updated. This approach will require a user specified convergence rate tolerance, for which a reasonable value is of the order 10^{-1}.

Different corrector iteration strategies can also used. A class of Newton algorithms called quasi-Newton methods have been developed in order to speed up the convergence of the chord Newton method and to improve the efficiency of the true Newton scheme see [51], .

Eigenvector projections can also be utilized to improve the convergence of the corrector iterations [58].

2.1.4 Continuation pseudocode

Continuation algorithm with block elimination method.

1. Predictor phase
   (a) starting point $(\boldsymbol{q}_0, \lambda_0)$ such that $\boldsymbol{f}(\boldsymbol{q}_0, \lambda_0) = \boldsymbol{0}$.
   (b) choose steplength Δs
   (c) evaluate $\boldsymbol{K}_0 = \boldsymbol{f}'(\boldsymbol{q}_0, \lambda_0) = \partial \boldsymbol{f}/\partial \boldsymbol{q}|_0$
   (d) factorize $\boldsymbol{K}_0 = \boldsymbol{LDU}$ and compute singularity test functions (see ch. 5)
   (e) solve $\boldsymbol{K}_0 \boldsymbol{q}_p = \boldsymbol{p}_r$
   (f) compute $\Delta\lambda_0$ from the constraint equation c
   (g) select the direction for traversing $\delta\lambda_0 = \pm\Delta\lambda_0$
   (h) update $\boldsymbol{q}_1 = \boldsymbol{q}_0 + \delta\lambda_0 \boldsymbol{q}_p$ and $\lambda_1 = \lambda_0 + \delta\lambda_0$
2. Corrector phase: iterate $i = 1, \ldots,$
   (a) evaluate $\boldsymbol{f}_i = \boldsymbol{f}(\boldsymbol{q}_i, \lambda_i)$
   (b) decide if the Jacobian should be updated
   if yes, then compute $\boldsymbol{K}_k = \boldsymbol{f}'(\boldsymbol{q}_i, \lambda_i)$ and factorize $\boldsymbol{K}_k$

(c) solve $\boldsymbol{K}_k \delta \boldsymbol{q}_f = -\boldsymbol{f}_i$ and $\boldsymbol{K}_k \boldsymbol{q}_p = \boldsymbol{p}_r$

(d) compute $\delta\lambda_i = -(c + \boldsymbol{c}^T \delta \boldsymbol{q}_f)/(e + \boldsymbol{c}^T \boldsymbol{q}_p)$ and $\delta \boldsymbol{q}_i = \delta \boldsymbol{q}_f + \delta\lambda_i \boldsymbol{q}_p$

(e) update $\boldsymbol{q}_{i+1} = \boldsymbol{q}_i + \delta \boldsymbol{q}_i$ and $\lambda_{i+1} = \lambda_i + \delta\lambda_i$

(f) if convergence set $\boldsymbol{q}_0 = \boldsymbol{q}_{i+1}$ and $\lambda_0 = \lambda_{i+1}$ and go to a new predictor step 1.

Exercises

1. Solve the equilibrium path of the Mises-truss example 1.3.1. Compare some constraint equations, like the normal plane and the elliptical one.

2. Solve the solution path of the Bratu problem 2 on page 14. The path has one limit point at $\lambda \approx 6.81$. Stop traversing the path when $\|u\|_\infty = 10$.

Chapter 3

Determination of critical points

3.1 Non-linear eigenvalue problem

A critical point along an equilibrium path can be determined by solving the non-linear eigenvalue problem: find the critical value of $\boldsymbol{q}, \lambda$ and the corresponding eigenvector $\boldsymbol{v}$ such that

$$\left\{ \begin{array}{rcl} \boldsymbol{f}(\boldsymbol{q},\lambda) & = & \boldsymbol{0} \\ \boldsymbol{f}'(\boldsymbol{q},\lambda)\boldsymbol{\phi} & = & \boldsymbol{0} \end{array} \right. \tag{3.1}$$

where $\boldsymbol{f}$ is the vector of unbalanced forces and $\boldsymbol{f}'$ denotes the Gateaux derivative (Jacobian matrix) with respect to the state variables $\boldsymbol{q}$. Equation $(3.1)_1$ is the equilibrium equation, which has to be satisfied at the critical point, and equation $(3.1)_2$ states the zero stiffness in the direction of the critical eigenmode $\boldsymbol{\phi}$, which is the actual criticality condition. Such a system is considered in Refs. [167], [196], [197]. Abbot [1] considers a different extended system where the criticality is identified by means of the determinant of the tangent stiffness matrix. The drawback of this procedure is that the directional derivative of the determinant is difficult to compute.

The system (3.1) consists of $2N+1$ unknowns, the displacement vector $\boldsymbol{q}$, the eigenmode $\boldsymbol{v}$ and the load parameter value λ at the critical state. Since the eigenvector $\boldsymbol{v}$ is defined uniquely up to a constant, the normalizing condition can be added to the system (3.1):

$$\boldsymbol{g}(\boldsymbol{q},\boldsymbol{\phi},\lambda) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q},\lambda) \\ \boldsymbol{K}(\boldsymbol{q},\lambda)\boldsymbol{\phi} \\ N(\boldsymbol{\phi}) \end{array} \right\} = \boldsymbol{0}, \tag{3.2}$$

where the Jacobian matrix $\boldsymbol{f}' = \partial \boldsymbol{f} / \partial \boldsymbol{q}$ is denoted by $\boldsymbol{K}$ and $N(\boldsymbol{\phi})$ defines some normalizing condition to the eigenvector.

The idea of augmenting the equilibrium equations with the criticality condition appears to be due to Keener and Keller [99], presented as early as in 1973. Most papers found in literature deal only with simple critical points, and the extension to multiple bifurcations, see Keener [98], will not be considered in these lecture notes.

3.2 Direct method for non-linear eigenvalue problem

For the stable solution of a non-linear system (3.2) using a Newton's method, it is important that the solution is isolated. Therefore the use of system (3.2) seems to be limited to the computation of limit points only [196]. However, it has been used also to compute bifurcation points in Refs. [167, 197].

The main problem in using Newton's method to the system (3.2) is the computation of the directional derivative of the tangent stiffness matrix. Finite differences are usually used for the approximation of the directional derivative, [61, 63, 117, 118, 119, 197]. The following description is adopted from Wriggers and Simo [197]. They employed the penalty regularization to improve the conditioning of the Jacobian of the extended system, appending the constraint $\boldsymbol{e}_i^T \boldsymbol{q} = \mu$ to the system (3.2):

$$\tilde{\boldsymbol{g}}(\boldsymbol{x}) = \tilde{\boldsymbol{g}}(\boldsymbol{q}, \boldsymbol{\phi}, \lambda, \mu) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q}, \lambda) + \gamma(\boldsymbol{e}_i^T \boldsymbol{q} - \mu)\boldsymbol{e}_i \\ \boldsymbol{K}(\boldsymbol{q}, \lambda)\boldsymbol{\phi} + \gamma(\boldsymbol{e}_i^T \boldsymbol{\phi} - \phi_0)\boldsymbol{e}_i \\ \boldsymbol{e}_i^T \boldsymbol{\phi} - \phi_0 \\ \boldsymbol{e}_i^T \boldsymbol{q} - \mu \end{array} \right\} = \boldsymbol{0}, \tag{3.3}$$

where γ is the non-negative regularizing penalty parameter and $\boldsymbol{e}_i$ is a unit vector having the unit value at position i corresponding to the smallest diagonal entry of the factorized tangent stiffness matrix. The Newton linearization step of system (3.3) results in a linear equation system of the following form:

$$\left[\begin{array}{cccc} \boldsymbol{K}_\gamma & \boldsymbol{0} & -\boldsymbol{p}_r & -\gamma \boldsymbol{e}_i \\ \dfrac{\partial}{\partial \boldsymbol{q}}(\boldsymbol{K}\boldsymbol{\phi}) & \boldsymbol{K}_\gamma & \dfrac{\partial}{\partial \lambda}(\boldsymbol{K}\boldsymbol{\phi}) & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{e}_i^T & 0 & 0 \\ \boldsymbol{e}_i^T & \boldsymbol{0}^T & 0 & -1 \end{array} \right] \left\{ \begin{array}{c} \delta \boldsymbol{q} \\ \delta \boldsymbol{\phi} \\ \delta \lambda \\ \delta \mu \end{array} \right\} = -\tilde{\boldsymbol{g}}(\boldsymbol{q}, \boldsymbol{\phi}, \lambda, \mu), \tag{3.4}$$

where the rank-one updated tangent matrix is

$$\boldsymbol{K}_\gamma = \boldsymbol{K} + \gamma \boldsymbol{e}_i \boldsymbol{e}_i^T. \tag{3.5}$$

If the system (3.4) is to be solved by a direct linear solver, a block factorization type strategy is feasible. Solution of $\delta \boldsymbol{q}$ is obtained by solving the following three systems of linear equations:

$$\boldsymbol{K}_\gamma \delta \boldsymbol{q}_p = \boldsymbol{p}_r, \quad \boldsymbol{K}_\gamma \delta \boldsymbol{q}_f = -\boldsymbol{f}, \quad \boldsymbol{K}_\gamma \delta \boldsymbol{q}_e = \boldsymbol{e}_i, \tag{3.6}$$

and thus

$$\delta \boldsymbol{q} = \delta\lambda \delta \boldsymbol{q}_p + \delta \boldsymbol{q}_f + \gamma(\mu + \delta\mu - \boldsymbol{e}_i^T \boldsymbol{q}) \delta \boldsymbol{q}_e. \tag{3.7}$$

Change in the eigenvector is computed from the second equation in (3.4)

$$\delta \boldsymbol{\phi} = -\boldsymbol{\phi} - \boldsymbol{K}_\gamma^{-1} \left[\frac{\partial}{\partial \boldsymbol{q}}(\boldsymbol{K}\boldsymbol{\phi}) \delta \boldsymbol{q} + \frac{\partial}{\partial \lambda}(\boldsymbol{K}\boldsymbol{\phi}) \delta\lambda - \gamma \phi_0 \boldsymbol{e}_i \right]. \tag{3.8}$$

At this stage the evaluation of the second derivatives of the residual (i.e. the directional derivatives of the tangent stiffness matrix) has to be performed. Introducing the vectors

$$\boldsymbol{h}_i = -\frac{\partial}{\partial \boldsymbol{q}}(\boldsymbol{K}\boldsymbol{\phi})\delta\boldsymbol{q}_i, \quad ,i = p, f, e \quad \boldsymbol{h}_\lambda = -\frac{\partial}{\partial \lambda}(\boldsymbol{K}\boldsymbol{\phi}), \tag{3.9}$$

and

$$\boldsymbol{w}_i = \boldsymbol{K}_\gamma^{-1}\boldsymbol{h}_i, \quad i = p, f, e, \lambda \tag{3.10}$$

the expression for the iterative change of the eigenvector can be written in terms of vectors $\boldsymbol{w}_i, \delta\boldsymbol{q}_i$. The load vectors $\boldsymbol{h}_i$ can be computed at element level and they are similar to the load vectors in Koiter's initial post-buckling approach. Using the notation of (3.9) and (3.10) the new value for the eigenvector is

$$\boldsymbol{\phi} + \delta\boldsymbol{\phi} = \delta\lambda(\boldsymbol{w}_p + \boldsymbol{w}_\lambda) + \boldsymbol{w}_f + \gamma(\mu + \delta\mu - \boldsymbol{e}_i^T\boldsymbol{q})\boldsymbol{w}_e + \phi_0\delta\boldsymbol{q}_e. \tag{3.11}$$

The final step is to solve the scalar parameters $\delta\lambda$ and $\delta\mu$ from the two remaining equations in (3.4):

$$\begin{bmatrix} \boldsymbol{e}_i^T(\boldsymbol{w}_p + \boldsymbol{w}_\lambda) & \gamma\boldsymbol{e}_i^T\boldsymbol{w}_e \\ \boldsymbol{e}_i^T\delta\boldsymbol{q}_p & \gamma\boldsymbol{e}_i^T\delta\boldsymbol{q}_e - 1 \end{bmatrix} \begin{Bmatrix} \delta\lambda \\ \delta\mu \end{Bmatrix} = \begin{Bmatrix} g_1 \\ g_2 \end{Bmatrix}, \tag{3.12}$$

where

$$\begin{aligned} g_1 &= \phi_0 - \boldsymbol{e}_i^T\left\{\boldsymbol{w}_f + \gamma\left[\phi_0\delta\boldsymbol{q}_e + (\mu - \boldsymbol{e}_i^T\boldsymbol{q})\boldsymbol{w}_e\right]\right\}, \\ g_2 &= \mu - \boldsymbol{e}_i^T\left[\boldsymbol{q} + \delta\boldsymbol{q}_f + \gamma(\mu - \boldsymbol{e}_i^T\boldsymbol{q})\delta\boldsymbol{q}_e\right]. \end{aligned}$$

Application of the described method requires complete description of the kinematical relations especially in cases where the pre-buckling state is non-linear and exhibits large deflections and rotations. In particular, the description has to be capable to handle large incremental rotations.

3.3 Polynomial eigenvalue problem

Assuming an equilibrium state $(\boldsymbol{q}_*, \lambda_*)$ with a regular tangent matrix, a Taylor expansion of the non-linear eigenvalue problem (3.1) with respect to the load parameter λ has the form

$$\boldsymbol{q} = \boldsymbol{q}_* + \Delta\lambda\boldsymbol{q}_1 + \tfrac{1}{2}(\Delta\lambda)^2\boldsymbol{q}_2 + \cdots, \tag{3.13}$$

$$\boldsymbol{f} = \boldsymbol{f}_* + \Delta\lambda\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\lambda}\bigg|_* + \frac{1}{2}(\Delta\lambda)^2\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}\lambda^2}\bigg|_* + \cdots = \boldsymbol{0} \tag{3.14}$$

$$\left(\boldsymbol{f}'_* + \Delta\lambda\frac{\mathrm{d}\boldsymbol{f}'}{\mathrm{d}\lambda}\bigg|_* + \frac{1}{2}(\Delta\lambda)^2\frac{\mathrm{d}^2\boldsymbol{f}'}{\mathrm{d}\lambda^2}\bigg|_* + \cdots\right)\boldsymbol{\phi} = \boldsymbol{0} \tag{3.15}$$

where $\Delta\lambda = \lambda - \lambda_*$. Expressions for the derivatives are [1]

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\lambda} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \lambda} + \frac{\partial \boldsymbol{f}}{\partial \lambda} = \boldsymbol{f}'\dot{\boldsymbol{q}} + \dot{\boldsymbol{f}}, \tag{3.16}$$

$$\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}\lambda^2} = \boldsymbol{f}'\ddot{\boldsymbol{q}} + \boldsymbol{f}''\dot{\boldsymbol{q}}\dot{\boldsymbol{q}} + 2\dot{\boldsymbol{f}}'\dot{\boldsymbol{q}} + \ddot{\boldsymbol{f}}, \tag{3.17}$$

$$\frac{\mathrm{d}\boldsymbol{f}'}{\mathrm{d}\lambda} = \boldsymbol{f}''\dot{\boldsymbol{q}} + \dot{\boldsymbol{f}}', \tag{3.18}$$

$$\frac{\mathrm{d}^2\boldsymbol{f}'}{\mathrm{d}\lambda^2} = \boldsymbol{f}''\ddot{\boldsymbol{q}} + \boldsymbol{f}'''\dot{\boldsymbol{q}}\dot{\boldsymbol{q}} + 2\dot{\boldsymbol{f}}''\dot{\boldsymbol{q}} + \ddot{\boldsymbol{f}}'. \tag{3.19}$$

Evaluating these quantities at the equilibrium state $(\boldsymbol{q}_*, \lambda_*)$, gives

$$\dot{\boldsymbol{q}}_* = \boldsymbol{q}_1, \quad \text{and} \quad \ddot{\boldsymbol{q}}_* = \boldsymbol{q}_2, \quad \text{etc..} \tag{3.20}$$

and the expressions (3.16-3.19) result in

$$\left.\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\lambda}\right|_* = \boldsymbol{f}'_*\boldsymbol{q}_1 + \dot{\boldsymbol{f}}_*, \tag{3.21}$$

$$\left.\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}\lambda^2}\right|_* = \boldsymbol{f}'_*\boldsymbol{q}_2 + \boldsymbol{f}''_*\boldsymbol{q}_1\boldsymbol{q}_1 + 2\dot{\boldsymbol{f}}'_*\boldsymbol{q}_1 + \ddot{\boldsymbol{f}}_*, \tag{3.22}$$

$$\left.\frac{\mathrm{d}\boldsymbol{f}'}{\mathrm{d}\lambda}\right|_* = \boldsymbol{f}''_*\boldsymbol{q}_1 + \dot{\boldsymbol{f}}'_*, \tag{3.23}$$

$$\left.\frac{\mathrm{d}^2\boldsymbol{f}'}{\mathrm{d}\lambda^2}\right|_* = \boldsymbol{f}''_*\boldsymbol{q}_2 + \boldsymbol{f}'''_*\boldsymbol{q}_1\boldsymbol{q}_1 + 2\dot{\boldsymbol{f}}''_*\boldsymbol{q}_1 + \ddot{\boldsymbol{f}}'_*, \tag{3.24}$$

where $\boldsymbol{f}_* = \boldsymbol{f}(\boldsymbol{q}_*, \lambda_*)$ etc. In the expansion of the equilibrium equations (3.14) all terms $\mathrm{d}^p\boldsymbol{f}/\mathrm{d}\lambda^p, p = 1, 2, \ldots$ has to vanish, thus giving the equation to solve the fields $\boldsymbol{q}_i$

$$\boldsymbol{f}'_*\boldsymbol{q}_1 = -\dot{\boldsymbol{f}}_*, \tag{3.25}$$

$$\boldsymbol{f}'_*\boldsymbol{q}_2 = -\left[\boldsymbol{f}''_*\boldsymbol{q}_1\boldsymbol{q}_1 + 2\dot{\boldsymbol{f}}'_*\boldsymbol{q}_1 + \ddot{\boldsymbol{f}}_*\right] \tag{3.26}$$

$$\vdots \tag{3.27}$$

It is worthwhile to notice that the coefficent matrix to solve $\boldsymbol{q}_1, \boldsymbol{q}_2 \ldots$ is the same for all cases. In structural mechanics, the symbol $\boldsymbol{K}$ is usually used to denote the stiffness matrix, thus the matrices in (3.15) can be written as

$$\boldsymbol{K}_{0|*} = \boldsymbol{f}'_*,$$

$$\boldsymbol{K}_{1|*} = \left.\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\lambda}\right|_* = \boldsymbol{f}''_*\boldsymbol{q}_1 + \dot{\boldsymbol{f}}'_*,$$

$$\boldsymbol{K}_{2|*} = \left.\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}\lambda^2}\right|_* = \boldsymbol{f}''_*\boldsymbol{q}_2 + \boldsymbol{f}'''_*\boldsymbol{q}_1\boldsymbol{q}_1 + 2\dot{\boldsymbol{f}}''_*\boldsymbol{q}_1 + \ddot{\boldsymbol{f}}'_*,$$

[1] Notice the difference between derivatives $\mathrm{d}\boldsymbol{f}/\mathrm{d}\lambda$ and $\dot{\boldsymbol{f}} = \partial\boldsymbol{f}/\partial\lambda$, i.e. $\mathrm{d}\boldsymbol{f}/\mathrm{d}\lambda = \boldsymbol{f}'(\partial\boldsymbol{q}/\partial\lambda) + \partial\boldsymbol{f}/\partial\lambda$.

and the polynomial eigenvalue problem can be written as

$$\left(\boldsymbol{K}_{0|*} + \Delta\lambda \boldsymbol{K}_{1|*} + \tfrac{1}{2}(\Delta\lambda)^2 \boldsymbol{K}_{2|*} + \cdots\right) \boldsymbol{\phi} = \boldsymbol{0}, \tag{3.28}$$

In the classical linear stability analysis the reference state is the undeformed stress free configuration. For the linear stability eigenvalue problem the matrices are simply the following:[2]

$$\begin{aligned} \boldsymbol{K}_{0|0} &= \boldsymbol{f}'(\boldsymbol{0}, 0) \\ \boldsymbol{K}_{1|0} &= \boldsymbol{f}''(\boldsymbol{0}, 0)\boldsymbol{q}_1, \end{aligned}$$

where $\boldsymbol{K}_{0|0}\boldsymbol{q}_1 = \boldsymbol{p}_r$. Therefore the strains are linear functions of the displacements $\boldsymbol{q}_1$ and the geometric stiffness matrix $\boldsymbol{K}_{1|0}$ is a linear function of the displacements $\boldsymbol{q}_1$.

It is seen from the definition of the $\boldsymbol{K}_{1|0}$ matrix that the "initial stress" state to the linear eigenvalue problem has to be linear with respect to the load parameter change. This is not true if the linear stability eigenvalue problem is solved from

$$\left(\boldsymbol{K}_{0|*} + s(\boldsymbol{K}_{0|*} - \boldsymbol{K}_{0|**})\right) \boldsymbol{\phi} = \boldsymbol{0},$$

where $\boldsymbol{K}_{0|*}$ and $\boldsymbol{K}_{0|**}$ are the tangent stiffness matrices from two consecutive equilibrium states. It will be a correct approximation to the linear eigenvalue problem only if the load increment $\Delta\lambda = \lambda_* - \lambda_{**}$ is small, i.e. $\boldsymbol{K}_{1|*} \approx (\Delta\lambda)^{-1}(\boldsymbol{K}_{0|*} - \boldsymbol{K}_{0|**})$.

Example 3.3.1. *The same Mises truss as discussed in example 1.3.1 on page 6 will be considered. The symmetric path exhibits snap-through behaviour. There are now other equilibrium paths for the symmetric load if the angle $\alpha < 54.74°$. Compute the snap-through load and displacement using the direct method amd polynomial approximations of first and second order. Length and the initial angle of the bars at the initial state are L and α, respectively.*

The equilibrium equation (1.30) of the symmetric mode is (divided by two)

$$f(q, \lambda) = (\sin^2\alpha)q - \tfrac{3}{2}(\sin\alpha)q^2 + \tfrac{1}{2}q^3 - \lambda = 0, \tag{3.29}$$

where $\lambda = P/EA$.

Direct method

The non-linear eigenvalue problem can be stated as: find λ_{cr} and q_{cr} such that

$$\boldsymbol{g}(q, \lambda) = \begin{cases} f(q, \lambda) = (\sin^2\alpha)q - \tfrac{3}{2}(\sin\alpha)q^2 + \tfrac{1}{2}q^3 - \lambda = 0 \\ f'(q, \lambda) = \sin^2\alpha - 3(\sin\alpha)q + \tfrac{3}{2}q^2 = 0 \end{cases} \tag{3.30}$$

[2] Assuming also dead weight loading, i.e. $\dot{\boldsymbol{f}}' \equiv \boldsymbol{0}$.

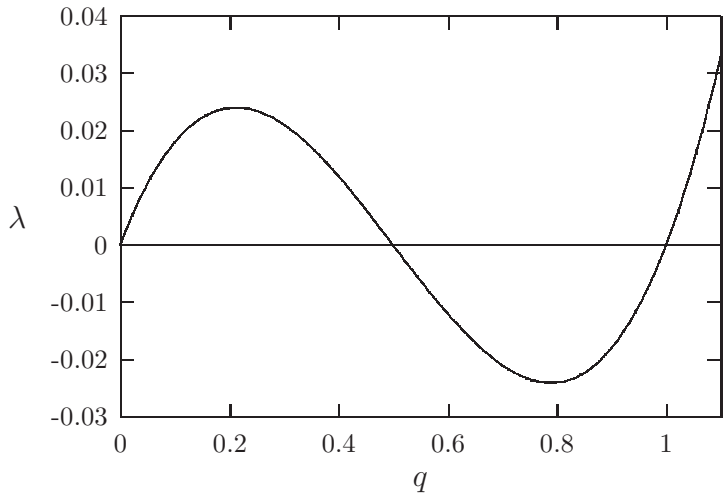


Figure 3.1: Equilibrium path of the Mises truss ($\alpha = 30°$).

This simple system of two unknowns can be solved analytically:

$$\begin{cases} q_{\mathrm{cr}}^{\mathrm{exact}} & = \frac{3-\sqrt{3}}{3} \sin\alpha \approx 0.4226 \sin\alpha \\ \lambda_{\mathrm{cr}}^{\mathrm{exact}} & = \frac{\sqrt{3}}{9} \sin^3\alpha \approx 0.1925 \sin^3\alpha \end{cases} . \tag{3.31}$$

In this case of single state variable, this non-linear eigenvalue problem can be solved by the Newton's method with two unknowns (q, λ), starting from the unloaded state ($q = 0, \lambda = 0$). The Newton step is to compute the iterative change ($\delta q, \delta\lambda$) from

$$\begin{bmatrix} f'(q_i,\lambda_i) & -1 \\ f''(q_i,\lambda_i) & 0 \end{bmatrix} \begin{Bmatrix} \delta q_i \\ \delta\lambda_i \end{Bmatrix} = - \begin{Bmatrix} f(q_i,\lambda_i) \\ f'(q_i,\lambda_i) \end{Bmatrix} \tag{3.32}$$

or shortly

$$\boldsymbol{G}_i \delta\boldsymbol{x}_i = -\boldsymbol{g}(\boldsymbol{x}_i) \tag{3.33}$$

First iteration:

$$\boldsymbol{G}_0 = \begin{bmatrix} \sin^2\alpha & -1 \\ -3\sin\alpha & 0 \end{bmatrix}, \quad \boldsymbol{g}_0 = \begin{Bmatrix} 0 \\ \sin^2\alpha \end{Bmatrix}, \quad \delta\boldsymbol{x}_0 = \begin{Bmatrix} \frac{1}{3}\sin\alpha \\ \frac{1}{3}\sin^2\alpha \end{Bmatrix}. \tag{3.34}$$

Second iteration:

$$\boldsymbol{G}_1 = \begin{bmatrix} \frac{1}{6}\sin^2\alpha & -1 \\ -2\sin\alpha & 0 \end{bmatrix}, \quad \boldsymbol{g}_1 = \begin{Bmatrix} -\frac{4}{27}\sin^3\alpha \\ \frac{1}{6}\sin^2\alpha \end{Bmatrix}, \quad \delta\boldsymbol{x}_1 = \begin{Bmatrix} \frac{1}{12}\sin\alpha \\ -\frac{29}{216}\sin^2\alpha \end{Bmatrix}. \tag{3.35}$$

Estimate to the solution after two iterations is thus

$$\boldsymbol{x}^2 = \boldsymbol{x}_1 + \delta\boldsymbol{x}_1 = \begin{Bmatrix} \frac{5}{12}\sin\alpha \\ \frac{43}{216}\sin^2\alpha \end{Bmatrix} \approx \begin{Bmatrix} 0.4167\sin\alpha \\ 0.1991\sin^2\alpha \end{Bmatrix}. \tag{3.36}$$

Third iteration:

$$\boldsymbol{G}_2 = \begin{bmatrix} \frac{1}{96}\sin^2\alpha & -1 \\ -\frac{21}{12}\sin\alpha & 0 \end{bmatrix}, \quad \boldsymbol{g}_2 = \begin{Bmatrix} -\frac{23}{3456}\sin^3\alpha \\ \frac{1}{96}\sin^2\alpha \end{Bmatrix}, \quad \delta\boldsymbol{x}_2 = \begin{Bmatrix} \frac{1}{168}\sin\alpha \\ -\frac{957}{145152}\sin^2\alpha \end{Bmatrix}. \tag{3.37}$$

After three iterations the solution vector has four correct significant digits:

$$\boldsymbol{x}_3 = \boldsymbol{x}_2 + \delta\boldsymbol{x}_2 = \begin{Bmatrix} \frac{71}{168}\sin\alpha \\ \frac{27939}{145152}\sin^2\alpha \end{Bmatrix} \approx \begin{Bmatrix} 0.4226\sin\alpha \\ 0.1925\sin^2\alpha \end{Bmatrix}. \tag{3.38}$$

Note, that the Jacobian $\boldsymbol{G}$ is regular at the solution point

$$\boldsymbol{G}(q_{\rm cr}^{\rm exact}, \lambda_{\rm cr}^{\rm exact}) = \begin{bmatrix} 0 & -1 \\ -\frac{\sqrt{3}}{3}\sin\alpha & 0 \end{bmatrix}. \tag{3.39}$$

Polynomial eigenvalue problem

To perform the linear stability analysis at the initial state, the reference displacement q_1 is solved from

$$f_0' q_1 = -\dot{f}_0 \tag{3.40}$$

Now $\dot{f} \equiv -1$ and $f_0' = \sin^2\alpha$, thus $q_1 = \sin^{-2}\alpha$. The initial stress matrix, or initial geometric stiffness matrix is

$$K_{1|0} = f_0'' q_1 + \dot{f}_0' = f_0'' q_1 = -\frac{3}{\sin\alpha} \tag{3.41}$$

and the eigenvalue problem

$$K_{0|0} + \lambda K_{1|0} = \sin^2\alpha - \lambda\frac{3}{\sin\alpha} = 0 \tag{3.42}$$

giving the result

$$\lambda_{\rm cr} = \tfrac{1}{3}\sin^3\alpha \tag{3.43}$$

For the quadratic eigenvalue problem, the displacement correction q_2 is solved from

$$f_0' q_2 = -(f_0'' q_1 q_1 + 2\dot{f}_0' q_1 + \ddot{f}_0) = -f_0'' q_1 q_1 \tag{3.44}$$

giving the result $q_2 = 3\sin^{-5}\alpha$. The stiffness "matrix" $K_{2|0}$ is computed from

$$K_{2|0} = f_0'' q_2 + f_0''' q_1 q_1 = -\frac{6}{\sin^4\alpha} \tag{3.45}$$

and the resulting quadratic eigenvalue problem is

$$K_{0|0} + \lambda K_{1|0} + \lambda^2 K_{2|0} = \sin^2\alpha - \lambda\frac{3}{\sin\alpha} - \lambda^2\frac{6}{\sin^4\alpha} = 0 \tag{3.46}$$

The positive root of this equation is

$$\lambda_{\rm cr} = \frac{1}{4}\big(\sqrt{\frac{11}{3}} - 1\big)\sin^3\alpha \tag{3.47}$$

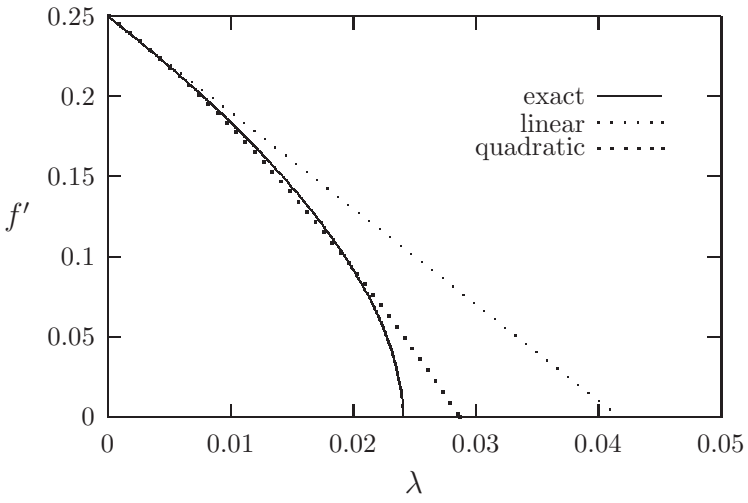


Figure 3.2: Stiffness as a function of the load parameter ($\alpha = 30°$).

Example 3.3.2. *Compute the critical points of the Mises truss using the full equilibrium system (1.23) and the augmentation*

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\phi}, \lambda) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q}, \lambda) \\ \boldsymbol{f}'(\boldsymbol{q}, \lambda)\boldsymbol{\phi} \\ \|\boldsymbol{\phi}\|^2 - 1 \end{array} \right\} = \boldsymbol{0}. \tag{3.48}$$

Consider both cases when $\alpha < 54.74°$ *and* $\alpha > 54.74°$.

The elements of the stiffness matrix $\boldsymbol{K}$ are denoted by

$$K_{11} = \frac{\partial f_1}{\partial q_1} = 2c^2 + 3q_1^2 - 2sq_2 + q_2^2, \tag{3.49}$$

$$K_{12} = \frac{\partial f_1}{\partial q_2} = 2q_1(q_2 - s) = \frac{\partial f_2}{\partial q_1} = K_{21} \tag{3.50}$$

$$K_{22} = \frac{\partial f_2}{\partial q_2} = 2s^2 + q_1^2 - 6sq_2 + 3q_2^2, \tag{3.51}$$

and the Jacobian matrix $\boldsymbol{G} = \boldsymbol{g}' = \partial \boldsymbol{g} / \partial \boldsymbol{x}$ is

$$\boldsymbol{G} = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 \\ K_{21} & K_{22} & 0 & 0 & -2 \\ Z_{11} & Z_{12} & K_{11} & K_{12} & 0 \\ Z_{21} & Z_{22} & K_{21} & K_{22} & 0 \\ 0 & 0 & 2\phi_1 & 2\phi_2 & 0 \end{bmatrix} \tag{3.52}$$

where

$$Z_{11} = 6q_1\phi_1 + 2(q_2 - s)\phi_2 \tag{3.53}$$
$$Z_{12} = 2(q_2 - s)\phi_1 + 2q_1\phi_2 = Z_{21} \tag{3.54}$$
$$Z_{22} = 2q_1\phi_1 + 6(q_2 - s)\phi_2 \tag{3.55}$$

Notice, that the Jacobian matrix is singular. if the initial starting vector $\boldsymbol{x}^0$ is a zero vector.

If $\alpha = 30°$ and using an initial guess $\boldsymbol{x} = (0, 0, 0, 0.5, 0)^T$ gives the solution

$$\boldsymbol{x} = \left\{ \begin{array}{c} q_1 \\ q_2 \\ \phi_1 \\ \phi_2 \\ \lambda \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0.2113 \\ 0 \\ 1 \\ 0.024056 \end{array} \right\} \tag{3.56}$$

Six iterations are needed to reduce the residual smaller than 10^{-8} using the criteria $\|\delta\boldsymbol{x}\|_2 < TOL\|\Delta\boldsymbol{x}\|_2$.

At the exact solution point the terms $K_{22}, K_{12} = K_{21}, Z_{12} = Z_{21}$ are zero and the Jacobian has the form

$$\boldsymbol{G}(\boldsymbol{x}_{\text{exact}}) = \left[\begin{array}{ccccc} K_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 \\ Z_{11} & 0 & K_{11} & 0 & 0 \\ 0 & Z_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{array} \right]. \tag{3.57}$$

It is clearly nonsingular since $\det(\boldsymbol{G}) = -4K_{11}^2 Z_{22} \neq 0$.

If $\alpha = 70°$, the bifurcation point is the first critical point to be found on the equilibrium path. Using the initial guess $\boldsymbol{x} = (0, 0, 0.5, 0, 0)^T$ gives the solution

$$\boldsymbol{x} = \left\{ \begin{array}{c} q_1 \\ q_2 \\ \phi_1 \\ \phi_2 \\ \lambda \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0.134046 \\ 1 \\ 0 \\ 0.094243 \end{array} \right\} \tag{3.58}$$

Five iterations are required with the same tolerance.

At the exact solution point the terms $K_{11}, K_{12} = K_{21}, Z_{11}$ and Z_{22} in the Jacobian are zero.

$$\boldsymbol{G}(\boldsymbol{x}_{\text{exact}}) = \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & K_{22} & 0 & 0 & -2 \\ 0 & Z_{12} & 0 & 0 & 0 \\ Z_{21} & 0 & 0 & K_{22} & 0 \\ 0 & 0 & 2 & 0 & 0 \end{array} \right]. \tag{3.59}$$

Clearly this matrix is singular.

Exercises

1. In practical computations critical points with positive load factor are of interest. Extended systems, like (3.3) or (3.48) can in principle converge to a critical point where the load parameter λ is negative. To avoid such situations, investigate the convergence of Newton's method to the following extended systems for the Mises truss example:

   (a)
   $$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\phi}, \lambda) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q}, \lambda) \\ \boldsymbol{f}'(\boldsymbol{q}, \lambda)\boldsymbol{\phi} \\ \|\boldsymbol{\phi}\|^2 - \lambda \end{array} \right\} = \boldsymbol{0}. \tag{3.60}$$

   (b)
   $$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\phi}, \lambda) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q}, \lambda) \\ \boldsymbol{f}'(\boldsymbol{q}, \lambda)\boldsymbol{\phi} \\ \lambda\|\boldsymbol{\phi}\|^2 - 1 \end{array} \right\} = \boldsymbol{0}. \tag{3.61}$$

   Use different starting values for the eigenvector $\boldsymbol{\phi}_0$ and use $\boldsymbol{q}_0 = \boldsymbol{0}$ and $\lambda_0 = 0$. Make conclusions of the applicability of such extensions to large systems for realistic practical computations.

2. Investigate the behaviour of Broyden's quasi-Newton method for the extended systems.

3. Investigate the behaviour of Newton's method to the extended system
   $$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{q}, \lambda) = \left\{ \begin{array}{c} \boldsymbol{f}(\boldsymbol{q}, \lambda) \\ \det(\boldsymbol{f}'(\boldsymbol{q}, \lambda)) \end{array} \right\} = \boldsymbol{0}. \tag{3.62}$$
   for the Mises truss example.

4. For large systems, the Jacobian of the extended system (3.62) has the form
   $$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \left[\begin{array}{cc} \boldsymbol{K} & -\boldsymbol{p}_{\mathrm{r}} \\ \boldsymbol{d}^T & \beta \end{array} \right] \tag{3.63}$$
   where $\boldsymbol{d}^T = \partial \det(\boldsymbol{K})/\partial \boldsymbol{q}$ and $\beta = \partial \det(\boldsymbol{K})/\partial\lambda$. These derivatives are difficult to obtain in practical large scale computations. Investigate the possibility to replace them by some properly chosen quantities and keep them constant during the iteration.

Chapter 4

Asymptotic approach

4.1 Introduction

Stability is a classical subject in structural mechanics. The history of the early days of structural stability analysis encompasses many of the great names in mechanics. Koiter lists the following names in the introduction of his revolutionary thesis [105]: Leonhard Euler, William Thomson, G.H. Bryan, R.V. Southwell, C.B. Biezeno, H. Hencky, E. Reissner, E. Trefftz, K. Marguerre, R. Kappus and Maurice Biot. Those early considerations were mainly restricted to the investigation of neutral equilibrium and directed towards the determination of stability limit. Phenomena that appear on reaching or even exceeding the stability limit were not considered.[1]

A general theory of post-buckling phenomena of elastic structures was developed by Warner Tjardus Koiter (1914 – 1997) during the second world war and culminated in his doctoral thesis on November 14, 1945 [105]. Unfortunately Koiter's work remained relatively unknown for a period of over two decades, until the English translation was published in 1967. During that period a similar theory of stability was developed by Sewell [164], Thompson and Hunt [181]. In contrast to Koiter's continuum formulation the British school of post-buckling theorists used the language of finite dimensional systems. In the works of Budiansky [37] and Hutchinson [91] variations of Koiter's energy formulation have usually been based on continuum concepts using the principle of virtual work.

Koiter's approach is asymptotic in nature, therefore called the initial post-buckling theory, and relies on perturbation methods. It gives qualitative answers on the type of post-buckling behaviour but its quantitative results are limited to the neighbourhood of the critical state. Actually it is an application of the Liapunov-Schmidt reduction to elasticity equations.

In perturbation methods the difficulty of solving non-linear equilibrium equations is avoided by solving a sequence of linear problems. However, they have gained little foot-

[1] Here we mean the general theory of stability. Post-buckling behaviour of specific structures, e.g. rods was considered by Leonhard Euler as early as 1744.

ing in computerized buckling analyses. One of the first attempts to implement Koiter's asymptotic initial post-buckling theory was the work by Haftka, Mallett and Nachbar [83]. However, their attempt was somewhat unorthodox, focusing only to the snap-through instability. Non-linearities in the pre-buckling state were considered as generalized initial imperfections of the perfect structure. Later, implementations which are more faithful to the original theory were presented, e.g. [66, 113, 139, 160].

The main stream of computer analyses of non-linear structures uses the incremental approach. It allows the handling of fully non-linear equilibrium equations without any restrictions to the kinematics. Therefore the problem of assessing the validity of the asymptotic approach is overcome. However, it is not easy to locate the singularities and to switch onto the post-buckling branches in a reliable, robust way. In addition, the literature deals mainly with simple critical points.

In the incremental approach a natural choice for the control parameter in structural mechanics is the load intensity. However, in many cases additional information of perturbations on the response of the system are of extreme importance; this is especially true near critical points. Extending the parameter space with specific perturbations of geometry, material characteristics or loading conditions provide a more complete picture of the system behaviour [63], [149].

It is evident that both of these methods, perturbation and continuation, have their pros and cons. Thus, some kind of synthesis would be welcome. Two quotations which are appropriate at this point are due to Potier-Ferry [142]: "*The most typical feature of instability theory is that its fundamental characteristics can be found in very simple models. Moreover, any complicated structural system is equivalent in some sense to one of these simple models, at least in the neighbourhood of a critical state*"; and due to Seydel [169]: "*The analysis of non-linear phenomena requires, on the other hand, tools that provide quantitative results* (continuation method) *and, on the other hand, the theoretical knowledge* (perturbation method) *of nonlinear behaviour that allows one to interpret these quantitative results*".

4.2 Liapunov-Schmidt reduction

In the Liapunov-Schmidt or Liapunov-Schmidt-Koiter reduction procedure the large non-linear system of equations (dimension N) is reduced into a locally equivalent system of non-linear equations the dimension of which is much smaller than the original one. Originally Koiter's initial postbuckling theory is a reduction from the infinite dimensional continuous problem into a small system of polynomial equations. Usually, as also in Koiter's thesis [105], the number of "post-buckling equilibrium equations" derived from the reduced potential energy expression equals the multiplicity of the buckling load. The early analytical investigations concentrated predominantly on the interaction between the local and the overall buckling of compressed structural members; consequently, the number of discrete equilibrium equations in most cases was two [141]. However, especially in com-

pressed shell structures many critical loads are involved. Koiter suggested a method to handle nearly coincident critical loads, while Byskov and Hutchinson presented a formulation for well separated critical loads [39]. It has also been shown experimentally that the interaction between well separated critical loads can occur [124]. Asymptotic analysis has been used to solve the initial post-buckling response for various structures in e.g. refs. [40], [113].

In the following the generalized Liapunov-Schmidt-Koiter (LSK) technique is briefly presented following the lines of refs. [80], [185], and [139]. Huang and Atluri [87] have used a similar technique for simple critical points. A key point in the LSK-reduction technique is the decomposition of the ambient space into summands related to the tangent operator at the critical point [80]. The residual $\boldsymbol{f}$ is a non-linear mapping from $\mathbb{X} \times \mathbb{R}$ to $\mathbb{Y}$ and the following notations for the decompositions are used in the sequel:

$$\mathbb{X} = \mathcal{N} \oplus \mathcal{N}^{\perp}, \tag{4.1a}$$

$$\mathbb{Y} = \mathcal{M} \oplus \mathcal{M}^{\perp}, \tag{4.1b}$$

where both spaces $\mathbb{X}$ and $\mathbb{Y}$ are equal to $\mathbb{R}^N$.[2] In the classical formulation, $\mathcal{N} = \mathrm{ker}\boldsymbol{K}, \mathcal{N}^{\perp} = \mathrm{range}\boldsymbol{K}^T, \mathcal{M} = \mathrm{ker}\boldsymbol{K}^T, \mathcal{M}^{\perp} = \mathrm{range}\boldsymbol{K}$. However, since the mode interaction problems are of interest, the generalized LSK-formulation [93] is adopted, where the $\mathcal{N}$-space is enlarged from only being the nullspace of the tangent operator. Thus, it is assumed that:

$$\mathrm{ker}\boldsymbol{K} \subset \mathcal{N}, \quad \mathrm{dim}(\mathrm{ker}\boldsymbol{K}) = L \leq \mathrm{dim}\mathcal{N} = M. \tag{4.2}$$

The original equilibrium equations (1.1) can thus be expanded to an equivalent pair of equations

$$\boldsymbol{P}\boldsymbol{f}(\boldsymbol{q}, \lambda) = \boldsymbol{0}, \tag{4.3a}$$

$$(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{f}(\boldsymbol{q}, \lambda) = \boldsymbol{0}, \tag{4.3b}$$

where $\boldsymbol{P}$ is a projector from $\mathbb{Y} \rightarrow \mathcal{M}^{\perp}$, with $\mathrm{ker}\boldsymbol{P} = \mathcal{M}$. Analogously, $\boldsymbol{I} - \boldsymbol{P}$ is a projector from $\mathbb{Y} \rightarrow \mathcal{M}$ with $\mathrm{ker}(\boldsymbol{I} - \boldsymbol{P}) = \mathcal{M}^{\perp}$. Expression for the projector can be written as $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{\Psi}\boldsymbol{\Psi}^T$, where $\boldsymbol{\Psi}$ is a matrix containing the basevectors of $\mathcal{M}$, i.e. $\boldsymbol{\Psi} = \left[\begin{array}{ccc} \boldsymbol{\psi}_1 & \cdots & \boldsymbol{\psi}_M \end{array} \right]$.

In view of the decomposition (4.1a), the displacement vector onto the post-bifurcation regime can be written in the form[3]

$$\boldsymbol{q} = \boldsymbol{q}_{\mathrm{cr}} + a_i \boldsymbol{\phi}_i + \boldsymbol{v}(a_i, \lambda), \tag{4.4}$$

where $\boldsymbol{\phi}_i$'s denote the base vectors spanning the space $\mathcal{N}$ and a_i's are the unknown amplitudes. The unknown vector $\boldsymbol{v}$ is required to be orthogonal to the vectors $\boldsymbol{\phi}_i$, i.e. $\boldsymbol{v} \in \mathcal{N}^{\perp}$.

[2] The spaces $\mathbb{X}$ and $\mathbb{Y}$ can be regarded as displacement and load spaces, respectively.

[3] Einstein's summation convention is adopted for repeated lower case indexes.

Since the matrix $\boldsymbol{K}_{cr}$ is invertible from $\mathcal{N}^{\perp} \rightarrow \mathcal{M}^{\perp}$, solution for $\boldsymbol{v}$ is unique near the critical point. Substituting the solution $\boldsymbol{v}$ into (4.3b) the reduced set of equilibrium equations is obtained

$$\begin{aligned} \boldsymbol{g}(a_i,\lambda) &= \boldsymbol{\Psi}^T(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{f}(\boldsymbol{q}_{\rm cr}+a_i\boldsymbol{\phi}_i+\boldsymbol{v}(a_i,\lambda),\lambda) \\ &= \boldsymbol{\Psi}^T\boldsymbol{f}(\boldsymbol{q}_{\rm cr}+a_i\boldsymbol{\phi}_i+\boldsymbol{v}(a_i,\lambda),\lambda). \end{aligned} \tag{4.5}$$

The Taylor series expansion of the vector $\boldsymbol{v}$ in the orthogonal complement of $\mathcal{N}$ is:

$$\boldsymbol{v}(a_i,\lambda) = a_i\boldsymbol{v}_i + \Delta\lambda\boldsymbol{v}_\lambda + \tfrac{1}{2}\left(a_ia_j\boldsymbol{v}_{ij} + 2\Delta\lambda a_i\boldsymbol{v}_{i\lambda} + (\Delta\lambda)^2\boldsymbol{v}_{\lambda\lambda}\right) + \cdots, \tag{4.6}$$

($\Delta\lambda = \lambda - \lambda_{\rm cr}$) and substituting it into the Taylor's series expansion of (4.3a) about the critical point, i.e. $a_i = 0, \lambda = \lambda_{\rm cr}$ gives the equations for the solution of the higher order terms. From the expansion, it can be concluded that $\boldsymbol{v}_i$ and $\boldsymbol{v}_\lambda, \boldsymbol{v}_{\lambda\lambda}$... will vanish. The only remaining displacement fields, up to second order, can be solved from equations:

$$-\boldsymbol{P}\boldsymbol{f}'\boldsymbol{v}_{ij} = \boldsymbol{P}\boldsymbol{f}''\boldsymbol{\phi}_i\boldsymbol{\phi}_j, \tag{4.7a}$$

$$-\boldsymbol{P}\boldsymbol{f}'\boldsymbol{v}_{i\lambda} = \boldsymbol{P}\boldsymbol{f}'_{,\lambda}\boldsymbol{\phi}_i, \tag{4.7b}$$

where the notation $\boldsymbol{f}' = \partial\boldsymbol{f}/\partial\boldsymbol{q}$ has been used.

Expansion of the reduced equilibrium equations at the critical state ($a_i = 0, \lambda = \lambda_{\rm cr}$) is:

$$\begin{aligned} \boldsymbol{g}(a_i,\lambda) = \boldsymbol{g}_{cr} + \boldsymbol{g}_{cr,i}a_i + \boldsymbol{g}_{{\rm cr},\lambda}\Delta\lambda + \tfrac{1}{2}\left(\boldsymbol{g}_{{\rm cr},ij}a_ia_j + 2\boldsymbol{g}_{{\rm cr},i\lambda}a_i\Delta\lambda + \boldsymbol{g}_{,\lambda\lambda}(\Delta\lambda)^2\right) \\ + \tfrac{1}{6}\left(\boldsymbol{g}_{{\rm cr},ijk}a_ia_ja_k + 3\boldsymbol{g}_{{\rm cr},ij\lambda}a_ia_j\Delta\lambda + 3\boldsymbol{g}_{{\rm cr},i\lambda\lambda}a_i(\Delta\lambda)^2 + \boldsymbol{g}_{,\lambda\lambda\lambda}(\Delta\lambda)^3\right)\ldots \end{aligned} \tag{4.8}$$

and can be written as

$$\begin{aligned} G_i\Delta\lambda + G_{ij}a_j + \tfrac{1}{2}(G_{ijk}a_ja_k + 2G_{ij\lambda}\Delta\lambda a_j) \\ + \tfrac{1}{6}(G_{ijk\ell}a_ja_ka_\ell + 3G_{ijk\lambda}\Delta\lambda a_ja_k) + \cdots = 0, \quad i = 1,\ldots,M \end{aligned} \tag{4.9}$$

where

$$\begin{aligned} G_i &= \boldsymbol{\psi}_i^T\boldsymbol{f}_{{\rm cr},\lambda}, \\ G_{ij} &= \boldsymbol{\psi}_i^T\boldsymbol{f}'_{\rm cr}\boldsymbol{\phi}_j, \\ G_{ijk} &= \boldsymbol{\psi}_i^T(\boldsymbol{f}'_{\rm cr}\boldsymbol{v}_{jk} + \boldsymbol{f}''_{\rm cr}\boldsymbol{\phi}_j\boldsymbol{\phi}_k), \\ G_{ij\lambda} &= \boldsymbol{\psi}_i^T\boldsymbol{f}'_{{\rm cr},\lambda}\boldsymbol{\phi}_j, \\ G_{ijk\ell} &= \boldsymbol{\psi}_i^T\left[\boldsymbol{f}'_{\rm cr}\boldsymbol{v}_{jk\ell} + \boldsymbol{f}''_{\rm cr}(\boldsymbol{\phi}_j\boldsymbol{v}_{k\ell} + \boldsymbol{\phi}_k\boldsymbol{v}_{j\ell} + \boldsymbol{\phi}_\ell\boldsymbol{v}_{jk}) + \boldsymbol{f}'''_{\rm cr}\boldsymbol{\phi}_j\boldsymbol{\phi}_k\boldsymbol{\phi}_\ell\right], \\ G_{ijk\lambda} &= \boldsymbol{\psi}_i^T\left[\boldsymbol{f}'_{\rm cr}\boldsymbol{v}_{jk\lambda} + \boldsymbol{f}'_{,\lambda}\boldsymbol{v}_{jk} + \boldsymbol{f}''_{\rm cr}(\boldsymbol{\phi}_j\boldsymbol{v}_{k\lambda} + \boldsymbol{\phi}_k\boldsymbol{v}_{j\lambda}) + \boldsymbol{f}'''_{{\rm cr},\lambda}\boldsymbol{\phi}_j\boldsymbol{\phi}_k\right]. \end{aligned}$$

Some remarks are in order. If the space $\mathcal{N}$ equals the nullspace of the tangent matrix, then all the components G_i and G_{ij} vanish. However, if $\dim\ker\boldsymbol{K}_{\rm cr} = L < \dim\mathcal{N} = M$,

then the product $G_{ij}a_j$ is necessarily zero in the vicinity of the critical point, since the branch directions are to be found from components $a_i, i = 1, \ldots, L$, thus $a_i = 0$, $i = L+1, \ldots, M$ and $G_{ij} \equiv 0$, when $i, j = 1, \ldots, L$.

As a summary the branch switching algorithm based on the LSK-reduction technique consists of the following steps:

1. Computation of the critical point, which can be done in many ways, either with standard incremental approach or in the case of simple bifurcation directly using an augmented system.

2. Solution of the eigenvalue problem in order to get the relevant eigenmodes. A standard eigenvalue problem is solved using the tangent stiffness matrix from the step nearest to the estimated bifurcation point.

3. Solution of the second-order (or higher) displacement fields. This is necessary only when the bifurcation is symmetric. However, it can be beneficial to compute it in any case.

4. Computation of the coefficients of the asymptotic expansion.

5. Solution of the reduced set of polynomial equilibrium equations.

6. Construction of the predictor of the bifurcating branches based on the solutions of the reduced system.

Since the dimension of the reduced problem is very small, any robust solution scheme can be applied. Notice that these equations are polynomial, hence, it is possible to find all the solutions with complex valued polynomial continuation algorithms described in ref. [125]. Alternatively, the multiresultant approach can be used to compute all real solutions of the polynomial system [8]. In the case of nearly simultaneous buckling loads, the system can be divided into two simmpler ones, see ref. [160].

Solving the amplitude equation in the vicinity of the critical point gives the local form of the equilibrium surface of the structure. The most severe limitation is that the range of validity of the results obtained is difficult to judge. Therefore, the perturbation method has primarily been considered as an “analytical tool” to get qualitative picture of the behaviour of the initial post-buckling regime.

The number of emanating branches can be large, see equations (5.3) and (5.4), therefore, for practical reasons the LSK-method is feasible when the number of interacting modes is of order ten, at maximum. However, solution of all the roots of the reduced polynomial equations can be done in parallel, if such a computer is available.

Another problem in the initial post-buckling method is to decide how many eigenmodes are relevant in the expansion. If one interacting mode is left out from the expansion, numerical computations show that it will appear in the second order field [107]. However, no mathematical proof is available. The range of validity of the asymptotic approach can

be extremely small in those cases. An example of this is given in ref. [107] where a T-beam is analysed. The interacting buckling modes comprise two local and one overall mode, the critical load of which is higher than the loads corresponding the local modes. If the overall mode is left out from the series expansion, the resulting two mode analysis deviates rapidly from the three mode analysis after the secondary bifurcation point, which lies in the immediate vicinity of the primary bifurcation point.

Example 4.2.1. *Perform the LSK reduction for the Mises truss at the bifurcation point. To have a bifurcation along the equilibrium path defined by equations (1.23) the angle* α *should satisfy* $\alpha > 54.74°$.

Chapter 5

Branch switching algorithms

5.1 Introduction

As explaned in section 2.1 continuation algorithms can be used to overcome limit (turning) points, but bifurcation points where two or more equilibrium paths cross each other, need special treatment To distinguish limit and bifurcation points a simple criteria

$$\begin{cases} \boldsymbol{\phi}^T \boldsymbol{p}_{\mathrm{r}} = 0 & \text{bifurcation point} \\ \boldsymbol{\phi}^T \boldsymbol{p}_{\mathrm{r}} \neq 0 & \text{limit point} \end{cases} \tag{5.1}$$

can be used, where $\boldsymbol{\phi}$ is the eigenvector corresponding to the singular value of the tangent stiffness matrix $\boldsymbol{K}$. The difficulty in criteria (5.1) is that the decision is usually made based on inadequate data; i.e. the equilibrium point in question can be at some finite distance drom the critical state.

When traversing the equilibrium path, a fundamental task is to monitor some criticality indicators, or singularity test functions. Such functions can be used to predict the possible existence of a critical point in subsequent steps.

5.2 Estimation of the critical point

Computation of the critical point along an equilibrium path can be done either by monitoring the evolution of certain singularity indicators or test functions during the incremental procedure or by evoking the directly solution scheme the non-linear eigenvalue problem [1, 68, 90, 99, 112, 167, 196, 197]. Such a procedure can also be started at some specific point on an equilibrium path, however, there should be some indication on the existence of a possible critical point.

To predict the existence of singular points several test functions are plausible if the linear equation is solved with direct methods. The most common ones are the following:

Determinant of the tangent stiffness matrix at an equilibrium point is a simple byproduct of the factorization step. It is unreliable if it is used without modifications. It cannot separate stable and unstable solution paths if the change in the number of negative eigenvalues is an even number. Since it is a product of all eigenvalues, the rate of change can be unrealistically high and cause uncertainty in the prediction of the critical point [108]. If it is used as a scaled quantity, the robustness as a test function is increased. However, the number of negative diagonal entries has to be monitored simultaneously. It is impossible to use with iterative linear equation solvers. A proper scaling can be defined as

$$\mathrm{sdet}(\boldsymbol{K}) = \prod_{i=1}^{N} |d_{ii}|^{1/N^{\gamma}}$$

where d_{ii} are the diagonals of the root free Cholesky decomposition $\boldsymbol{K} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$, γ is a parameter $\gamma \in (0, 1)$, which should reflect the proportion of the average rate of change in the eigenvalues. The value $\gamma = 0$ is mostly used in the technical literature, however, it will result in high variations in the values of the test function, especially for large FE models. If *a priori* knowledge is available, the proper choice of γ will improve the predictive quality of the determinant based singularity test function.

The determinant based singularity test function can be defined as a relative quantity

$$\mathrm{dbstf} = \mathrm{chsign}(\boldsymbol{K}) \frac{\mathrm{sdet}(\boldsymbol{K}_n)}{\mathrm{sdet}(\boldsymbol{K}_1)}, \tag{5.2}$$

where the subscript refers to the increment. The "change in signature"- function (chsign) is defined to be ± 1, and changes sign when a change in the signature of the stiffness matrix occurs along the path.

Current stiffness parameter (CSP) [25] can only be used with limit point singularities. Easiness of evaluation is the main advantage of the CSP and it can be also used with iterative linear equation solvers. When the tangent to the equilibrium path is parallel to the load axis the CSP goes to infinity, which makes the design of a step-length control algorithm based on CSP somewhat difficult.

The smallest eigenvalue (in absolute value) is perhaps the most reliable singularity test function. If the decomposition of the tangent stiffness is available, the inverse iteration can be easily used to evaluate the nearest to zero eigenvalue. However, the inverse iteration is not fully robust, since the convergence is obtained only if the lowest eigenvalue is single. In addition, the convergence is towards the lowest eigenvalue in absolute value, therefore special care has to be paid on unstable equilibrium paths to avoid convergence towards the smallest positive eigenvalue. Since the eigenvalue and its associated eigenvector from previous equilibrium configuration provides good starting values for the iteration, in practice only two or three inverse iterations are required for convergence.

Computation of the lowest eigenvalue is easy if an unpreconditioned iterative linear solver is used. Supression of the preconditioning step for some cycles in the PCG-iteration facilitates computation of the extreme eigenvalues of the tangent stiffness matrix from the triadiagonal matrix

$$\boldsymbol{T}_m = \text{tridiag}\left[\eta_i, \delta_i, \eta_{i+1}\right],$$

associated with the m-th step of the Lanczos iteration. Expressions for the coefficients η_i, δ_i are easily obtained from the CG algorithm, see [155]. Another strategy would be to combine the inverse iteration with Rayleigh quotient iteration [180].

Unfortunately the signature of the matrix is not easily available if iterative linear solvers are used.

The smallest pivot (in absolute value) can only be used with the direct linear equation solvers. Easy to compute.

As can be seen from the above list, none of these are good for the estimation of bifurcation points if iterative linear algebra is used.

5.2.1 Number of bifurcating branches

An essential feature for construction of a reliable bifurcation procedure is the determination of the number of possible solutions branches emanating from the critical point. This problem has been explored in the late 60's by Sewell [165, 166], Johns and Chilver [95, 94]. Depending on the symmetry properties of the system, the maximum number of different post-buckling branches is

$$2^M - 1 \tag{5.3}$$

for a system without symmetry, and

$$\tfrac{1}{2}(3^M - 1) \tag{5.4}$$

when the system is perfectly symmetric. The minimum number of post-buckling paths is 1 for the former case and M for the latter. The complexity of a multi-mode buckling problem grows enormously with the multiplicity of the critical point. Unfortunately, there exist no simple rules for the number of real branches. However, using complex polynomial continuation methods in connection with the Liapunov-Schmidt-Koiter reduction technique, all branches can, in principle, be found.

5.3 Critical points

5.3.1 Characterization and algorithmic requirements

Continuation methods characterized by the augmented equation system (2.3) are especially designed to handle the simplest case of critical points, i.e. the limit point. Since the

Jacobian of the augmented system remains regular at limit points, the implicit function theorem guarantees locally the uniqueness of the solution. This is not the case in other singular states. Bifurcations, i.e. points where two or more equilibrium paths intersect, can also emerge from the equilibrium path [179]. An algorithm capable to handle limit and bifurcation points should include the following procedures:

1. estimation and detection of a critical state [28], [59], [126], [151], [168], [172], [173], [175], [197],

2. reliable and cost-effective distinction between limit and bifurcation points [151], [153], [194],

3. branching capability on secondary paths in the case of bifurcation [28], [147], [151], [152] [183], [192], [194],

4. verification of the existence of all possible solution branches

5.3.2 Some existing branching procedures

In this section a short review of existing branch switching techniques for multiple bifurcations is given. The objective of these algorithms is to seek solutions for the load parameter λ and the projections a_i of the tangent vectors of the branches onto the critical eigenmodes $\boldsymbol{\phi}_i, i = 1, ..., M$.

Rheinboldt [147] developed an elegant and computationally favourable branch switching algorithm for simple bifurcation. He also described a generalization of his method to multiple bifurcation. However, the question of initial values for the projections a_i remained unanswered. In ref. [108] a variant of Rheinboldt's algorithm is proposed.

Keller [100] presented four algorithms, which are denoted methods I-IV. The method I uses a perturbation approach and the solutions for the branch directions are obtained from the algebraic bifurcation equation (ABE), see also ref. [103]. In the evaluation of the coefficients in ABE, second derivatives of the residual vector $\boldsymbol{f}$ are needed, or they have to be approximated by finite differences. This method will fail when ABE is degenerate, e.g. at symmetric bifurcations. In order to avoid the determination of coefficients of ABE, Keller proposed method II where the idea is to seek solutions on some subset parallel to the tangent but displaced from the bifurcation point in some direction normal to the tangent. Obviously this method will work well in simple bifurcations, but the problem with multiple bifurcation is how to parametrize in a reasonable way the subset where the solution is to be found. Remaining two methods III and IV seem to be the most robust and also computationally the most demanding. However, they are described in ref. [100] only in the case of simple bifurcation.

Kearfott [97] developed a technique where, in principle, all solution arcs can be found by locating the minima of $\|\boldsymbol{f}\|$ in the region near the critical point spanned by the critical eigenvectors, i.e finding the solution branches on a sphere centered to the estimate of

the critical point. A drawback of this method is that it requires numerous evaluations of the residual $\boldsymbol{f}$. Determination of the necessary resolution needed to find all solutions is an open question. If the resolution to scan over the sphere is too low, the probability of missing some branches increases, on the other hand, tightening the resolution increases the computational cost. Huitfeldt [89] included also the tangent vector of the primary path in the definition of the sphere where the minimization takes place. Pajunen [132] has used the residual minimization technique to solve double bifurcation problem of a truss structure.

Allgower and Chien [4] used the local perturbation method introduced by Georg [75] to multiple bifurcation problems. The idea is to introduce a perturbation near the bifurcation point and solve the perturbed problem

$$\boldsymbol{f}(\boldsymbol{q},\lambda)+\tau\boldsymbol{b}=\boldsymbol{0} \tag{5.5}$$

from a point on the primary path and traverse a perturbed path until it is near a point on a branch. The theoretical foundation of this method is based on a version of a generalized Sard's theorem. For successful branching the choice of the perturbation vectors plays a key role. In their numerical examples the components in the perturbation vectors are chosen in such a way that they oscillate correspondingly to those of the bifurcating solutions. This means that one should have *a priori* knowlegde of the solution of the problem which has to be solved. No specific theory or rules for the selection of the perturbation vectors was given in ref. [4], and the method seems to be used best as computing the solution curves interactively by trial and error fashion.

A major improvement to the local perturbation algorithm is given by Huitfeldt [89]. He introduced an auxiliary equation which defines with the perturbed equilibrium equations (5.5) a closed one dimensional curve in $N+2$-dimensional space. This curve passes exactly through one point on each branch (or half branch) determined by the unperturbed equation (1.1). When passing such a point the perturbation parameter τ changes sign. The problem is then to locate the zero points of perturbation parameter τ while traversing the branch connecting curve (BCC). Thus the branch switching problem is reduced to a path following task of the augmented system

$$\boldsymbol{h}(\boldsymbol{q},\lambda,\tau)=\left\{\begin{array}{rcl}\boldsymbol{f}(\boldsymbol{q},\lambda)+\tau\boldsymbol{b} &=& \boldsymbol{0}\\ c_{\mathrm{b}}(\boldsymbol{q},\lambda,\tau) &=& 0\end{array}\right. , \tag{5.6}$$

which can be solved with standard continuation algorithms. A constraint that defines a closed surface around the critical point is of spherical (elliptical) form:

$$c_{\mathrm{b}}(\boldsymbol{q},\lambda,\tau)=\tfrac{1}{2}\left(\|\boldsymbol{q}-\boldsymbol{q}_{\mathrm{cr}}\|_W^2+\alpha^2(\lambda-\lambda_{\mathrm{cr}})^2+\beta^2\tau^2-\rho^2\right), \tag{5.7}$$

where α,β are scaling factors and ρ is the radius of the sphere. In principle this method does not require expensive evaluation of the basis of the nullspace of the tangent stiffness matrix. Huitfeldt [89] used a random vector as perturbation $\boldsymbol{b}$.

There are some shortcomings with this conceptually simple and elegant method. It is not known if the branch connecting equation always defines a closed curve. It is believed, as also argued by Huitfeldt, that using a constraint defining a closed surface guarantees a closed path defined by the branch connecting equations (5.6), (5.7). No mathematical proof of this is known to the author. Secondly, there is no guarantee that all bifurcating branches have been found. This obviously depends on the choice of the perturbation. In addition, the computational expense can be very high for large problems, fortunately it grows only linearily with respect to the emanating branches from the bifurcation point[1]. However, the number of branches in multimode buckling with higher multiplicity can be very large as will be explained in the following.

5.3.3 Asymptotic approach

However, in the case of multimode buckling it is not easy to switch onto the post-buckling branches in a reliable, robust way. In comparison to the widely used continuation procedure, the asymptotic approach can provide some additional information such as the shape of the worst imperfection; it also enables the classification of buckling problem in terms of the catastrophe theory as described, for example, by Thompson and Hunt [182], so giving insight into the mechanism of the non-linear mode-interaction. Therefore it seems to be ideal (except the evaluation of higher order derivatives of $\boldsymbol{f}$ to combine some of the features of the asymptotic analysis to the general continuation procedure in order to handle multiple bifurcation and mode interaction problems, see [109].

[1]It is assumed that for reliable detection of the zeros of the perturbation parameter on the BCC, a minimum number of steps, say 4-5, has to separate two consecutive roots.

Chapter 6

Some linear algebra

6.1 Algebraic eigenvalue problem

6.1.1 Polynomial eigenvalue problem

The non-linear eigenvalue problem

$$(\boldsymbol{K}_0 + \lambda \boldsymbol{K}_1 + \lambda^2 \boldsymbol{K}_2 + \cdots + \lambda^r \boldsymbol{K}_r)\boldsymbol{q} = \boldsymbol{0} \tag{6.1}$$

can be transformed into a linear eigenvalue problem of r-times the size of problem (6.1) by defining: $\boldsymbol{q}_1 = \lambda \boldsymbol{q}, \boldsymbol{q}_2 = \lambda \boldsymbol{q}_1, ...$ [140], which results in an eigenvalue problem of the form

$$\left(\begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 & \cdots & \boldsymbol{K}_{r-1} & \boldsymbol{K}_r \\ -\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \right) \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_{r-1} \end{Bmatrix} = \boldsymbol{0}. \tag{6.2}$$

At first glance the formulation seems unattractive if r is large. However, solving linear equations with the first matrix of (6.2) requires only solution with $\boldsymbol{K}_0$ and the Arnoldi type iteration can be easily applied.

The quadratic form of the non-linear eigenvalue problem (3.15) is often utilized to correct the linear eigenvalue predictions. Using the transformation to a linear eigenvalue problem (using the notation $\boldsymbol{v} = \boldsymbol{q}_1$)

$$\left(\begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \right) \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{v} \end{Bmatrix} = \boldsymbol{0}. \tag{6.3}$$

This is not the only legitimate linearized version of a quadratic eigenvalue problem, other

possible forms of the problem are:

$$\left(\begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{K}_2 \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \\ \boldsymbol{K}_2 & \boldsymbol{0} \end{bmatrix} \right) \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{v} \end{Bmatrix} = \boldsymbol{0}, \tag{6.4}$$

$$\left(\begin{bmatrix} \boldsymbol{K}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_2 \end{bmatrix} + \lambda \begin{bmatrix} \boldsymbol{K}_1 & \boldsymbol{K}_2 \\ -\boldsymbol{K}_2 & \boldsymbol{0} \end{bmatrix} \right) \begin{Bmatrix} \boldsymbol{q} \\ \boldsymbol{v} \end{Bmatrix} = \boldsymbol{0}. \tag{6.5}$$

These forms are not as useful for practical computation, but they can give some insight to the problem [138], see also [184] for a review of quadratic eigenvalue problems. In the form (6.4), both matrices are symmetric, if the submatrices $\boldsymbol{K}_i$ are symmetric, however, the global matrices are indefinite. In the second form (6.5), the first matrix is positive definite if the submatrices are also positive definite, but this is achieved at the expense of symmetry in the other.

6.1.2 Linear eigenvalue problem

Introduction

There are many algorithms to solve linear algebraic eigenvalue problem. However, for efficient solution, one has to specify what to compute: (a) all eigenvalues and eigenvectors, (b) only eigenvalues, (c) only a small number of eigenvalues and (d) possibly associated eigenvectors. In structural analysis we are most often faced with eigenvalue problems where the matrices are large and sparse and only some part of the eigenvalue spectrum is of interest.

Power iteration method finds the dominant eigenvalue and the corresponding eigenvector of a give matrix $\boldsymbol{A}$. Suppose the matrix $\boldsymbol{A}$ is diagonaliizable, that is $\boldsymbol{\Phi}^{-1}\boldsymbol{A}\boldsymbol{\Phi} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ with $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$ and the eigenvalues satisfy $|\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_{N-1}| < |\lambda_N|$. Starting from a initial vector $\boldsymbol{x}_0$, such that $\|\boldsymbol{x}_0\|_2 = 1$, iterate $i = 1, 2, \ldots$

1. compute $\boldsymbol{z}_i = \boldsymbol{A}\boldsymbol{x}_{i-1}$
2. normalize $\boldsymbol{x}_i = \boldsymbol{z}_i / \|\boldsymbol{z}_i\|_2$
3. compute $\tilde{\lambda}_i = \boldsymbol{x}_i^T \boldsymbol{A} \boldsymbol{x}_i$

and if the iterate converges after k iterations, then $\tilde{\lambda}_k$ is an approximation to the largest eigenvalue λ_N and the $\boldsymbol{x}_k$ to the corresponding eigenvector $\boldsymbol{\phi}_N$. Convergence of the Power method is linear and depends on the distance between λ_{N-1} and λ_N:

$$|\lambda_N - \tilde{\lambda}_k| = \mathcal{O}\left(\left| \frac{\lambda_{N-1}}{\lambda_N} \right|^k \right). \tag{6.6}$$

This can be easily seen if we assume that the initial guess can be expressen as a linear combination of the eigenvectors

$$\boldsymbol{x}_0 = \alpha_1 \boldsymbol{\phi}_1 + \alpha_2 \boldsymbol{\phi}_2 + \cdots + \alpha_N \boldsymbol{\phi}_N \tag{6.7}$$

and assume that $\alpha_N \neq 0$. Then

$$\boldsymbol{A}\boldsymbol{x}_0 = \sum_{i=1}^{N}(\lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T) \sum_{j=1}^{N}(\alpha_j \boldsymbol{\phi}_j) = \sum_{i=1}^{N} \alpha_i \lambda_i \boldsymbol{\phi}_i \tag{6.8}$$

$$= \alpha_N \lambda_N \left(\boldsymbol{\phi}_N + \sum_{i=1}^{N-1} \frac{\alpha_i}{\alpha_N} \frac{\lambda_i}{\lambda_N} \boldsymbol{\phi}_i \right), \tag{6.9}$$

and after k iterations

$$\boldsymbol{A}^k \boldsymbol{\phi}_0 = \alpha_N \lambda_N^k \left[\boldsymbol{\phi}_N + \sum_{i=1}^{N-1} \frac{\alpha_i}{\alpha_N} \left(\frac{\lambda_i}{\lambda_N} \right)^k \boldsymbol{\phi}_i \right]. \tag{6.10}$$

Thus the algorithm converges to λ_N and $\boldsymbol{\phi}_N$, provided that the initial vector $\boldsymbol{x}_0$ has a component in the direction of the dominant eigenvector $\boldsymbol{\phi}_N$. The rate of convergence of the iteration vector is linear.

In structural stability analysis the lowest eigenvalue is usually of interest. Modifying the step 1 in the power iteration method to solve system $\boldsymbol{A}\boldsymbol{z}_i = \boldsymbol{x}_{i-1}$, gives the inverse power method which converges towards the smallest eigenvalue λ_1, provided it is simple, i.e. $|\lambda_1| < |\lambda_2|$.

The inverse iteration for a generalized eigenvalue problem

$$\boldsymbol{A}\boldsymbol{\phi} = \lambda \boldsymbol{B}\boldsymbol{\phi} \tag{6.11}$$

can be stated as:[1] starting with an initial vector $\boldsymbol{x}_0$, compute $\boldsymbol{y}_0 = \boldsymbol{B}\boldsymbol{x}_0$ and iterate $i = 1, 2, \ldots$

1. solve $\boldsymbol{A}\bar{\boldsymbol{x}}_i = \boldsymbol{y}_{i-1}$
2. compute $\bar{\boldsymbol{y}}_i = \boldsymbol{B}\bar{\boldsymbol{x}}_i$
3. compute $\rho_i = \dfrac{\bar{\boldsymbol{x}}_i^T \boldsymbol{y}_{i-1}}{\bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{y}}_i}$
4. normalize $\boldsymbol{y}_i = \bar{\boldsymbol{y}}_i / (\bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{y}}_i)^{1/2}$

then, provided that $\boldsymbol{y}_1$ has a nonzero component in the direction of the eigenmode $\boldsymbol{\phi}_1$ and the lowest eigenvalue is isolated, the iteration converges such as

$$\rho_k \approx \lambda_1 \qquad \boldsymbol{y}_k \approx \boldsymbol{B}\boldsymbol{\phi}_1. \tag{6.12}$$

Observe, that the eigenmode is now normalized wrt the matrix $\boldsymbol{B}$, i.e. $\boldsymbol{\phi}_1 \boldsymbol{B} \boldsymbol{\phi}_1 = 1$, and thus $\boldsymbol{\phi}_1 \boldsymbol{A} \boldsymbol{\phi}_1 = \lambda_1$.

[1] If the matrix $\boldsymbol{B}$ is nonsingular, then the generalized eigenvalue problem $\boldsymbol{A}\boldsymbol{\phi} = \lambda \boldsymbol{B}\boldsymbol{\phi}$ can be also written as $\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{\phi} = \lambda \boldsymbol{\phi}$.

Rayleigh quotiet iteration Shifting can improve the rate of convergence in vector iteration methods. One possibility is to use the Rayleigh quotient ρ_i (at phase 3 in the previous algorithm), calculated during the iteration process. If the Rayleigh quotient is used as a shift at every iteration, the procedure is called the Rayleigh quotient iteration and its rate of convergence is cubic if the starting vector has a big enough component of the eigenvector $\boldsymbol{\phi}_1$. The procedure can be stated as: starting with an initial vector $\boldsymbol{x}_0$, compute $\boldsymbol{y}_0 = \boldsymbol{B}\boldsymbol{x}_0$, select a starting shift ρ_0 (usually zero) and iterate $i = 1, 2, \ldots$

1. solve $(\boldsymbol{A} + \rho_{i-1}\boldsymbol{B})\bar{\boldsymbol{x}}_i = \boldsymbol{y}_{i-1}$
2. compute $\bar{\boldsymbol{y}}_i = \boldsymbol{B}\bar{\boldsymbol{x}}_i$
3. compute $\rho_i = \dfrac{\bar{\boldsymbol{x}}_i^T \boldsymbol{y}_{i-1}}{\bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{y}}_i} + \rho_{i-1}$
4. normalize $\boldsymbol{y}_i = \bar{\boldsymbol{y}}_i / (\bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{y}}_i)^{1/2}$

Notice, that the phase 1 requires the solution of a system with different coefficient matrix at every iteration. Therefore the cost of the procedure is much higher in comparison to inverse iteration if direct solvers are used. However, the situation is different in the case of iterative linear solvers, see ref. [180].

Solution of large eigenproblems

Two videly used strategies to solve this generalized eigenvalue problem in large scale finite element computations are the subspace (simultaneous) iteration and the Lanczos method [137], [18]. There is growing evidence that the Lanczos method is faster in solving the generalized eigenvalue problem, especially in the case of large clustered eigenvalue spectrum, which appear in multi-mode buckling problems. These two methods are based on the shift and invert strategy, which requires the factorization of a matrix. For very large problems this can be impossible. Van der Vorst and his co-workers have proposed Jacobi-Davidson method for polynomial eigenvalue problems [29, 177, 189]. It can be applied without inversion of matrices or transformation to the standard case. In the present notes this promising method is not considered.

Since the buckling eigenvalue problem is slightly different from the frequency analysis, a version of the subspace iteration suitable for stability analysis is briefly described. This slight difference is due to the properties of the initial stress matrix, which is indefinite in many cases.[2]

[2] In frequency analysis the corresponding matrix (mass matrix) is always positive definite or positive semidefinite.

Subspace iteration method

The generalized eigenvalue problem in question is to solve the lowest p eigenvalues and corresponding eigenvectors satisfying

$$\boldsymbol{K}_0\bar{\boldsymbol{\Phi}} = -\boldsymbol{K}_1\bar{\boldsymbol{\Phi}}\boldsymbol{\Lambda}, \tag{6.13}$$

where the diagonal matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ contains the critical eigenvalues and the matrix $\bar{\boldsymbol{\Phi}} = [\boldsymbol{q_1}, ..., \boldsymbol{q_p}]$ the corresponding eigenvectors. The subspace solution algorithm of this problem is the following [16]:

For, $k = 1, 2, ...,$ iterate:

$$\boldsymbol{K}_0\bar{\boldsymbol{\Phi}}_{k+1} = -\boldsymbol{K}_1\bar{\boldsymbol{\Phi}}_k.$$

Find the projections of operators $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$:

$$\boldsymbol{A}_{k+1} = \bar{\boldsymbol{\Phi}}_{k+1}^T\boldsymbol{K}_0\bar{\boldsymbol{\Phi}}_{k+1}, \quad \text{and} \quad \boldsymbol{B}_{k+1} = -\bar{\boldsymbol{\Phi}}_{k+1}^T\boldsymbol{K}_1\bar{\boldsymbol{\Phi}}_{k+1}.$$

Solve for the eigensystem of projected operators:

$$\boldsymbol{A}_{k+1}\boldsymbol{Q}_{k+1} = \boldsymbol{B}_{k+1}\boldsymbol{Q}_{k+1}\boldsymbol{\Lambda}_{k+1}. \tag{6.14}$$

Find an improved approximation to eigenvectors:

$$\bar{\boldsymbol{\Phi}}_{k+1} = \bar{\boldsymbol{\Phi}}_{k+1}\boldsymbol{Q}_{k+1}.$$

If the vectors in $\bar{\boldsymbol{\Phi}}_1$ are not orthogonal to one of the required eigenvectors, the algorithm converges, i.e. $\boldsymbol{\Lambda}_{k+1} \longrightarrow \boldsymbol{\Lambda}$ and $\bar{\boldsymbol{\Phi}}_{\mathbf{k+1}} \longrightarrow \bar{\boldsymbol{\Phi}}$ as $k \longrightarrow \infty$.

Since the projected matrix $\boldsymbol{B}_{k+1}$ is not necessarily positive definite, the projected generalized eigenvalue problem (6.14) is first written in the inverse form

$$\boldsymbol{B}_{k+1}\boldsymbol{Q}_{k+1} = \boldsymbol{A}_{k+1}\boldsymbol{Q}_{k+1}\boldsymbol{\Lambda}_{k+1}^{-1}. \tag{6.15}$$

Now, the projected matrix $\boldsymbol{A}_{k+1}$ is positive definite and the Cholesky decomposition $\boldsymbol{A}_{k+1} = \boldsymbol{L}\boldsymbol{L}^T$ is possible. The generalized problem (6.15) is then reduced to the standard eigenvalue problem

$$\boldsymbol{C}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda}^{-1}, \tag{6.16}$$

where $\boldsymbol{C} = \boldsymbol{L}^{-1}\boldsymbol{B}_{k+1}(\boldsymbol{L}^{-1})^T$ and $\boldsymbol{X} = \boldsymbol{L}^T\boldsymbol{Q}_{k+1}$.

Solution of the standard eigenvalue problem (6.16) can be obtained in three phases. First, the coefficient matrix $\boldsymbol{C}$ is reduced to tridiagonal form by Householder transformations. The eigenvalues of the tridiagonal matrix are obtained by the QL decomposition algorithm using implicit shifts in order to speed up the convergence and to maintain good numerical conditioning. Finally, the eigenvectors are computed using the inverse iteration.

Convergence of the iterative process can be accelerated by using a shift. However, the requirement that the projected matrix $\boldsymbol{A}$ of the shifted stiffness matrix $\boldsymbol{K}_0 + \sigma\boldsymbol{K}_1$ has to be positive definite, restricts the shift σ to satisfy

$$|\sigma| < \min|\lambda_i|.$$

Acceleration of the subspace iteration has been considered e.g. in refs. [3], [15], [144].

Lanczos method

Krylov subspaces based methods such as Lanczos and Arnoldi algorithms are widely used for treating eigenproblems with large sparse matrices. They are shown to perform better than vector iteration methods. For an overview of these methods see refs. [79], [137]. A well-known robust implementation is due to Grimes, Lewis and Simon [81] which is also incorporated in the MSC/NASTRAN structural analysis code.

The basic idea Cornelius Lanczos presented the algorithm in 1950 to compute some of the extreme eigenvalues of a given symmetric matrix $\boldsymbol{A}$. It is based on sequence of vectors like $\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}^2\boldsymbol{x}, \ldots,$, and the method generates a sequence of tridiagonal matrices $\boldsymbol{T}_j$ which have the property that the extremal eigenvalues of $\boldsymbol{T}_j \in \mathbb{R}^{j\times j}$ are progressively better estimates of the extremal eigenvalues of $\boldsymbol{A}$. Let $\boldsymbol{T} = \boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}$, and $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots \boldsymbol{q}_N]$ and

$$\boldsymbol{T} = \begin{bmatrix} \alpha_1 & \beta_1 & & \cdots & 0 \\ \beta_1 & \alpha_1 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{N-1} \\ 0 & \cdots & & \beta_{N-1} & \alpha_N \end{bmatrix}. \tag{6.17}$$

Since $\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{T}$, the orthonormal vectors $\boldsymbol{q}_j$ satisfy

$$\boldsymbol{A}\boldsymbol{q}_j = \beta_{j-1}\boldsymbol{q}_{j-1} + \alpha_j\boldsymbol{q}_j + \beta_j\boldsymbol{q}_{j+1}. \tag{6.18}$$

The entries of the symmetric tridiagonal matrix are easy to find, multiplying (6.18) by $\boldsymbol{q}_j$ gives

$$\alpha_j = \boldsymbol{q}_j^T\boldsymbol{A}\boldsymbol{q}_j. \tag{6.19}$$

The β_j term can be obtained by rewriting equation (6.18) as

$$\beta_j\boldsymbol{q}_{j+1} = \boldsymbol{A}\boldsymbol{q}_j - \alpha_j\boldsymbol{q}_j - \beta_{j-1}\boldsymbol{q}_{j-1} = \boldsymbol{r}_j. \tag{6.20}$$

The Lanczos iteration can be stated as:

1. Initialize $j = 0, \boldsymbol{q}_0 = 0, \beta_0 = 1$ select $\boldsymbol{q}_1$ such that $\|\boldsymbol{q}_1\| = 1$ and set $\boldsymbol{r}_0 = \boldsymbol{q}_1$.
2. Iterate while $\beta_j \neq 0$
   (a) $j := j + 1$
   (b) $\alpha_i = \boldsymbol{q}_j^T\boldsymbol{A}\boldsymbol{q}_j$
   (c) $\boldsymbol{r}_j = (\boldsymbol{A} - \alpha_j\boldsymbol{I})\boldsymbol{q}_j - \beta_{j-1}\boldsymbol{q}_{j-1}$
   (d) $\beta_j = \|\boldsymbol{r}_j\|$

The sequence of the Lanczos vectors $\boldsymbol{q}_j$ has two basic properties

1. each $\boldsymbol{q}_{j+1}$ is a combination of $\boldsymbol{q}_1, \boldsymbol{A}\boldsymbol{q}_1, \ldots \boldsymbol{A}^j\boldsymbol{q}_1$
2. each $\boldsymbol{q}_{j+1}$ is orthogonal to all combinations of $\boldsymbol{q}_1, \boldsymbol{A}\boldsymbol{q}_1, \ldots \boldsymbol{A}^{j-1}\boldsymbol{q}_1$

The main problem in Lanczos iteration is its numerical instability. In practice after a few steps orthogonality is lost and the vectors are not linearly independent. A complete reorthogonalization is expensive, thus selective and partial reorthogonalization strategies have been developed, see section 10.6 in [88] and [121].

Block Lanczos algorithm: The approach by blocks allows better convergence properties when there are multiple eigenvalues which is of primary importance in stability analyses of thin shells. Next, the block Lanczos algorithm as coded in the program package BLZPACK of Marques [121] will be briefly described. The BLZPACK employs a combination of modified partial reorthogonalization and selective orthogonalization strategies to preserve the orthogonality of the bases generated by the algorithm.

The eigenvalue problem (6.13) in the BLZPACK is transformed to a form

$$\boldsymbol{K}_0(\boldsymbol{K}_0 + \sigma\boldsymbol{K}_1)^{-1}\boldsymbol{K}_0\boldsymbol{\phi} = \theta\boldsymbol{K}_0\boldsymbol{\phi} \qquad \text{i.e.} \qquad \boldsymbol{A}\boldsymbol{\phi} = \theta\boldsymbol{B}\boldsymbol{\phi}, \tag{6.21}$$

where $\theta = \lambda/(\lambda - \sigma)$ and $\sigma \neq 0$ is the shift. Implementational details concerning the monitoring of the orthogonality, the spectral transformation, the spectrum slicing strategy and the data management during the generation process can be found in ref. [121] and are not repeated here.

The algorithm can be summarized as [121]: ($\boldsymbol{Q}_j$ and $\boldsymbol{R}_j$ are $N \times m$ matrices, $\hat{\boldsymbol{A}}_j$ is $m \times m$ and $\hat{\boldsymbol{B}}$ is $m \times m$ upper triangular)

1. Initialization:
   (a) set the number of vectors in a block m, the shift σ
   (b) compute $\boldsymbol{A}_\sigma = \boldsymbol{A} - \sigma\boldsymbol{B}$
   (c) set $\boldsymbol{Q}_0 = \boldsymbol{0}$, $\boldsymbol{R}_0 \neq \boldsymbol{0}$
   (d) factorize $\boldsymbol{R}_0 = \boldsymbol{Q}_1\hat{\boldsymbol{B}}_1$, such that $\boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Q}_1 = \boldsymbol{I}$
2. Lanczos steps:, iterate for $j = 1, 2, \ldots, NSTEPS$
   (a) compute $\boldsymbol{R}_j = \boldsymbol{A}_\sigma^{-1}\boldsymbol{B}\boldsymbol{Q}_j$
   (b) update $\boldsymbol{R}_j := \boldsymbol{R}_j - \boldsymbol{Q}_{j-1}\hat{\boldsymbol{B}}_j^T$
   (c) compute $\hat{\boldsymbol{A}}_j = \boldsymbol{Q}_j^T\boldsymbol{B}\boldsymbol{R}_j$
   (d) update $\boldsymbol{R}_j := \boldsymbol{R}_j - \boldsymbol{Q}_j\hat{\boldsymbol{A}}_j^T$
   (e) factorize $\boldsymbol{R}_j = \boldsymbol{Q}_{j+1}\hat{\boldsymbol{B}}_{j+1}$, such that $\boldsymbol{Q}_{j+1}^T\boldsymbol{B}\boldsymbol{Q}_{j+1} = \boldsymbol{I}$
   (f) if required ortogonalize $\boldsymbol{Q}_j$ and $\boldsymbol{Q}_{j-1}$ against the vectors in $\boldsymbol{\mathcal{Q}}_{j-1}$

(g) insert $\boldsymbol{Q}_j$ into $\boldsymbol{\mathcal{Q}}_j$ and $\hat{\boldsymbol{A}}_j, \hat{\boldsymbol{B}}_j$ into $\boldsymbol{T}_j$

(h) solve the reduced problem $\boldsymbol{T}_j \boldsymbol{s}_k = \theta_k \boldsymbol{s}_k$, $k = 1, 2, \ldots m \times j$

(i) set $\hat{\lambda}_k = \sigma + 1/\theta_k$

(j) test convergence by checking the number of eigenpairs for which $TOL \geq \|\hat{\boldsymbol{B}}_{j+1} \boldsymbol{s}_j^{(k)}\| = \|\boldsymbol{A}\boldsymbol{\phi}_k - \hat{\lambda}_k \boldsymbol{B}\boldsymbol{\phi}_k\|$, and exit if enough eigenpairs have converged

3. compute the converged eigenvectors $\boldsymbol{\phi}_k = \boldsymbol{\mathcal{Q}}_j \boldsymbol{s}_k$.

The block tridiagonal matrix $\boldsymbol{T}_j$ is

$$\boldsymbol{T}_j = \begin{bmatrix} \hat{\boldsymbol{A}}_1 & \hat{\boldsymbol{B}}_2^T & & & \\ \hat{\boldsymbol{B}}_2 & \hat{\boldsymbol{A}}_2 & \hat{\boldsymbol{B}}_3^T & & \\ & \hat{\boldsymbol{B}}_3 & \hat{\boldsymbol{A}}_3 & \ddots & \\ & & \ddots & \ddots & \hat{\boldsymbol{B}}_j^T \\ & & & \hat{\boldsymbol{B}}_j & \hat{\boldsymbol{A}}_j \end{bmatrix} \tag{6.22}$$

6.2 Solution of the linear equation system

6.2.1 Introduction

In most non-linear structural codes the solution of the linear system is performed with a direct solver. If the stiffness matrix is symmetric, the root free Cholesky or Crout $\boldsymbol{LDL}^T$ decomposition is used, while for unsymmetric matrices the $\boldsymbol{LU}$ decomposition is used. For large 3-dimensional problems the decomposition time and the storage requirements will be prohibitively high when Gaussian elimination type factorizations are used. The decomposition time dominates the overall cost of the continuation process, since the asymptotic operation count for standard decomposition is of order $E^{3-2/d}$, where E is the number of elements and d is the space dimension[3], while the time needed to compute and assemble the internal force vector and stiffness matrix is naturally linearly proportional to the number of elements. Special sparse matrix techniques have been developed which try to minimize the fill in during the decomposition. Iterative methods seem to be ideal for modern vector and parallel computers to solve systems of linear equations. For large problems they require much less storage than direct solvers and computing times are also in many cases reduced.

For a discussion of state of the art of direct solution techniques of linear systems, see ref. [55].

[3] Assuming uniform mesh with approximately same number of elements in each coordinate axis direction.

In the sequel, a generic linear equation system will be denoted by

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

where the coefficient matrix $\boldsymbol{A}$ can be symmetric or unsymmetric. An equivalent preconditioned system is

$$\boldsymbol{M}_1^{-1}\boldsymbol{A}\boldsymbol{M}_2^{-1}\boldsymbol{y} = \boldsymbol{M}_1^{-1}\boldsymbol{b}, \tag{6.23}$$

where $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2$ is the preconditioning matrix and $\boldsymbol{M}_1, \boldsymbol{M}_2$ are the left- and right-preconditioning matrices, respectively. In practice, this split form is not always needed. It is usually possible to rewrite the iterative method in a way that only one computational step: find $\boldsymbol{u}$ from $\boldsymbol{u} = \boldsymbol{M}^{-1}\boldsymbol{v}$, is necessary, so the preconditioner applies in its entirety. However, the system (6.23) gives possibility for different preconditioning strategies. It should be noted that the spectra of the three associated operators $\boldsymbol{M}^{-1}\boldsymbol{A}$, $\boldsymbol{A}\boldsymbol{M}^{-1}$ and $\boldsymbol{M}_1^{-1}\boldsymbol{A}\boldsymbol{M}_2^{-1}$ are identical. Therefore, similar convergence behaviour should be expected. However, it is well known that the eigenvalues do not always govern convergence [155]. For these preconditioning versions different residuals are available which in each case may affect the stopping criterion and may cause the algorithm to stop either prematurely or with delay. This can happen in case $\boldsymbol{M}$ is ill-conditioned.

Most of the preconditioned iterative techniques require the preconditioning to be a constant operator. However, several iterative procedures are developed in the literature that can accomodate the variations in the preconditioner. Perhaps one of the most well-known of such iterations is the flexible variant of the generalized minimum residual algorithm. These flexible variants are not considered in the present study.

Different preconditioning techniques are briefly described in the subsequent section.

6.2.2 Krylov subspace methods

Krylov subspace methods seem to be among the most important iterative techniques available for solving large linear systems [11], [155], [190]. These techniques are based on projections onto Krylov subspaces, which are subspaces spanned by vectors which are obtained recursively by multiplying the previous residual with the matrix: i.e. [4]

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}_0) = \operatorname{span}\left\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, ..., \boldsymbol{A}^{m-1}\boldsymbol{r}_0\right\},$$

where $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$. Approximate solution of the system is found from a m-dimensional subspace $\boldsymbol{x}_0 + \mathcal{K}_m$ by imposing the Petrov-Galerkin condition requiring the residual to be orthogonal to another m-dimensional subspace $\mathcal{L}_m$.

Next, the following algorithms will be presented without derivations:

- conjugate gradient
- symmetric QMR
- bi-conjugate gradient

[4] In place of $\boldsymbol{A}$ there could be e.g. $\boldsymbol{M}^{-1}\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{M}^{-1}$.

- bi-conjugate gradient stabilized

The algorithms are presented as left preconditioned versions.

The most well known Krylov subspace method is the preconditioned conjugate gradient (PCG) method for symmetric positive definite (SPD) matrices. There are many different implementations of the PCG-iteration, but the following algorithm is perhaps the most common.

Preconditioned conjugate gradient algorithm: construct $\boldsymbol{M}$ (or directly $\boldsymbol{M}^{-1}$), initialize $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, apply preconditioner $\boldsymbol{d}_0 = \boldsymbol{M}^{-1}\boldsymbol{r}_0$, compute $\tau_0 = \boldsymbol{r}_0^T\boldsymbol{d}_0$ and iterate $i = 0, 1, 2, ...$ until convergence:

1. compute: $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{d}_i, \quad \alpha_i = \tau_i / \boldsymbol{d}_i^T\boldsymbol{s}$,
2. update: $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i\boldsymbol{d}_i, \quad \boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i\boldsymbol{s}$,
3. apply preconditioner: $\boldsymbol{z} = \boldsymbol{M}^{-1}\boldsymbol{r}_{i+1}$,
4. compute $\tau_{i+1} = \boldsymbol{r}_{i+1}^T\boldsymbol{z}, \quad \beta_i = \tau_{i+1}/\tau_i$,
5. update $\boldsymbol{d}_{i+1} = \boldsymbol{z} + \beta_i\boldsymbol{d}_i$.

It is a Galerkin (orthogonal projection) type Krylov subspace method, i.e. $\mathcal{L}_m = \mathcal{K}_m$. One iterate of the PCG method requires one matrix-vector product, five[5] level-1-operations and one application of the preconditioning operation: $\boldsymbol{z} = \boldsymbol{M}^{-1}\boldsymbol{r}$. The residual norm can be evaluated after step 2 in the above algorithm, however, a cheap measure for monitoring the convergence is obtained in the weighted norm: $\sqrt{\tau} = \|\boldsymbol{r}\|_{M^{-1}} = (\boldsymbol{r}^T\boldsymbol{M}^{-1}\boldsymbol{r})^{1/2}$.

If the matrix $\boldsymbol{A}$ is symmetric but indefinite, the PCG-algorithm can become unstable and even break down. Paige and Saunders [131] were the first to devise stable algorithms for symmetric indefinite systems. These two algorithms called SYMMLQ and MINRES are based on Lanczos tridiagonalization, which exists also in indefinite case.

The drawback of the SYMMLQ and MINRES algorithms are that the preconditioner $\boldsymbol{M}$ need to be a SPD-matrix. For highly indefinite systems, this restriction seems to be rather unnatural. The symmetric QMR algorithm [73] allows the use of arbitrary symmetric nonsingular preconditioner. The QMR iterate is characterized by a quasi-minimization of the preconditioned residual norm. If the preconditioner is positive definite, then MINRES and symmetric QMR iterations are mathematically equivalent, and the residual norms are tryly minimized.

Preconditioned symmetric QMR algorithm: construct $\boldsymbol{M}$ (or directly $\boldsymbol{M}^{-1}$), initialize $\boldsymbol{s}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, apply preconditioner $\boldsymbol{q}_0 = \boldsymbol{M}^{-1}\boldsymbol{r}_0$, compute $\tau_0 = \|\boldsymbol{q}_0\|$, $\rho_0 = \boldsymbol{s}_0^T\boldsymbol{q}_0$, set $\varphi_0 = 0$, $\boldsymbol{d} = \boldsymbol{0}$ and iterate $i = 1, 2, ...$ until convergence:

1. compute: $\boldsymbol{t} = \boldsymbol{A}\boldsymbol{q}_{i-1}, \quad \sigma_{i-1} = \boldsymbol{q}_{i-1}^T\boldsymbol{t}, \quad \alpha_{i-1} = \rho_{i-1}/\sigma_{i-1}$

[5] PCG requires an additional norm evaluation if the convergence is checked from the residual $\boldsymbol{r}$.

2. update: $\boldsymbol{s}_i = \boldsymbol{s}_{i-1} - \alpha_i \boldsymbol{t}$,
3. apply preconditioner: $\boldsymbol{t} = \boldsymbol{M}^{-1}\boldsymbol{s}_i$,
4. compute $\vartheta_i = \|\boldsymbol{t}\|/\tau_{i-1}, \quad c_i = 1/\sqrt{1+\vartheta_i^2}, \quad \tau_i = \tau_{i_1}\vartheta_i c_i$
5. update $\boldsymbol{d}_i = c_i^2\vartheta_{i-1}^2\boldsymbol{d}_{i-1} + c_j^2\alpha_{i-1}\boldsymbol{q}_{i-1}$ and $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \boldsymbol{d}_i$
6. compute $\rho_i = \boldsymbol{t}^T\boldsymbol{s}_i, \qquad \beta_i = \rho_i/\rho_{i-1}$
7. update $\boldsymbol{q}_i = \beta_i\boldsymbol{q}_{i-1} + \boldsymbol{t}$.

For unsymmetric matrices the situation is much more complex. The CG method for SPD systems has two important properties. It is based on three term recurrence, and it minimizes the error with respect to the energy norm. Unfortunately these two properties can only be fulfilled for nonsymmetric CG-type schemes for a very limited class of matrices, namely the shifted and rotated Hermitean matrices.

Most of the existing iterative algorithms for solving nonsymmetric linear systems are based either on the full orthogonalization method of Arnoldi or the Lanczos biorthogonalization methods. Saad and Schultz [156] suggested the generalized minimum residual method (GMRES), which is a projection method with the choice $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}_0)$ and $\mathcal{L}_m = \boldsymbol{A}\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}_0)$. It can also be viewed as an extension of the MINRES to nonsymmetric problems. There are many possible variations of the GMRES method, see ref. [155]. The main disadvantage of the GMRES is long recurrencies and in practical computations its restarted versions are mostly used.

In this work only those algorithms are considered which retain the short recurrencies thus being more favourable with respect to memory requirements. Biconjugate gradient (Bi-CG) type algorithms are based on the Lanczos biorthogonalization algorithm which builds a pair of biorthogonal bases for the two subspaces $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{r}_0)$ and $\mathcal{L}_m(\boldsymbol{A}^T, \tilde{\boldsymbol{r}}_0)$. The Bi-CG algorithm can be implemented as follows.

Preconditioned bi-conjugate gradient algorithm: construct $\boldsymbol{M}$ (or directly $\boldsymbol{M}^{-1}$), initialize $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, choose $\tilde{\boldsymbol{r}}_0$, apply preconditioner $\boldsymbol{d}_0 = \boldsymbol{M}^{-1}\boldsymbol{r}_0$ and $\tilde{\boldsymbol{d}}_0 = \boldsymbol{M}^{-T}\tilde{\boldsymbol{r}}_0$, compute $\tau_0 = \tilde{\boldsymbol{r}}_0^T\boldsymbol{d}_0$ and iterate $i = 0, 1, 2, ...$ until convergence:

1. compute: $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{d}_i, \quad \tilde{\boldsymbol{s}} = \boldsymbol{A}^T\tilde{\boldsymbol{d}}_i, \quad \alpha_i = \tau_i/\tilde{\boldsymbol{d}}_i^T\boldsymbol{s}$,
2. update: $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i\boldsymbol{d}_i, \quad \boldsymbol{r}_{i+1} = \boldsymbol{r}_i - \alpha_i\boldsymbol{s}, \quad \tilde{\boldsymbol{r}}_{i+1} = \tilde{\boldsymbol{r}}_i - \alpha_i\tilde{\boldsymbol{s}}$,
3. apply preconditioner: $\boldsymbol{z} = \boldsymbol{M}^{-1}\boldsymbol{r}_{i+1}, \quad \tilde{\boldsymbol{z}} = \boldsymbol{M}^{-T}\tilde{\boldsymbol{r}}_{i+1}$,
4. compute $\tau_{i+1} = \tilde{\boldsymbol{r}}_{i+1}^T\boldsymbol{z}, \quad \beta_i = \tau_{i+1}/\tau_i$,
5. update $\boldsymbol{d}_{i+1} = \boldsymbol{z} + \beta_i\boldsymbol{d}_i, \quad \tilde{\boldsymbol{d}}_{i+1} = \tilde{\boldsymbol{z}} + \beta_i\tilde{\boldsymbol{d}}_i$.

The algorithm fails whenever $\tau_i = \tilde{\boldsymbol{r}}^T \boldsymbol{z} = 0$. Although such a breakdown is very unprobable in practice, near breakdowns when $\tau_i \approx 0$ are possible and cause a serious numerical stability problem.

In some applications the multiplications with $\boldsymbol{A}^T$ and preconditioning steps with $\boldsymbol{M}^T$ can be impossible to perform. Sonneveld [178] developed the biconjugate gradient squared (CGS) method which eliminated the need of transposed matrices.[6] However, since CGS is derived by squaring the polynomials associated to the residual and direction vectors, rounding errors can be more harmful than in the standard Bi-CG algorithm. Van der Vorst [188] devised a stabilized version of the CGS which is called the bi-conjugate gradient stabilized iteration, Bi-CGSTAB for short. Many modifications of the Bi-CGSTAB scheme have been proposed in the literature, see e.g. refs. [43], [176] [199]. Here, the procedure is given in the original form [188].

Preconditioned bi-conjugate gradient stabilized algorithm: construct $\boldsymbol{M}$ (or directly $\boldsymbol{M}^{-1}$), initialize $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, choose $\tilde{\boldsymbol{r}}$, compute $\rho_0 = \tilde{\boldsymbol{r}}^T \boldsymbol{r}_0$, set $\boldsymbol{d}_0 = \boldsymbol{r}_0$ and iterate $i = 0, 1, 2, ...$ until convergence:

1. apply preconditioner: $\boldsymbol{z} = \boldsymbol{M}^{-1} \boldsymbol{r}_i$,
2. compute: $\boldsymbol{v}_i = \boldsymbol{A}\boldsymbol{z}, \quad \alpha_i = \rho_i / \tilde{\boldsymbol{r}}^T \boldsymbol{v}_i, \quad \boldsymbol{s} = \boldsymbol{r}_i - \alpha_i \boldsymbol{v}_i$,
3. apply preconditioner: $\tilde{\boldsymbol{s}} = \boldsymbol{M}^{-1} \boldsymbol{s}$,
4. compute: $\boldsymbol{w} = \boldsymbol{A}\tilde{\boldsymbol{s}}, \quad \omega_i = \boldsymbol{w}^T \boldsymbol{s} / \boldsymbol{w}^T \boldsymbol{w}$,
5. update: $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \boldsymbol{z} + \omega_i \tilde{\boldsymbol{s}}, \quad \boldsymbol{r}_{i+1} = \boldsymbol{s} - \omega_i \boldsymbol{w}$,
6. compute $\rho_{i+1} = \tilde{\boldsymbol{r}}^T \boldsymbol{r}_{i+1}$ and $\beta_{i+1} = (\rho_{i+1}/\rho_i)(\alpha_i/\omega_i)$,
7. update $\boldsymbol{d}_{i+1} = \boldsymbol{r}_{i+1} + \beta_{i+1}(\boldsymbol{d}_i - \omega_i \boldsymbol{v}_i)$.

Since both coefficients ρ and ω have to be nonzero, there are three possible breakdown points in the Bi-CGSTAB method, i.e. $\tilde{\boldsymbol{r}}^T \boldsymbol{v}_i \neq 0$, $\boldsymbol{w}^T \boldsymbol{s} \neq 0$ and $\tilde{\boldsymbol{r}}^T \boldsymbol{r}_i \neq 0$. In the literature a common choice for the vector $\tilde{\boldsymbol{r}}$ is the initial residual $\boldsymbol{r}_0$. Bulgakov [38] recommends the vector $\tilde{\boldsymbol{r}} = \boldsymbol{M}^{-1} \boldsymbol{r}_0$. If the initial approximation $\boldsymbol{x}_0$ is chosen to be a random vector these two approaches perform almost identically. However, if the initial approximation is a zero vector and the load vector $\boldsymbol{b}$ consists of only few nonzero components, the choice $\tilde{\boldsymbol{r}} = \boldsymbol{r}_0$ is not recommendable. A reasonable choice seems to be $\tilde{\boldsymbol{r}} = \boldsymbol{r}_0 + \boldsymbol{a}$, where $\boldsymbol{a}$ is a random vector.

To cure the situation the look-ahead Lanczos algorithms have been developed. The drawback of look-ahead steps is the increased complexity of the algorithm. Therefore simpler remedies, like restarting the Lanczos procedure, can be adequate.

For a unified general description of these methods with numerous references see refs. [14], [155], [190].

[6] Many other transpose free modifications of the Bi-CG algorithm exist, although the CGS and the Bi-CGSTAB are perhaps the most well known, see discussion in ref. [36].

6.2.3 Preconditioning

It is well known that the performance of iterative solvers depends on the eigenvalue distribution and on the possible non-normality[7] of the coefficient matrix. These problems can be avoided, to some extent, by employing a preconditioner. It seems to be generally agreed that the choice of the preconditioner is even more critical than the choice of the type of the Krylov subspace iteration [21].

There are two major conflicting requirements in the development of a preconditioned iteration, namely, the construction[8] and use of a preconditioner should be inexpensive and its resemblance with matrix $\boldsymbol{A}$ should be as close as possible. The most general preconditioning strategies can be grouped into classes:

1. preconditioners based on classical iterations like Jacobi, SSOR,
2. incomplete sparse LU-decompositions (ILU or IC for symmetric matrices),
3. polynomial preconditioners,
4. explicit sparse approximate inverse preconditioners,
5. multigrid or multilevel preconditioners.

Incomplete factorization is perhaps the most well known strategy. There are many variants of ILU-decompositions differing, for instance on the way how the nonzero pattern of the preconditioner is defined. The simplest strategy is to have the same nonzero pattern for the $\boldsymbol{L}$ and $\boldsymbol{U}$ factors as $\boldsymbol{A}$. This incomplete factorization known as ILU(0) is easy and inexpensive to compute, but often leads to a crude approximation resulting in many iterations in the accelerator to converge. Several alternative ILU factorizations have been developed in which the fill-in is determined either by using the concept of level of fill or by a threshold strategy where the nonzero pattern of the preconditioner is determined dynamically neglecting small elements in the factorization.

Meijerik and Van der Vorst [123] proved the existence of the ILU factorization for arbitrary fill patterns when the coefficient matrix is a M-matrix[9]. This is often the case, e.g. for matrices arising from discretizations of the heat equation. However, matrices arising from problems in structural mechanics usually do not have this property. In order to circumvent this problem different strategies exist. Shifting is perhaps the most straightforward remedy, the factorization is carried out for the shifted matrix $\boldsymbol{A} + \rho\mathrm{diag}(\boldsymbol{A})$,

[7] A matrix $\boldsymbol{A}$ is said to be normal if $\boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{A}^H\boldsymbol{A}$, where the superscript H denotes the conjugate transposition. A normal matrix is the most general matrix which has a diagonal Schur form. Therefore, all its eigenvalues and eigenvectors are well-conditioned: the spectral representation is stable with respect to perturbations. The bad effect of non-normality is the possible deterioration of the numerical quality for iterative methods run in finite precision arithmetic [30], [41].

[8] If the preconditioner has to be used many times more effort could be paid to its construction.

[9] A matrix is a M-matrix if its off-diagonal elements are nonpositive and all the elements of the inverse are positive.

where ρ is a parameter. However, finding an optimal value for the shift parameter ρ is a non-trivial task. Another approach is to apply an additional reduction step where an M-matrix is determined from the stiffness matrix and the incomplete factorization scheme is applied to this matrix [158]. The incomplete factorization is then guaranteed to exist, but, unfortunately, the reduction step can produce a matrix the resemblage of which to the original matrix is not good enough.

Ajiz and Jennings [2] proposed the corrected IC factorization (CIC), [10] which guarantees a positive definite preconditioner if the matrix itself is SPD, but it often results in too large modifications to the diagonal which slows down the convergence of the accelerator iteration.

Mathematical analysis reveals that for second-order elliptic boundary value problems the ILU(0) approach is asymptotically no better than the unpreconditioned iteration. More precisely, the condition number of the ILU preconditioned operator is of the same order as that of matrix $\boldsymbol{A}$. Several variants of the basic ILU have been presented in the literature e.g. MILU, RILU and DRILU (modified, relaxed and dynamically relaxed) [158]. However, when considering real engineering problems these modified versions do not necessarily perform any better than the basic ILU.

It should be remembered that the effectiveness of a preconditioning strategy is highly problem and architecture dependent. For instance, incomplete factorizations are difficult to implement on high-performance computers, due to the sequential nature of the triangular solves. On the other hand, sparse approximate inverse preconditioning required only matrix-vector products, which are relatively easy to vectorize and parallelize, but they are usually not as robust as ILU-factorization based strategies [21].

For second-order elliptic PDE's discretized by low order finite elements many of the listed preconditioning techniques can be used. However, for finite element models of thin-shells only the incomplete factorization allowing some degree of fill-in [19] or a multilevel preconditioner [38], [69], [187] seems to be the only reasonable choices.

For a certain type of a preconditioning technique, the computational complexity can be reduced. Construction of a preconditioning matrix $\boldsymbol{M}$ in a form

$$\boldsymbol{M} = (\tilde{\boldsymbol{D}} + \boldsymbol{E})\hat{\boldsymbol{D}}(\tilde{\boldsymbol{D}} + \boldsymbol{F}), \tag{6.24}$$

where $\tilde{\boldsymbol{D}}, \hat{\boldsymbol{D}}$ are diagonal matrices and $\boldsymbol{E}$ and $\boldsymbol{F}$ are the strictly lower and upper parts of $\boldsymbol{A} = \mathrm{diag}(\boldsymbol{A}) + \boldsymbol{E} + \boldsymbol{F}$, allows implementation of the preconditioned CG or Bi-CG-type methods in which the computational labor is comparable to the unpreconditioned case. This strategy is due to Eisenstat [56], and it is commonly called the Eisenstat trick, see also refs. [190],[155]. Unfortunately, the usefulness of this strategy is somewhat limited. For a very sparse matrices, such as resulting from a low order FE discretizations of the diffusion equation, the triangular solution including short rows is the main bottleneck in a typical supercomputer implementation. Also, the quality of the split-preconditioners (6.24), which can be used in the Eisenstat trick, is not good enough in shell problems.

[10] The name corrected incomplete Cholesky is adopted from ref. [159].

Sparse approximate inverse preconditioners have recently received considerable attention, mainly because of their good vectorization and parallelization properties. These techniques are based on the explicit construction of a sparse matrix $\boldsymbol{M}^{-1}$ which directly approximates $\boldsymbol{A}^{-1}$. This is in contrast to more traditional implicit techniques where the matrix $\boldsymbol{M}$, rather than $\boldsymbol{M}^{-1}$, is explicitly available. The preconditioning step with an approximate inverse preconditioner $\boldsymbol{M}^{-1}$ requires only matrix-vector products, and is easily implemented on vector and parallel architectures. On the other hand, the construction of the preconditioner itself can be time-consuming, and the convergence rates obtained are often not as good as those obtained with implicit techniques.

Approximate inverse techniques rely on the assumption that for a given sparse matrix $\boldsymbol{A}$ it is possible to find a sparse matrix which is a good approximation of $\boldsymbol{A}^{-1}$. However, this is not necessarily obvious, since the inverse of a sparse matrix is usually dense see refs.[22], [23]. There are two main categories of approximate inverse techniques: methods which directly computes the entries of the approximate inverse [12], [82], [155], or the inverse factors of the matrix [20], [106].

Advantages of the *Factorized* sparse approximate inverse technique, commonly refered to as the FSAI method, in comparison to the sparse approximate inverse preconditioners (SPAI) are that the symmetry and positive definiteness are easy to insure. In the FSAI approach a lower triangular matrix $\boldsymbol{G}$ is computed as the (unique) solution of the constrained minimization problem

$$\min \ \|\boldsymbol{I} - \boldsymbol{G}\boldsymbol{L}\| \quad \text{subject to } \boldsymbol{G} \in \mathcal{L}$$

where $\boldsymbol{L}$ now denotes the Cholesky factor of $\boldsymbol{A}$ and $\mathcal{L}$ is a set of lower triangular matrices with a prescribed nonzero pattern (which must include the main diagonal). Here the matrix norm is the Frobenius norm or some weighted variant of it. It is possible to solve the above minimization problem without any knowledge of $\boldsymbol{L}$, just working with the original matrix $\boldsymbol{A}$; see [106]. The minimization problem decouples in n independent linear systems of relatively small size which can be solved in parallel. The approximate inverse preconditioner is then $\boldsymbol{M}^{-1} = \boldsymbol{G}^T\boldsymbol{G}$. The main difficulty associated with this approach is the choice of the sparsity pattern of $\boldsymbol{G}$, i.e., the determination of the constraint set $\mathcal{L}$. A simple solution is to restrict $\boldsymbol{G}$ to have the same sparsity pattern as the lower triangular part of $\boldsymbol{A}$, but this choice works well only for simple problems. Nonzero patterns associated with higher powers of $\boldsymbol{A}$ could also be used, but then the costs associated with the preconditioner construction and application increase. Moreover, for difficult problems even this more expensive approach may be ineffective.

Another approach to factorized approximate inverse preconditioning was proposed in [20]. This approach, which does not require that the sparsity pattern be known in advance, is based on a $\boldsymbol{A}$-orthogonalization process—that is, a Gram–Schmidt process with respect to the energy inner product $\langle \boldsymbol{x},\ \boldsymbol{y}\rangle = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}$. Given $\boldsymbol{A}$ and an arbitrary set of n linearly independent vectors, this algorithm computes a set of n vectors $\{\boldsymbol{z}_i\}_{i=1}^n$ which are conjugate with respect to $\boldsymbol{A}$, i.e. $\boldsymbol{A}$-orthogonal. If we introduce the matrix $\boldsymbol{Z} = [\boldsymbol{z}_1,\ \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n]$

then

$$\boldsymbol{Z}^T \boldsymbol{A} \boldsymbol{Z} = \boldsymbol{D} = \mathrm{diag}(p_1, p_2, \ldots, p_n)$$

where $p_i = \boldsymbol{z}_i^T \boldsymbol{A} \boldsymbol{z}_i \neq 0$. It follows that

$$\boldsymbol{A}^{-1} = \boldsymbol{Z} \boldsymbol{D}^{-1} \boldsymbol{Z}^T = \sum_{i=1}^{n} \frac{\boldsymbol{z}_i \boldsymbol{z}_i^T}{p_i}$$

and a factorized form of $\boldsymbol{A}^{-1}$ is obtained.

When the $\boldsymbol{A}$-orthogonalization process is applied to the standard basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$, it is easy to see that $\boldsymbol{Z}$ is unit upper triangular, and indeed $\boldsymbol{Z} = \boldsymbol{L}^{-T}$ where $\boldsymbol{A} = \boldsymbol{L} \boldsymbol{D} \boldsymbol{L}^T$ is the root-free Cholesky factorization of $\boldsymbol{A}$.

In order to get a sparse preconditioner, $\boldsymbol{Z}$ is computed incompletely, by dropping entries in the vector update operations. This can be done either on the basis of position, whereby nonzero entries outside a prescribed nonzero pattern are dropped, or on the basis of magnitude, whereby nonzeros are dropped if smaller than a prescribed drop tolerance in absolute value. This leads to approximate factors $\bar{\boldsymbol{Z}} \approx \boldsymbol{Z}$ and $\bar{\boldsymbol{D}} \approx \boldsymbol{D}$, and a factorized approximate inverse is obtained as $\boldsymbol{M}^{-1} = \bar{\boldsymbol{Z}} \bar{\boldsymbol{D}}^{-1} \bar{\boldsymbol{Z}}^T$. The stability of this procedure for certain classes of matrices, including diagonally dominant ones, was proved in [20]. In addition, numerical experiments in [20] and [22] showed that this approach performs well on linear systems arising from various applications, such as the discretization by finite differences of elliptic partial differential equations and the finite element analysis of simple structures. In particular, the experiments in [22] showed that on vector computers this technique can be superior to IC methods because of good vectorization properties. However, for thin shells the FSAI approach seems to be more robust.

The difficulty with the drop tolerance based AINV strategy is that the rejection strategy seems to drop out all the terms related to membrane deformations. This problem is also present in the drop tolerance based incomplete factorization preconditioners.[11] The dropping criteria used by Ajiz and Jennings [2] for IC factorizations seems to perform fairly well.

Orderings can also have a profound effect on the convergence of the accelerator iteration. Classical paper on the effect of orderings on incomplete factorizations is by Duff and Meurant [54]. For approximative inverse preconditioners the effect is studied in refs. [23], [31].

Element by element techniques are attractive due to their good parallelization properties [116], [128]. However, their convergence in thin shell applications seems to be slower than the IC-factorization based preconditioners [159].

[11] In this case the dropping strategy easily neglects terms relevant for bending deformations.

Bibliography

[1] J.P. Abbot. An efficient algorithm for the determination of certain bifurcation points. *Journal Computational and Applied Mathematics*, 4:19–27, 1987.

[2] M.A. Ajiz and A. Jennings. A robust incomplete Cholesky conjugate gradient algorithm. *International Journal for Numerical Methods in Engineering*, 20:949–966, 1984.

[3] F.A. Akl, W.H. Dilger, and B.M. Irons. Acceleration of subspace iteration. *International Journal for Numerical Methods in Engineering*, 18:583–589, 1982.

[4] E.L. Allgower and C.-S. Chien. Continuation and local perturbation for multiple bifurcations. *SIAM Journal on Scientific and Statistical Computing*, 7:1265–1281, 1986.

[5] E.L. Allgower, C.-S. Chien, K. Georg, and C.-F. Wang. Conjugate gradient methods for continuation problems. *Journal Computational and Applied Mathematics*, 38:1–16, 1991.

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

[7] E.L. Allgower and K. Georg. *Continuation and path following*, volume 2 of *Acta Numerica*, pages 1–64. Cambridge University Press, 1993.

[8] E.L. Allgower, K. Georg, and R. Miranda. The method of resultants for computing real solutions of polynomial systems. *SIAM Journal on Numerical Analysis*, 29:831–844, 1992.

[9] J.H. Argyris. Continua and discontinua. In *Conference of Matrix Methods in Structural Mechanics*, Wright Patterson AFB, Ohio, 1965.

[10] M. Avriel. *Nonlinear Programming, Analysis and Methods*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.

[11] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

[12] S.T. Barnard, L.M. Bernardo, and H.D. Simon. An MPI implementation of the SPAI preconditioner on the T3E. Technical Report 40794, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA, September 1997.

[13] E. Barragy and C.F. Carey. A partitioning scheme and iterative solution for sparse bordered systems. *Computer Methods in Applied Mechanics and Engineering*, 70:321–327, 1988.

[14] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for iterative methods*. SIAM, 1994.

[15] K.-J. Bathe and S. Ramaswamy. An accelerated subspace iteration method. *Computer Methods in Applied Mechanics and Engineering*, 23:313–330, 1980.

[16] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[17] K.J. Bathe and E.N. Dvorkin. On the automatic solution of non-linear finite element equations. *Computers and Structures*, 17:871–879, 1983.

[18] K. Bell. *Eigensolvers for Structural Problems*. Delft University Press, 1998.

[19] M. Benzi, R. Kouhia, and M. Tůma. An assesment of some preconditioning techniques in shell problems. *Communications in Numerical Methods in Engineering*, 1998. in press.

[20] M. Benzi, C.D. Meyer, and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM Journal on Scientific Computing*, 15(5):1135–1149, 1996.

[21] M. Benzi and M. Tůma. Numerical experiments with two approximate inverse preconditioners, 1997. CERFACS TR/PA/97/11.

[22] M. Benzi and M. Tůma. A comparative study of sparse approximative inverse preconditioners, January 1998. Technical Report LA-UR-98-0024, Los Alamos National Laboratory, Los Alamos, MN.

[23] M. Benzi and M. Tůma. Orderings for sparse approximative inverse preconditioners, May 1998. Technical Report LA-UR-98-2175, Los Alamos National Laboratory, Los Alamos, MN.

[24] P.G. Bergan. Solution by iteration in displacement and load spaces. In W. Wunderlich, E. Stein, and K.-J. Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 217–235, Bochum, Germany, 1981. Ruhr Universität, Springer Verlag.

[25] P.G. Bergan, G. Horrigmoe, and B. Kråkeland. Solution techniques for nonlinear finite element problems. *International Journal for Numerical Methods in Engineering*, 12:1677–1696, 1978.

[26] L. Bernspång. Iterative and adaptive solution techniques in computational plasticity. Technical Report 91:8, Chalmers Univ. of Tech., Department of Structural Mechanics, 1991.

[27] N. Bićanić and K.H. Johnson. Who was '-Raphson'? *International Journal for Numerical Methods in Engineering*, 14:148–152, 1979.

[28] R.O. Bjærum. *Finite element formulations and solution algorithms for buckling and collapse analysis of thin shells*. PhD thesis, Division of Structural Engineering, The Norwegian Institute of Technology, 1992.

[29] J.G.L. Booten, H.A. van der Vorst, P.M. Meijer, and H.J.J. te Riele. A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems. Technical Report NM-R9414, CWI, July 1994.

[30] T. Braconnier, F. Chatelin, and V. Frayssé. The influence of large nonnormality on the quality of convergence of iterative methods in linear algebra. Technical Report TR/PA/94/07, Cerfacs, 1994.

[31] R. Bridson and W.-P. Tang. Ordering, anisotropy and factored sparse approximate inverses, 1998. Preprint, Department of Computer Science, University of Waterloo.

[32] K.W. Brodlie, A.R. Gourlay, and J. Greenstadt. Rank-one and rank-two corrections to positive definite matrices expressed in product form. *J. Inst. Maths Applics*, 11:73–82, 1973.

[33] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593, 1965.

[34] C.G. Broyden. A new double-rank minimization algorithm. *Notices Amer. Math. Soc.*, 16:670, 1969.

[35] C.G. Broyden. The convergence of single-rank quasi-newton methods. *Mathematics of Computation*, 24:365–382, 1970.

[36] A.M. Bruaset. *A Survey of Preconditioned Iterative Methods*. Number 328 in Pitman Research Notes in Mathematics Series. Longman Scientific & Technical, 1995.

[37] B. Budiansky. *Theory of buckling and post buckling of elastic structures*, volume 14 of *Advances in Applied Mechanics*, pages 1–65. Academic Press, London, 1974.

[38] V.E. Bulgakov. The use of the multi-level iterative aggregation method in 3-D finite element analysis of solid, truss, frame and shell structures. *Computers and Structures*, 63(5):927–938, 1997.

[39] E. Byskov and J.W. Hutchinson. Mode interaction in axially stiffened cylindrical shells. *AIAA Journal*, 15:941–948, 1977.

[40] R. Casciaro, G. Salerno, and A.D. Lanzo. Finite element asymptotic analysis of slender elastic structures: a simple approach. *International Journal for Numerical Methods in Engineering*, pages 1397–1426, 1992.

[41] F. Chaitin-Chatelin. Is nonnormality a serious computational difficulty in practice? Technical Report TR/PA/96/33, Cerfacs, 1996.

[42] T.F. Chan and Y. Saad. Iterative methods for solving bordered systems with applications to continuation methods. *SIAM Journal on Scientific and Statistical Computing*, 6(2):438–451, 1985.

[43] T.F. Chan and T. Szeto. Composite step product methods for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 17(6):1491–1508, 1996.

[44] H. Chen and G.E. Blandford. Work-increment-control method for non-linear analysis. *International Journal for Numerical Methods in Engineering*, 36:909–930, 1993.

[45] M.A. Crisfield. A faster modified newton-raphson iteration. *Computer Methods in Applied Mechanics and Engineering*, 20:267–278, 1979.

[46] M.A. Crisfield. A fast incremental/iterative solution procedure that handles snap-through. *Computers and Structures*, 13:55–62, 1981.

[47] M.A. Crisfield. Accelerated solution techniques and concrete cracking. *Computer Methods in Applied Mechanics and Engineering*, 33:585–607, 1982.

[48] M.A. Crisfield. A quadratic Mindlin element using shear constraints. *Computers and Structures*, 18:833–852, 1984.

[49] M.A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons, 1991.

[50] D.W. Decker and C.T. Kelley. Expanded convergence domains for Newton's method at nearly singular roots. *SIAM Journal on Scientific and Statistical Computing*, 6:951–966, 1985.

[51] J.E. Dennis and J.J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[52] J.E. Dennis and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Classics in Applied Mathematics. SIAM, Philadelphia, 1996. First published by Pretice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[53] P. Deuflhard, R. Freund, and A. Walter. Fast secant methods for the iterative solution of large nonsymmetric linear systems. *Impact of Computing in Sciece and Engineering*, 2:244–276, 1990.

[54] I. Duff and G.A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29:635–657, 1989.

[55] I.S. Duff. Sparse numerical linear algebra: direct methods and preconditioning. Technical Report TR/PA/96/22, CERFACS, 1996.

[56] S.C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM Journal on Scientific and Statistical Computing*, 2:1–4, 1981.

[57] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in inexact Newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.

[58] A. Eriksson. Using eigenvector projections to improve convergence in non-linear finite element equilibrium iterations. *International Journal for Numerical Methods in Engineering*, 24:497–512, 1987.

[59] A. Eriksson. On some path-related measures for non-linear structural F.E. problems. *International Journal for Numerical Methods in Engineering*, 26:1791–1803, 1988.

[60] A. Eriksson. On linear constraints for Newton-Raphson corrections and critical point searches in structural F.E. problems. *International Journal for Numerical Methods in Engineering*, 28:1317–1334, 1989.

[61] A. Eriksson. Derivatives of tangential stiffness matrices for equilibrium path descriptions. *International Journal for Numerical Methods in Engineering*, 32:1093–1113, 1991.

[62] A. Eriksson. On improved predictions for structural equilibrium path evaluations. *International Journal for Numerical Methods in Engineering*, 36:201–220, 1993.

[63] A. Eriksson. Fold lines for sensitivity analyses in structural instability. *Computer Methods in Applied Mechanics and Engineering*, 114:77–101, 1994.

[64] A. Eriksson and R. Kouhia. On step size adjustments in structural continuation problems. *Computers and Structures*, 55:495–505, 1995.

[65] A. Eriksson, C. Pacoste, and A. Zdunek. Numerical analysis of complex instability behaviour using incremental-iterative strategies. *Computer Methods in Applied Mechanics and Engineering*, 179:265–305, 1999.

[66] G.M. van Erp. *Advanced Buckling Analyses of Beams with Arbitrary Cross Sections*. PhD thesis, Eindhoven University of Technology, 1989.

[67] F. Ficken. The continuation method for functional equations. *Communications on Pure and Applied Mathematics*, 4:435–456, 1951.

[68] J.P. Fink and W.C. Rheinboldt. The role of tangent mapping in analyzing bifurcation behaviour. *Zeitschrift für Angewandte Mathematik und Mechanik*, 64(9):407–412, 1984.

[69] J. Fish and V. Belsky. Generalized aggregation multilever solver. *International Journal for Numerical Methods in Engineering*, 40:4341–4361, 1997.

[70] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[71] D.R. Fokkema, G.L.G. Sleijpen, and H.A. van der Vorst. Accelerated inexact Newton schemes for large systems of nonlinear equations. Technical Report 918, Universiteit Utrecht, Department of Mathematics, July 1995.

[72] B.W.R. Forde and S.F. Stiemer. Improved arc-length orthogonality methods for nonlinear finite element analysis. *Computers and Structures*, 27:625–630, 1987.

[73] R.W. Freund. Preconditioning of symmetric, but highly indefinite linear systems. In A. Sydow, editor, *15th IMACS World Congress on Scientific Modelling and Applied Mathematics, Vol 2 Numerical Mathematics*, pages 551–556, 1997.

[74] I. Fried. Orthogonal trajectory accession to the equilibrium curve. *Computer Methods in Applied Mechanics and Engineering*, 47:283–297, 1984.

[75] K. Georg. On tracing an implicitely defined curve by quasi-Newton steps and calculating bifurcation by local perturbations. *SIAM Journal on Scientific and Statistical Computing*, 2:35–50, 1981.

[76] K. Georg. A note on stepsize control for numerical curve following. In B.C. Eaves, F.J. Gould, H.-O. Peitgen, and M.J. Todd, editors, *Homotopy Methods and Global Convergence*, pages 145–154. Plenum, 1983.

[77] M. Geradin, M. Hogge, and S. Idelsohn. Implicit finite element methods. In T. Belytchko and T.J.R. Hughes, editors, *Computational Methods for Transient Analysis*, chapter 7. North-Holland, 1983.

[78] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.

[79] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1989.

[80] M. Golubitsky and D.G. Schaeffer. *Singularities and Groups in Bifurcation Theory*, volume 1. Springer-Verlag, 1985.

[81] R.G. Grimes, J.G. Lewis, and H.D. Simon. A shifted block Lanczos algorithm for solving sparse symmetric eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 15:228–272, 1994.

[82] M.J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing*, 18(3):838–853, 1997.

[83] R.T. Haftka, R.H. Mallet, and W. Nachbar. Adaptation of Koiter's method to finite element analysis of snap-through buckling behaviour. *International Journal of Solids and Structures*, 7:1427–1445, 1971.

[84] A.R. Hall. *Isaac Newton, Adventurer in Thought*. Cambridge University Press, 1996.

[85] C.B. Haselgrove. The solution of non-linear equations and of differential equations with two point boundary conditions. *Computer Journal*, 4:225–259, 1961.

[86] C. den Heijer and W.C. Rheinboldt. On steplength algorithms for a class of continuation methods. *SIAM Journal on Numerical Analysis*, 18(5):925–948, 1981.

[87] B.-Z. Huang and S.N. Atluri. A simple method to follow post-buckling paths in finite element analysis. *Computers and Structures*, 57(3):477–489, 1995.

[88] T.J.R. Hughes. *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[89] J. Huitfeldt. Nonlinear eigenvalue problems - prediction of bifurcation points and branch switching. Technical Report 17, Department of Computer Sciences, Chalmers University of technology, 1991.

[90] J. Huitfeldt and A. Ruhe. A new algorithm for numerical path following applied to an example from hydrodynamic flow. *SIAM Journal on Scientific and Statistical Computing*, 11:1181–1192, 1990.

[91] J.W. Hutchinson. *Plastic buckling*, volume 14 of *Advances in Applied Mechanics*, pages 67–144. Academic Press, London, 1974.

[92] B. Irons and A. Elsawaf. The conjugate Newton algorithm for solving finite element equations. In K.J. Bathe, T.J. Oden, and W. Wunderlich, editors, *Formulations and Computational Algorithms in Finite Element Analysis*, pages 656–672. MIT Press, 1977.

[93] A.D. Jepson and A. Spence. On a reduction process for nonlinear equations. *SIAM Journal on Mathematical Analysis*, 20(1):39–56, 1989.

[94] K.C. Johns. Simultaneous buckling in symmetric structural systems. *Engineering Mechanics Division, Proceedings of the American Society of Civil Engineers*, 98:835–848, 1972.

[95] K.C. Johns and A.H. Chilver. Multiple path generation at coincident branching points. *International Journal of Engineering Science*, 13:899–910, 1971.

[96] D. Karamanlidis, A. Honecker, and K. Knothe. Large deflection finite element analysis of pre- and postcritical response of thin elastic frames. In W. Wunderlich, E. Stein, and K.-J. Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 217–235, Bochum, Germany, 1981. Ruhr Universität, Springer Verlag.

[97] R.B. Kearfott. Some general bifurcation techniques. *SIAM Journal on Scientific and Statistical Computing*, 4:52–68, 1983.

[98] J.P. Keener. Perturbed bifurcation theory at multiple eigenvalues. *Archive for Rational Mechanics and Analysis*, 56:348–366, 1974.

[99] J.P. Keener and H.B. Keller. Perturbed bifurcation theory. *Archive for Rational Mechanics and Analysis*, 50:159–175, 1973.

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

[101] H.B. Keller. The bordering algorithm and path following near singular points of higher nullity. *SIAM Journal on Scientific and Statistical Computing*, 4:573–582, 1983.

[102] H.B. Keller. *Lectures on Numerical Methods in Bifurcation Problems*. Springer Verlag, 1987.

[103] H.B. Keller and W.F. Langford. Iterations, perturbations and multiplicities for nonlinear bifurcation problems. *Arch. Rational Mech. Anal.*, 48:83–108, 1972.

[104] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.

[105] W.T. Koiter. *Over de stabiliteit van het elastisch evenwicht* (in Dutch). PhD thesis, Technische Hogeschool, Delft, 1945. English tranlations: NASA TT F10, 833 (1967) and AFFDL, TR-7025 (1970).

[106] L.Y. Kolotilina and A.Y. Yeremin. Factorized sparse approximate inverse preconditionings I. Theory. *SIAM Journal on Matrix Analysis and Applications*, 14:45–58, 1993.

[107] R. Kouhia, C.M. Menken, M. Mikkola, and G.-J. Schreppers. Computing and understanding interactive buckling. In R.A.E. Mäkinen and P. Neittaanmäki, editors, *Proceedings of the 5th Finnish Mechanics Days*, pages 53–61, 1994.

[108] R. Kouhia and M. Mikkola. Tracing the equilibrium path beyond simple critical points. *International Journal for Numerical Methods in Engineering*, 28(12):2933–2941, 1989.

[109] R. Kouhia and M. Mikkola. Tracing the equilibrium path beyond compound critical points. *International Journal for Numerical Methods in Engineering*, 46:1049–1074, 1999.

[110] S. Krenk. An orthogonal residual procedure for non-linear finite element equations. *International Journal for Numerical Methods in Engineering*, 38(5):823–839, 1995.

[111] S. Krenk and O. Hededal. A dual orthogonality procedure for non-linear finite element equations. *Computer Methods in Applied Mechanics and Engineering*, 123:95–107, 1995.

[112] P. Kunkel. Quadratically convergent methods for the computation of unfolded singularities. *SIAM Journal on Numerical Analysis*, 25(6):1392–1408, 1988.

[113] A.D. Lanzo, G. Garcea, and R. Casciaro. Asymptotic post-buckling analysis of rectangular plates by HC finite elements. *International Journal for Numerical Methods in Engineering*, 38:2325–2345, 1995.

[114] S.H. Lee. Rudimentary considerations for effective quasi-newton updates in nonlinear finite element analysis. *Computers and Structures*, 33:463–476, 1989.

[115] S.H. Lee. Rudimentary considerations for effective quasi-Newton updates in nonlinear finite element analysis. *Computers and Structures*, 33(2):463–476, 1989.

[116] J.-Y. L'Excellent. *Utilisation de préconditionneurs élément-par-élément pour la résolution de problèmes d'optimisation de grande taille*. PhD thesis, Institut National Polytechnique de Toulouse, 1995.

[117] S. Lopez. Detection of bifurcation points along a curve traced by a continuation method. *International Journal for Numerical Methods in Engineering*, 53:983–1004, 2002.

[118] S. Lopez. Post-critical analysis of structures with a nonlinear pre-buckling state in the presence of imperfections. *Computer Methods in Applied Mechanics and Engineering*, 191:4421–4440, 2002.

[119] A. Magnusson and I. Svensson. Numerical treatment of complete load-deflection curves. *International Journal for Numerical Methods in Engineering*, 41:955–971, 1998.

[120] R.H. Mallet and P.V. Marcal. Finite element analysis of non-linear structures. *Journal of Structural Division, ASCE*, 94:2081–2105, 1968.

[121] O.A. Marques. BLZPACK: description and users guide. Technical Report TR/PA/95/30, CERFACS, 1995.

[122] H. Matthies and G. Strang. The solution of nonlinear finite element equations. *International Journal for Numerical Methods in Engineering*, 14:1613–1626, 1979.

[123] J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31:148–162, 1977.

[124] C.M. Menken, W.J. Groot, and G.A.J. Stallenberg. Interactive buckling of beams in bending. *Thin-Walled Structures*, 12:415–434, 1991.

[125] A. Morgan. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*. Prentice-Hall, 1987.

[126] E. O n ate and W.T. Matias. A critical displacement approach for predicting structural instability. *Computer Methods in Applied Mechanics and Engineering*, 134:135–161, 1996.

[127] J. Nocedal. *Theory of algorithms for unconstrained optimization*, volume 1 of *Acta Numerica*, pages 199–242. Cambridge University Press, 1992.

[128] B. Nour-Omid and B.N. Parlett. Element preconditioning using splitting techniques. *SIAM Journal on Scientific and Statistical Computing*, 6(3):761–770, 1985.

[129] J.T. Oden. Numerical formulation of non-linear elasticity problems. *Journal of Structural Division, ASCE*, 93:235–255, 1967.

[130] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

[131] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12:617–629, 1975.

[132] S. Pajunen. Sauvarakenteiden epälineaarinen analysointi (Nonlinear analysis of bar structures), 1997. Licentiates thesis, (in Finnish) Tampere University of Technology, Department of Civil Engineering.

[133] M. Papadrakakis. Post-buckling analysis of spatial structures by vector iteration methods. *Computers and Structures*, 14(5–6):393–402, 1981.

[134] M. Papadrakakis. A truncated Newton-Lanczos method for overcoming limit and bifurcation points. *International Journal for Numerical Methods in Engineering*, 29:1065–1077, 1990.

[135] M. Papadrakakis and C.J. Gantes. Truncated Newton methods for nonlinear finite element analysis. *Computers and Structures*, 30:705–715, 1988.

[136] M. Papadrakakis and C.J. Gantes. Preconditioned conjugate- and secant-Newton methods for non-linear problems. *International Journal for Numerical Methods in Engineering*, 28:1299–1316, 1989.

[137] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[138] B.N. Parlett. Symmetric matrix pencils. *Journal Computational and Applied Mathematics*, 38:373–385, 1991.

[139] R. Peek and M. Kheyrkhahan. Postbuckling behaviour and imperfection sensitivity of elastic structures by the Lyapunov-Schmidt-Koiter approach. *Computer Methods in Applied Mechanics and Engineering*, 108:261–279, 1993.

[140] G. Peters and J.H. Wilkinson. $Ax = \lambda Bx$ and the generalized eigenproblem. *SIAM Journal on Numerical Analysis*, 7(4):479–492, 1970.

[141] M. Pignataro, A. Luongo, and N. Rizzi. On the effect of the local-overall interaction on the post-buckling of uniformly compressed channels. *Thin-Walled Structures*, 3:1470–1486, 1986.

[142] M. Potier-Ferry. *Buckling and Post-Buckling*, volume 288 of *Lecture Notes in Physics*, pages 205–223. Springer-Verlag, 1987.

[143] M.J.D. Powell. Hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, chapter 6. Gordon and Breach Science Publishers, 1970.

[144] Y. Quian and G. Dhatt. An accelerated subspace method for generalized eigenproblems. *Computers and Structures*, 54(6):1127–1134, 1995.

[145] E. Ramm. Strategies for tracing the nonlinear responce near limit points. In W. Wunderlich, E. Stein, and K.-J. Bathe, editors, *Nonlinear Finite Element Analysis in Structural Mechanics*, pages 63–89, Bochum, Germany, 1981. Ruhr Universität, Springer Verlag.

[146] W.C. Rheinboldt. Numerical continuation methods for finite element applications. In K.J. Bathe et al., editor, *Formulations and and Computational Algorithms in FE analysis*, pages 599–6xx, 1977.

[147] W.C. Rheinboldt. Numerical methods for a class of finite dimensional bifurcation problems. *SIAM Journal on Numerical Analysis*, 15:1–11, 1978.

[148] W.C. Rheinboldt. *Numerical Analysis of Parametrized Nonlinear Equations*. Wiley, 1986.

[149] W.C. Rheinboldt. On the computation of multi-dimensional solution manifolds of parametrized equations. *Numerische Mathematik*, 53:165–181, 1988.

[150] E. Riks. On the numerical solution of snapping problems in the theory of elastic stability. Technical report, Stanford University, Department of Aeronautics and Astronautics, 1970.

[151] E. Riks. The incremental solution of some basic problems in elastic stability. Technical Report NLR TR 74005 U, National Aerospace Laboratory, The Netherlands, 1974.

[152] E. Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15:529–551, 1979.

[153] E. Riks. Some computational aspects of the stability analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 57:219–259, 1984.

[154] K. Runesson, A. Samuelsson, and L. Bernspång. Numerical technique in plasticity including solution advancement control. *International Journal for Numerical Methods in Engineering*, 22:769–788, 1986.

[155] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, 1996.

[156] Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[157] A.B. Sabir and A.C. Lock. The application of finite elements to the large-deflection geometrically non-linear behaviour of cylindrical shells. In *Proceedings of the International Conference on Variational Methods in Engineering*, pages 67–76, 1972.

[158] P. Saint-Georges, G. Warzee, R. Beauwens, and Y. Notay. High-performance PCG solvers for FEM structural analysis. *International Journal for Numerical Methods in Engineering*, 39:1313–1340, 1996.

[159] P. Saint-Georges, G. Warzee, Y. Notay, and R. Beauwens. Fast iterative solvers for finite element analysis in general and shell analysis in particular. In B.H.V. Topping, editor, *Advances in Finite Element Technology*, pages 273–282, Edinburgh, 1996. Civil-Comp Press.

[160] G. Salerno and R. Casciaro. Mode jumping and attractive paths in multimode elastic buckling. *International Journal for Numerical Methods in Engineering*, 40(5):833–861, 1997.

[161] W.F. Schmidt. Adaptive step size selection for use with the continuation method. *International Journal for Numerical Methods in Engineering*, 12:677–694, 1978.

[162] K.H. Schweizerhof and P. Wriggers. Consistent linearization for path following methods in nonlinear FE analysis. *Computer Methods in Applied Mechanics and Engineering*, 59:261–279, 1986.

[163] H. Schwetlick. On the choice of steplength in path following methods. *Zeitschrift für Angewandte Mathematik und Mechanik*, 64(9):391–396, 1984.

[164] M.J. Sewell. On the connection between stability and the shape of the equilibrium surface. *Journal of Mechanics and Physics of Solids*, 14:203–230, 1966.

[165] M.J. Sewell. A general theory of equilibrium paths through critical points. *Proceedings of the Royal Society - A*, 306:201–238, 1968.

[166] M.J. Sewell. On the branching of equilibrium paths. *Proceedings of the Royal Society - A*, 315:499–518, 1970.

[167] R. Seydel. Numerical computation of branch points in nonlinear equations. *Numerische Mathematik*, 33:339–352, 1979.

[168] R. Seydel. On detecting stationary bifurcations. *International Journal on Bifurcation and Chaos*, 1:335–337, 1991.

[169] R. Seydel. *Practical Bifurcation and Stability Analysis*. Springer-Verlag, 1994.

[170] D.F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.

[171] P. Sharifi and E.P. Popov. Nonlinear buckling analysis of sandwich arches. *Journal of the Engineering Mechanics Division*, 97:1397–1411, 1971.

[172] J. Shi and M.A. Crisfield. A simple indicator and branch switching technique for hidden unstable equilibrium paths. *Finite Elements in Analysis and Design*, 12:303–312, 1992.

[173] J. Shi and M.A. Crisfield. A semi-direct approach for the computation of singular points. *Computers and Structures*, 51:107–15, 1994.

[174] J.C. Simo, P. Wriggers, K.Schweizerhof, and R.L. Taylor. Finite deformation postbuckling analysis involving inelasticity and contact constraints. *International Journal for Numerical Methods in Engineering*, 23:779–800, 1986.

[175] G. Skeie and C.A. Felippa. Detecting and traversing bifurcation points in nonlinear structural analysis. *International Journal of Space Structures*, 6(2):77–98, 1991.

[176] G.L.G. Sleijpen and H.A. van der Vorst. An overview of approaches for the stable computation of hybrid Bi-CG methods. Technical Report 908, Universiteit Utrecht, Department of Mathematics, March 1995.

[177] G.L.G. Sleijpen, H.A. van der Vorst, and M. van Gijzen. Quadratic eigenproblems are no problem. *SIAM News*, 29(7):8–9, 1996.

[178] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 10:36–52, 1989.

[179] I. Stakgold. Branching of solutions of nonlinear equations. *SIAM Review*, 13(3):289–332, 1971.

[180] D. Szyld. Criteria for combining inverse and Rayleigh quotient iteration. *SIAM Journal on Numerical Analysis*, 25(6):1369–1375, 1988.

[181] J.M.T. Thompson and G.W. Hunt. *A General Theory of Elastic Stability*. Wiley, London, 1973.

[182] J.M.T. Thompson and G.W. Hunt. *Elastic Instability Phenomena*. Wiley, Chichester, 1984.

[183] G.A. Thurston. Continuation of newton's method through bifurcation points. *Journal of Applied Mechanics*, 9:425–430, 1969.

[184] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001.

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

[186] M.J. Turner, E.H. Dill, H.C. Martin, and R.J. Melosh. Large deflection of structures subject to heating and external load. *Journal of the Aerospace Sciences*, 27:97–106, 1960.

[187] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid on unconstrained meshes. *Computing*, 56:179–196, 1996.

[188] H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly convergent variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[189] H.A. van der Vorst and G.H. Golub. 150 years old and still alive: eigenproblems. In I.S. Duff and G.A. Watson, editors, *The State of the Art in Numerical Analysis*, pages 93–119. Clarendon Press, 1997.

[190] H. Voss. Iterative methods for linear systems of equations. University of Jyväskylä, Department of Mathematics, lecture notes 27, 1993.

[191] W. Wagner. A path-following algorithm with quadratic predictor. *Computers and Structures*, 39:339–348, 1991.

[192] W. Wagner and P. Wriggers. A simple method for the calculation of postcritical branches. *Engineering Computation*, 5:103–109, 1988.

[193] H.F. Walker. An adaptation of Krylov subspace methods to path following problems. *SIAM Journal on Scientific Computing*, 21:1191–1198, 1999.

[194] Z. Waszczyszyn. Numerical problems of nonlinear stability analysis of elastic structures. *Computers and Structures*, 17:13–24, 1983.

[195] G.A. Wempner. Discrete approximations related to the nonlinear theories of solids. *International Journal of Solids and Structures*, 7:1581–1599, 1971.

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

[197] P. Wriggers and J.C. Simo. A general procedure for the direct computation of turning and bifurcation problems. *International Journal for Numerical Methods in Engineering*, 30:155–176, 1990.

[198] T.J. Ypma. Historical development of the Newton-Raphson method. *SIAM Review*, 37(4):531–551, 1995.

[199] S.-L. Zhang. GPBi-CG: generalized product-type methods based on Bi-CG for solving non-symmetric linear systems. *SIAM Journal on Scientific Computing*, 18(2):537–551, 1997.