22

Accelerators and Line Search

TABLE OF CONTENTS

§22.1. Introduction

Newton-like methods were initially used in geometrically nonlinear analysis with a unit h. This is optimal for conventional Newton (CNM) near a solution, but not necessarily so otherwise. Since a key reason for letting h vary is to accelerate convergence to equilibrium, schemes that utilize nonunitary stepsizes are called *accelerators*. Another application is diagnosing and overcoming divergence in "difficult" regions of the response. The latter application is tied up with the *line search* technique that comes from the field of optimization. In the present Chapter we study acceleration formulas and line search schemes based on the relaxation equations.

§22.2. General Comments

The device of letting h be a free parameter has proven useful in the following situations.

1. *Oscillatory Response*. Successive iterates "bounce" around so that the residuals at successive steps are almost equal and opposite. This behavior is quite common when the structure "stiffens" globally. The cure is to go half-way, which means $h \approx 0.5$. Taking $h < 1$ to kill oscillations is sometimes called *underrelaxation*.
2. *Slow Monotonic Convergence*. Successive residuals "point" the same way, but hardly decrease. This behavior is quite common when the structure "softens" globally, as in extensive creep or plasticity. The remedy is to "overrelax" with $h > 1$.
3. *Erratic Behavior*. The response of many nonlinear structures exhibits "difficult regions" where sucessive residuals do not show significant correlation, and divergence is likely. If the nonlinearities are smooth, this behavior is observed as one approaches bifurcation points or sharp limit points. It is especially pronounced when several such points are clustered, as often happens in optimized structures. In hard nonlinear problems, such as contact and cracking, this behavior emanates from local effects. The recommended cure is cautious line search. If the search delivers a very small h, say $h < 0.1$, the program should recompute the stiffness matrix.

The determination of h at each iteration step can be used for monitoring these situations. For example, if $h \approx 0.5$ for two or three consecutive steps, chances are that the iterates are oscillating. Note that it does not necessarily follow that such h is in fact applied. For example, the author has sucessfully used the rule that if the predicted h is in the range 0.7 to 1.7, a value of $h = 1$ is used. The reason for this strategy is that the application of a nonunitary h generally demands additional residual calculations.

From the preceding considerations it follows that a "best" value of h need not be know precisely; *e.g.*, there is little practical difference between $h = 1.8$ and $h = 2.0$. This low accuracy requirement allows wide latitude in simplifying optimal-stepsize formulas so they are practical to use.

§22.3. Accelerator Models

Two models may be use to derive optimal stepsize formulas. At an equilibrium solution, $\mathbf{r} = 0$ and (if not a critical point) $\dot{\mathbf{u}} = \mathbf{0}$. Accelerators based on these two models are studied in the following subsections.

§22.3.1. Tangent Accelerators for Fixed λ

We begin by studying accelerators based on the zero-velocity model. If the nonlinearities are smooth, and an estimate of $\dot{\mathbf{K}}$ or of $\dot{\mathbf{K}}\dot{\mathbf{u}}$ can be easily procured, a technique due to Park [3] may be used for estimating h. Take the fixed-λ second-order relaxation equation in which the term $\ddot{\mathbf{W}}\mathbf{r}$ is neglected:

$$\mathbf{K}\ddot{\mathbf{u}} + (\dot{\mathbf{K}} + \mathbf{W}\mathbf{K})\dot{\mathbf{u}} = \mathbf{0}. \tag{22.1}$$

Treat this equation with the forward Euler integrator on $\dot{\mathbf{u}}$:

$$\dot{\mathbf{u}}^{(k+1)} - \dot{\mathbf{u}}^{(k)} = h\ddot{\mathbf{u}}^{(k)}, \tag{22.2}$$

and require that h be such that

$$\dot{\mathbf{u}}^{(k+1)} \approx \mathbf{0}, \tag{22.3}$$

whence

$$(\dot{\mathbf{K}}^{(k)} + \mathbf{W}^{(k)}\mathbf{K}^{(k)})\,\dot{\mathbf{u}}^{(k)} h \approx \mathbf{K}^{(k)}\dot{\mathbf{u}}^{(k)}. \tag{22.4}$$

For fixed λ we have $\dot{\mathbf{r}} = \mathbf{K}\dot{\mathbf{u}}$, which can be rewritten as

$$(\dot{\mathbf{K}}^{(k)}\dot{\mathbf{u}}^{(k)} + \mathbf{W}^{(k)}\dot{\mathbf{r}}^{(k)})\,h \approx \dot{\mathbf{r}}^{(k)}. \tag{22.5}$$

This is a highly undetermined system for h. Three main avenues of research open up here, leading to a scalar stepsize, a full-matrix stepsize and a diagonal-matrix stepsize, respectively. Only the first case is treated in some detail; the others are relegated to Remarks below.

Scalar Stepsize. The straightforward way to solve (22.5) is to premultiply both sides by a "weighting" vector $\mathbf{w}$. Park [3] proposes to use $\mathbf{w} = \dot{\mathbf{u}}$, a choice that makes h a Rayleigh quotient with respect to the direction $\dot{\mathbf{u}}$; the underlying idea being to anihilate the dominant error term. The result is

$$\boxed{h = \frac{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{r}}^{(k)}}{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{K}}^{(k)}\dot{\mathbf{u}}^{(k)} + \dot{\mathbf{u}}^{(k)\,T}\,\mathbf{W}^{(k)}\,\dot{\mathbf{r}}^{(k)}}} \tag{22.6}$$

This formula can now be specialized to various Newton-like methods by inserting the appropriate $\mathbf{W}$.

Remark 22.1. Other weighting vectors may be chosen. For example, $\Delta\mathbf{u}$, $\dot{\mathbf{r}}$ or $\mathbf{r}$. The latter choice yields accelerators of "minimal residual form", which are studied later as part of the $\mathbf{r} \approx \mathbf{0}$ model.

Conventional Newton. If $\mathbf{W} = \mathbf{I}$, (22.6) specializes to

$$h_{CNM} = \frac{1}{1+\nu}, \qquad \text{where} \quad \nu = \frac{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{K}}^{(k)}\dot{\mathbf{u}}^{(k)}}{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{r}}^{(k)}}. \tag{22.7}$$

If $\mathbf{K}$ is constant (linear system), then $\nu = 0$ and $h = 1$, which of course gives the correct answer. If the structure is stiffening (softening) in the $\dot{\mathbf{u}}$ direction, then h is less (greater) than 1, which is the right trend. Numerical experiments with this accelerator suggest that the "averaged" stepsize

$$h_{Halley} = \tfrac{1}{2}(1 + \frac{1}{1+\nu}). \tag{22.8}$$

gives marginally better performance; for a scalar equation the use of in fact yields Halley's third order iteration function; see e.g. Traub [18, p. 232].

Remark 22.2. For the quadratic scalar equation $r = u^2 = 0$, $u^{(0)} \neq 0$, we have $\dot{r} = 2u\dot{u}$, $\dot{K} = 2\dot{u}$, $\nu = -0.5$ and $h_{CNM} = 2$, which yields the exact solution $u = 0$ in one step. Similarly, for the equation $r = u^m = 0$ we get $h_{CNM} = m$ and again we arrive at the solution in one step.

Remark 22.3. At noncritical points the performance of h_{CNM} is not so dramatic as above; in fact it tends to "overshoot" the solution by the same amount as $h = 1$ tends to undershoot it. This is the rationale behind the Halley accelerator.

Modified Newton . Insertion of $\mathbf{W} = \mathbf{K}\overline{\mathbf{F}}$ into yields

$$\boxed{h_{MNM} = \frac{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{r}}^{(k)}}{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{K}}^{(k)}\dot{\mathbf{u}}^{(k)} + \dot{\mathbf{r}}^{(k)\,T}\,\overline{\mathbf{F}}\,\dot{\mathbf{r}}^{(k)}}} \tag{22.9}$$

If the term involving $\dot{\mathbf{K}}$ is neglected,

$$\boxed{h^*_{MNM} = \frac{\dot{\mathbf{u}}^{(k)\,T}\,\dot{\mathbf{r}}^{(k)}}{\dot{\mathbf{r}}^{(k)\,T}\,\overline{\mathbf{F}}\,\dot{\mathbf{r}}^{(k)}}} \tag{22.10}$$

which is more easily calculable given the information available in a MNM iteration. (A similar simplification in (80) would be too drastic, for it would reduce it to $h_{CNM} = 1$; but $\dot{\mathbf{K}}\dot{\mathbf{u}}$ is more easily calculable in CNM.) Accelerators for the damped Newton method (73) can be constructed in a similar manner.

Formulas such as (79) and (82) are called *tangent accelerators* as they depend only on the state at pseudotime t_k and t-derivatives there. But the presence of rates such as $\dot{\mathbf{K}}$ and $\dot{\mathbf{r}}$ causes computational difficulties because the rates can be readily calculated only in simple problems. The problem can be overcome by using secant information as discussed later.

Remark 22.4. *Stepsize Matrix*. A second approach to solving (78) is to transmute h into a "stepsize matrix" $\mathbf{H}$ given by

$$\mathbf{H} = (\dot{\mathbf{K}} + \mathbf{W}\mathbf{K})^{-1}\mathbf{K}. \tag{22.11}$$

This may be inserted to produce the iteration family

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - (\dot{\mathbf{K}}^{(k)} + \mathbf{W}^{(k)}\mathbf{K}^{(k)})^{-1}\mathbf{W}^{(k)}\,\mathbf{r}^{(k)}, \tag{22.12}$$

into which we may replace $\dot{\mathbf{K}}$ by the rank-two estimate derived from Quasi-Newton formulas in Felippa [19]. This approach is presently unexplored.

Remark 22.5. *Diagonalised Stepsize Matrix*. Yet a third approach that merits attention is to transmute h into a diagonal matrix, $\mathbf{H}_D$. This effectively assigns stepsizes *component by component*, which is in line with the empirical "alpha method" of Nayak and Basu [20]. To get the stepsize h_i associated with the i^{th} component of $\mathbf{u}$, premultiply both sides by $\mathbf{e}_i^T$, which is a row vector with unit i^{th} component and zero otherwise. The result is

$$h_i = \frac{\mathbf{e}_i^{(k)\,T}\,\dot{\mathbf{r}}^{(k)}}{\mathbf{e}_i^{(k)\,T}\,\dot{\mathbf{K}}^{(k)}\dot{\mathbf{u}}^{(k)} + \mathbf{e}_i^{(k)\,T}\,\mathbf{W}^{(k)}\,\dot{\mathbf{r}}^{(k)}} \tag{22.13}$$

Appropiate $\mathbf{W}$ may now be inserted and rates replaced by secant relations.

Tangent Accelerators for Variable λ

Let now λ vary during the corrective process. Take the following second-order relaxation equation which results from neglecting the terms $\dot{\mathbf{W}}\mathbf{r}$ and $\dot{\mathbf{q}}\dot{\lambda}$:

$$\mathbf{K}\ddot{\mathbf{u}} + (\dot{\mathbf{K}} + \mathbf{W}\mathbf{K})\mathbf{u} = \mathbf{q}\ddot{\lambda} + \mathbf{W}\mathbf{q}\dot{\lambda}. \tag{22.14}$$

Treat this equation by the integrators

$$\dot{\mathbf{u}}^{(k+1)} - \dot{\mathbf{u}}^{(k)} = h\ddot{\mathbf{u}}^{(k)}, \qquad \dot{\lambda}^{(k+1)} - \dot{\lambda}^{(k)} = h\ddot{\lambda}^{(k)}, \tag{22.15}$$

and again determine h from

$$\dot{\mathbf{u}}^{(k+1)} \approx \mathbf{0}, \qquad \dot{\lambda}^{(k+1)} \approx 0. \tag{22.16}$$

Upon replacing $\mathbf{K}\dot{\mathbf{u}} - \mathbf{q}\dot{\lambda}$ by $\dot{\mathbf{r}}$ in the resulting equation, we obtain again the previous result. Therefore, all previous formulas remain valid for the variable-λ case provided that the assumptions made above hold. Note that the result is *independent of the constraint condition assumed for increment control*.

Secant Accelerators

The presence of rates such as $\dot{\mathbf{K}}$ and $\dot{\mathbf{r}}$ in the tangent accelerators gives rise to computational difficulties in complex nonlinear problems. More practical is to use secant-type (finite difference) information to obtain *secant accelerators*. The idea is to formally replace

$$\begin{aligned}
\dot{\mathbf{K}} &\to \Delta\mathbf{K}^{(k)} = \mathbf{K}^{(k+1)} - \mathbf{K}^{(k)},\\
\dot{\mathbf{u}} &\to \mathbf{d}^{(k)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)} = \Delta\mathbf{u}^{(k+1)} - \Delta\mathbf{u}^{(k)},\\
\dot{\mathbf{r}} &\to \mathbf{g}^{(k)} = \mathbf{r}^{(k+1)} - \mathbf{r}^{(k)},
\end{aligned} \tag{22.17}$$

in the tangent accelerators. This formal procedure, however, has the disadvantage that $\Delta\mathbf{K}$ is not only unwieldy to calculate in CNM but unavailable in MNM. Fortunately in the tangent accelerators $\dot{\mathbf{K}}$ appears always in the combination $\dot{\mathbf{K}}\dot{\mathbf{u}}$; consequently $\Delta\mathbf{K}$ is always postmultiplied by $\mathbf{d}$ in its secant counterparts. A convenient expression for $\Delta\mathbf{K}\,\mathbf{d}$ can be obtained from the Quasi-Newton formulas, which are multidimensional generalizations of the scalar secant method. Pertinent expressions for the Broyden family [21] are worked out in Fletcher [19], from where one gets

$$\dot{\mathbf{K}}\dot{\mathbf{u}} \to \Delta\mathbf{K}^{(k)}\,\mathbf{d}^{(k)} = \mathbf{g}^{(k)} - \mathbf{K}^{(k)}\mathbf{d}^{(k)}. \tag{22.18}$$

Substituting the secant expressions yields

$$\boxed{\bar{h} = \frac{\mathbf{g}^{(k)\,T}\,\mathbf{d}^{(k)}}{\mathbf{g}^{(k)\,T}\,\mathbf{W}^{(k)}\,\mathbf{d}^{(k)} + \mathbf{d}^{(k)\,T}\,(\mathbf{g}^{(k)} - \mathbf{K}^{(k)}\mathbf{d}^{(k)})}} \tag{22.19}$$

On replacing the appropriate $\mathbf{W}$, secant counterparts of h_{CNM}, h_{MNM} and h^*_{MNM} are obtained as

$$\bar{h}_{CNM} = \frac{1}{1+\nu}, \qquad \nu = \frac{\mathbf{d}^{(k)\,T}\,\mathbf{K}^{(k)}\mathbf{d}^{(k)}}{\mathbf{g}^{(k)\,T}\,\mathbf{d}^{(k)}}, \tag{22.20}$$

$$\bar{h}_{MNM} = \frac{\mathbf{g}^{(k)\,T}\,\mathbf{d}^{(k)}}{\mathbf{g}^{(k)\,T}\,\overline{\mathbf{F}}\mathbf{g}^{(k)} + \mathbf{d}^{(k)\,T}\,(\mathbf{g}^{(k)} - \mathbf{K}^{(k)}\mathbf{d}^{(k)})}. \tag{22.21}$$

$$\bar{h}^*_{MNM} = \frac{\mathbf{g}^{(k)\,T}\,\mathbf{d}^{(k)}}{\mathbf{g}^{(k)\,T}\,\overline{\mathbf{F}}\mathbf{g}^{(k)}}, \tag{22.22}$$

Remark 22.6. The important equation (22.18) is not sensitive to the choice of Quasi-Newton formula. In fact one always has

$$\mathbf{K}^{(k+1)}\mathbf{d}^{(k)} = (\mathbf{K}^{(k)} + \Delta\mathbf{K}^{(k)})\,\mathbf{d}^{(k)} = \mathbf{g}^{(k)}, \tag{22.23}$$

which is called the *Quasi-Newton condition* and serves as a point of departure for defining such formulas.

A Secant Accelerator Algorithm

For the computer implementation it is important to note that secant accelerators involve $\mathbf{u}^{(k+1)}$ and $\mathbf{r}^{(k+1)}$. Since these are not known until the stepsize h has been selected, the process is inherently iterative. The basic structure of a model algorithm follows.

1. Select an initial $\bar{h}$ equal to the final stepsize used in the previous iteration step; if the first step, set $\bar{h} = 1$.

2. Compute $\mathbf{u}^{(k+1)}$ and $\mathbf{r}^{(k+1)}$ based on this $\bar{h}$.

3. Calculate $\bar{h}$. If this value differs substantially from the assumed one (say by more than $\pm 50\%$) repeat steps 2 and 3 but no more than a fixed number of times (typically 1).

4. If a $\bar{h}$ has been accepted, use to calculate $\mathbf{u}^{(k+1)}$. Otherwise branch to a line search procedure.

Many stylistic variations are possible. For example, one might decide never to repeat steps 2–3. This is the same as extrapolating $\bar{h}$ from the previous-step information, which is economical but potentially dangerous.

Minimum Residual Accelerator

Accelerators may also be derived from the $\mathbf{r} \approx \mathbf{0}$ model, in which the residual is viewed as a function of the stepsize:

$$\mathbf{r}(h) = \mathbf{r}(\mathbf{u}^{(k)} + \mathbf{c}^{(k+1)} + h\mathbf{s}^{(k)}) \tag{22.24}$$

where $\mathbf{s}^{(k)}$ is defined by (65). Note that for variable λ vector $\mathbf{c}^{(k+1)} = \bar{\mathbf{v}}\Delta\lambda^{(k+1)}$ depends indirectly on h. The stepsize h is to be chosen so that a norm of $\mathbf{r}$ is minimized. Most commonly used is the scaled Euclidean norm:

$$r = \tfrac{1}{2}\mathbf{r}^T\,\mathbf{D}\mathbf{r}, \tag{22.25}$$

where $\mathbf{D}$ is a positive-definite scaling matrix (usually diagonal) A non-identity diagonal $\mathbf{D}$ is useful when the residual components have different physical units. The choice of $\mathbf{D} \equiv \mathbf{F} = \mathbf{K}^{-1}$ (if positive definite) is also of some interest because the residual norm becomes an energy norm.

The general approach to minimizing $r(h)$ is through a line-search method as described in the next subsection. But if the dependence of $\mathbf{r}$ on h is mildly nonlinear, an approximate minimizer can be obtained by assuming a *linear* dependence of $\mathbf{r}$ on h:

$$\mathbf{r}(h) \approx h\,\mathbf{r}(0) + (1-h)\mathbf{r}(1), \tag{22.26}$$

where $\mathbf{r}(0)$ and $\mathbf{r}(1)$ denote the residual vectors evaluated at $h = 0$ and $h = 1$, respectively; if $k > 0$ then $\mathbf{r}(0)$ should be available from the previous iteration. The minimizing condition $\partial r/\partial h = 0$ yields

$$\boxed{\hat{h} = \frac{\mathbf{r}(0)^T\mathbf{D}\mathbf{r}(0) - \mathbf{r}(1)^T\mathbf{D}\mathbf{r}(0)}{\mathbf{r}(0)^T\mathbf{D}\mathbf{r}(0) - 2\mathbf{r}(1)^T\mathbf{D}\mathbf{r}(0) + \mathbf{r}(1)^T\mathbf{D}\mathbf{r}(1)}} \tag{22.27}$$

Inasmuch as the assumption (22.26) is only strictly valid for linear response behavior, it is recommended to check whether

$$\boxed{r(\hat{h}) < r(1) \quad \text{and} \quad r(\hat{h}) < r(0)} \tag{22.28}$$

holds before accepting $\hat{h}$. If this condition is not verified, a line search may be called for.

Remark 22.7. Unlike the accelerators based on the zero-velocity model, the formula (22.27) depends on the corrective method only indirectly through $\mathbf{r}(1)$. This gives it an implementation edge over secant accelerators in the sense that the stepsize calculations are less method dependent. On the other hand, its performance is not usually as good as that of secant accelerators.

Remark 22.8. If $\mathbf{D}$ is the identity matrix, ξ is the residual magnitude ratio $\xi = \|\mathbf{r}(1)\|/\|\mathbf{r}(0)\| = \sqrt{\mathbf{r}(1)^T\mathbf{r}(1)/\mathbf{r}(0)^T\mathbf{r}(0)}$ and φ is the angle subtended by $\mathbf{r}(1)$ and $\mathbf{r}(0)$, the above formula can be presented as

$$\hat{h} = \frac{1 - \xi\cos\varphi}{1 - 2\xi\cos\varphi + \xi^2} \tag{22.29}$$

which admits of a geometric interpretation studied in Felippa [22].

Remark 22.9. Is it worthwhile accepting a non-unitary $\hat{h}$? If so, an extra residual evaluation at $\hat{h}$ would be required. A detailed analysis by Felippa [22] indicates that $\hat{h}$ should be accepted only if it is near 0.5 (underrelaxation) or near 2.0 (overrelaxation).

A Minimum Residual Accelerator Algorithm

The following model algorithm has some points in common with that described for a secant accelerator, but is more cautious. It is offered as an illustration of the kind of strategies that may be implemented in a general-purpose nonlinear analysis program.

1. Select initial stepsize $h = 1$. If the first iteration, calculate $\mathbf{r}(0)$.

2. Evaluate the residual $\mathbf{r}(1)$ and calculate $\hat{h}$ from (100). If the first iteration, accept $h = 1$ but record $\hat{h}$; else drop through.

3. Branch according to $\hat{h}$: (a) if $\hat{h}$ is outside a "reasonable range" (for example, 0.4 to 2.5) for two consecutive steps, branch to a line search procedure. (b) if outside the range for this step, accept $h = 1$ if $\hat{h} > 2.5$, else set $h = 0.5$, and proceed to next iteration. (c) if $0.75 \leq \hat{h} \leq 1.7$, accept $h = 1$ and proceed to next iteration. Else drop through.

4. Set $h = \hat{h}$ and evaluate $r(h)$. If condition (101) is verified, accept this h and go to the next iteration. Else branch to a line search procedure.

Line Search

A *line search* is a systematic procedure to find a stepsize h that approximately minimizes a residual-magnitude measure, which for definiteness shall be assumed to be (98). Vector $\mathbf{s}^{(k)}$ is called the *search direction*.

Line search procedures are highly developed in optimization work, where they are essential components of general-purpose function minimization programs: a line search is carried out at each iteration. On the other hand, their role in nonlinear equation solving is secondary. Because line search is fairly expensive in terms of residual evaluations, it is invoked only when a solution procedure runs into severe difficulties.

The acceptance tests stated below involve the directional derivative $g = \partial r(h)/\partial h$ along $\mathbf{s}$. If $\mathbf{D}$ is constant,

$$g(h) = \frac{\partial r}{\partial h} = \mathbf{r}^T\mathbf{D}\frac{\partial \mathbf{r}}{\partial h} = \mathbf{r}^T\mathbf{D}\mathbf{K}(\mathbf{s}^{(k)} + \bar{\mathbf{v}}\frac{\partial \lambda}{\partial h}) \tag{22.30}$$

The last expression is computationally cumbersome because it involves the tangent stiffness matrix. More practical is to use finite differences to estimate the factor $\partial\mathbf{r}/\partial h$.

Line search algorithms "backtrack" h in a systematic fashion until an *acceptance condition* is verified. A widely used acceptance condition in optimization programs of the the early 1970s is the Goldstein-Armijo (GA) rule which may be stated as

$$0 < -\eta_1 h g(h) \leq r(0) - r(h) \leq -\eta_2 h g(h), \tag{22.31}$$

where η_1 and η_2 are scalars that satisfy $0 < \eta_1 \leq \eta_2 < 1$; typical values being $\eta_1 \approx 0.1$ and $\eta_2 \approx 0.9$. The upper and lower bounds in (104) insure that h is neither "too large" nor "too small". In more modern optimization work [23], the GA rule is replaced by slope conditions of the form

$$|g(h)| \approx \left|\frac{g(h) - g(h_1)}{h - h_1}\right| \leq -\eta g(0), \tag{22.32}$$

$$r(0) - r(h) \geq -\xi h g(0). \tag{22.33}$$

where $0 \leq \eta < 1$ and $0 < \xi \leq 0.5$, $\xi \leq \eta$, and $0 \leq h_1 < h$. A value of $\eta \approx 0.8$, which gives a not very restrictive line search, can be recommended.

A Line Search Algorithm

There are many variations on this theme and readers interested in additional details are referred to the abundant literature on practical optimization methods. The following algorithm is meant only to illustrate the basic procedural steps.

1. Estimate $g(0)$ by finite differences, for example

$$g(0) \approx \mathbf{r}(0)^T \mathbf{D} \frac{(\mathbf{r}(0.1) - \mathbf{r}(0))}{0.1} \tag{22.34}$$

   If $g(0) \geq 0$ exit with $h = 0$, which should trigger a refactoring. Else set $h_1 = 0$, $h_2 = 10$ (say), $h = 1$.

2. Evaluate $r(h)$.

3. Test for (22.32). If not satisfied, compute $\hat{h}$ by restricted interpolation from data at h_1 and h; set $h_2 = h$, $h = \hat{h}$, and return to 2.

4. Test for (22.31). If satisfied, exit. If not, compute $\hat{h}$ by restricted interpolation from data at h_1 and h; reset $h_1 = h$, $h = \hat{h}$ and go to 2.

If the exit $h < 0.1$ (say), the iteration step should be abandoned and the stiffness matrix refactored at $h = 0$.

As stressed, a line search procedure should be initiated only when erratic or divergent behavior is suspected. This being the case, the initial steps of the search procedure should take advantage of previously computed information such as $\mathbf{r}(0)$ and $\mathbf{r}(1)$. Similarly (22.27), suitably safeguarded, can be utilized for the interpolation process in steps 3 and 4; its key advantage being that it does not require g derivatives.

§22.4. References

[22.1] OTTER, J. R. H., "Computations for Prestressed Concrete Reactor Pressure Vessels Using Dynamic Relaxation," *Nucl. Struct. Engng.*, **1**, 61–75, 1965.

[22.2] UNDERWOOD, P. G., "Dynamic Relaxation — A Review," Ch. 5 in *Computational Methods for Transient Dynamic Analysis*, (eds. T. Belytschko and T. J. R. Hughes), North-Holland, Amsterdam, 1983.

[22.3] PARK, K. C., "A Family of Solution Algorithms for Nonlinear Structural Analysis Based on the Relaxation Equations," *Int. J. Num. Meth. Engng.*, **18**, 1337–1347, 1982.

[22.4] RIKS, E., "The Application of Newton's Method to the Problem of Elastic Stability,", *J. Appl. Mech.*, **39**, 1060–1065, 1972.

[22.5] CRISFIELD, M. A., "Incremental/Iterative Solution Procedures for Nonlinear Structural Analysis," in *Numerical Methods for Nonlinear Problems, Vol. 1*, (eds. C. Taylor, E. Hinton and D. R. J. Owen), Pineridge Press, Swansea, U. K., 261–290, 1980.

[22.6] CRISFIELD, M. A., "An Incremental-Iterative Algorithm that Handles Snap-Through," *Computer & Structures*, **13**, 55–62, 1981.

[22.7] KUBÍČEK, M. and HLAVÁČEK, V., *Numerical Solution of Nonlinear Boundary Value Problems with Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[22.8] BAKHVALOV, N. S., *Numerical Methods*, Mir Publishers, Moscow, 1975.

[22.9] ORTEGA, J. M. and RHEINBOLDT, W. C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[22.10] BOGGS, P. T., "The Solution of Nonlinear Systems of Equations by *A*-Stable Integration Techniques," *SIAM J. Num. Anal.*, **8**, 767–785, 1971.

[22.11] BRANIN, F. H. and HOO, S. K., "A Method for Finding Multiple Extrema of a Function of *n* Variables," in *Numerical Methods for Nonlinear Optimization*, (ed. F. A. Lootsma), Academic Press, London, 1972.

[22.12] KELLER, H. B., "Global Homotopies and Newton Methods," in *Recent Advances in Numerical Analysis*, (eds. C. de Boor and G. H. Golub), Academic Press, New York, 1978.

[22.13] WACKER, Hj. (ed.), *Continuation Methods*, Academic Press, NY, 1978.

[22.14] ALLGOWER, E. L., "A Survey of Homotopy Methods for Smooth Mappings," in *Numerical Solution of Nonlinear Equations*, (eds. E. L. Allgower *et al.*), Lecture Notes in Mathematics 878, Springer-Verlag, Berlin, 1981.

[22.15] PADOVAN, J., "Self-Adaptive Predictor-Corrector Algorithm for Static Nonlinear Structural Analysis," Report NASA CR-165410 to Lewis Research Center, The University of Akron, Akron, Ohio, 1981.

[22.16] BATHE, K. J. and DVORKIN, E., "On the Automatic Solution of Nonlinear Finite Element Equations," *Computer & Structures*, **17**, 871–879, 1983.

[22.17] FELIPPA, C. A., "Finite Element Analysis of Three-Dimensional Cable Structures," in *Computational Methods in Nonlinear Mechanics*, (ed. J. T. Oden *et al.*), The Texas Institute for Computational Mechanics, University of Texas, Austin, Texas, 311–324, 1974.

[22.18] TRAUB, J. F., *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[22.19] FELIPPA, C. A., "Dynamic Relaxation and Quasi-Newton Methods," in *Numerical Methods for Nonlinear Problems 2*, (eds C. Taylor, E. Hinton, D. R. J. Owen & E. Oñate), Pineridge Press, Swansea, U. K., 27–38, 1984.

[22.20] BASU, A. K., "New Light on the Nayak Alpha Technique," *Int. J. Num. Meth. Engng.*, **6**, 152–153, 1973.

[22.21] BROYDEN, C. G., "Quasi-Newton Methods and their Application to Function Minimization," *Maths. Comput.*, **21**, 368–381, 1967.

[22.22] FELIPPA, C. A., "Procedures for Computer Analysis of Large Nonlinear Structural Systems," in *Large Engineering Systems*, (ed. A. Wexler), Pergamon Press, Oxford, 60–101, 1977.

[22.23] GILL, P. E., MURRAY, W. and WRIGHT, M. H., *Practical Optimization*, Academic Press, London, 1981.