19

Pseudo-Force Incremental Methods

TABLE OF CONTENTS

The disadvantages of purely incremental methods in terms of solution "drift" and high computational expense were recognized as applications of nonlinear finite element analysis expanded to cover wider classes of problems. Many techniques aimed at avoiding these difficulties were proposed and tested during the late 1960s and early 1970s. From the mass of experience accumulated during this period, two principal strategies emerged.

For *path-independent, smooth* nonlinearities typified by finite deflection and nonlinear-elastic behavior, the power of Newton-like corrective methods was eventually recognized. In such problems, incremental methods were relegated to the secondary role of *predictors* for starting a corrective process.

For path-dependent material nonlinearities, however, purely incremental methods have remained important because of the reasons noted in Chapter 18. These methods underwent modifications aimed primarily at reducing the computational expense while retaining the advantages of numerical stability, implementation simplicity and physical transparency. Unfortunately these goals, being largely contradictory, can only be met half-way. The most successful attempt in this direction has been the development of pseudo-force incremental methods, which are covered below.

§19.1. Pseudo Force Formulation

In the pseudo-force reformulation of incremental methods the pervasive role that the tangent stiffness matrix plays in the methods discussed in Chapters 17-18 is relaxed. Instead the deviation from a "reference linear response" is collected in a pseudo-force vector. This approach allows a *reference stiffness matrix* to be reused over many incremental steps. Since this avoids having to repeatedly assemble and factor the tangent stiffness matrix, the gain in speed *per step* over the conventional incremental methods may be very substantial in two- and three-dimensional problems. There is no free lunch, however, for the speed gain is counterbalanced by two disadvantages:

1. Pronounced accuracy loss as nonlinearities become severe, which may force extremely small increments to be taken.

2. Increasing danger of numerical instability, especially in "hardening" portions of the response.

The pseudo-force method can be explained more conveniently by starting from the "force balance" form (4.9) of the residual:

$$\mathbf{r} = \mathbf{f} - \mathbf{p} = \mathbf{0}, \tag{19.1}$$

where $\mathbf{f}$ and $\mathbf{p}$ are the internal and external (applied) force vectors, respectively. Furthermore for simplicity we shall assume the *separable* form (4.10), that is $\mathbf{f}(\mathbf{u}) = \mathbf{p}(\lambda)$.

Decompose the internal residual force $\mathbf{f}$ as follows:

$$\mathbf{f} = \mathbf{K}_{ref}(\mathbf{u} - \mathbf{u}_{ref}) + \mathbf{n}, \tag{19.2}$$

where $\mathbf{K}_{ref}$ is a nonsingular *reference stiffness matrix* that is kept fixed as long as possible and $\mathbf{u}_{ref}$ is the state at $\lambda = \lambda_{ref}$ (usually 0). The deviation

$$\mathbf{n} = \mathbf{f} - \mathbf{K}_{ref}(\mathbf{u} - \mathbf{u}_{ref}), \tag{19.3}$$

is called the *pseudo-force* vector. Inserting (19.2) in the residual expression (19.1) we get

$$\mathbf{r} = \mathbf{K}_{ref}(\mathbf{u} - \mathbf{u}_{ref}) + \mathbf{n} - \mathbf{p} = \mathbf{0}. \tag{19.4}$$

Differentiating (19.4) with respect to λ yields the rate form

$$\mathbf{r}' = \mathbf{K}_{ref}\,\mathbf{u}' + \mathbf{n}' - \mathbf{q} = \mathbf{0}, \tag{19.5}$$

where $\mathbf{q} = \partial\mathbf{p}/\partial\lambda$ is the incremental load vector. Since $\mathbf{K}_{ref}$ is assumed nonsingular, solving for $\mathbf{u}'$ yields

$$\mathbf{u}' = \mathbf{K}_{ref}^{-1}(\mathbf{q} - \mathbf{n}') = \mathbf{v}^f, \tag{19.6}$$

Vector $\mathbf{v}^f$ is called the *pseudo incremental velocity*. It plays a similar role to that of $\mathbf{v} = \mathbf{K}^{-1}\mathbf{q}$ in the conventional incremental methods discussed in Chapters 14 and 15.

Remark 19.1. The pseudo-force vector $\mathbf{n}$ may be viewed as a "force deviation from the linear reference response" $\mathbf{K}_{ref}\,(\mathbf{u} - \mathbf{u}_{ref})$. Now if the structural response is linear so that

$$\mathbf{K}_0\mathbf{u} = \mathbf{p}, \qquad (\mathbf{K}_0 \text{ constant}), \tag{19.7}$$

choosing $\mathbf{K}_{ref}$ equal to the linear stiffness $\mathbf{K}_0$ and $\mathbf{u}_{ref} = \mathbf{0}$ gives $\mathbf{n} = \mathbf{0}$. For this reason many authors call $\mathbf{n}$ the *nonlinear force vector*.

Remark 19.2. Differentiating (19.3) with respect to $\mathbf{u}$ gives, for a separable residual,

$$\frac{\partial\mathbf{r}}{\partial\mathbf{u}} = \mathbf{K} = \mathbf{K}_{ref} + \mathbf{K}_{non}, \qquad \mathbf{K}_{non} = \frac{\partial\mathbf{n}}{\partial\mathbf{u}}. \tag{19.8}$$

If $\mathbf{K}_{ref}$ is kept equal to the linear stiffness throughout, $\mathbf{K}_{non}$ is called the *nonlinear stiffness matrix*.

§19.2. Computing the Reference Stiffness and Internal Force

A very common choice for $\mathbf{K}_{ref}$, though far from the only one, is the stiffness $\mathbf{K}_0$ at the start $\lambda = 0$ of the response-calculation stage. Choosing $\mathbf{u}_{ref} = \mathbf{u}_0$ accordingly, (19.3) becomes

$$\mathbf{r} = \mathbf{K}_0(\mathbf{u} - \mathbf{u}_0) + \mathbf{n} - \mathbf{p} = \mathbf{0}. \tag{19.9}$$

Response calculation procedures based on the choice (19.9) are sometimes called the *initial stiffness method*.

Another common strategy is to start with the initial stiffness and continue with as many incremental steps as possible, resetting $\mathbf{K}_{ref} = \mathbf{K}_n$ if numerical stability or accuracy problems are detected at the $(n+1)^{th}$ step. Still a third approach is to keep $\mathbf{K}_{ref}$ for a preassigned number of incremental steps, say m, and update $\mathbf{K}_{ref}$ at steps $0, m, 2m, \ldots$, unless numerical stability or accuracy problems are encountered.

The effectiveness of pseudo-force methods in finite element programming depends largely on the ability to compute the internal force vector $\mathbf{f}$ directly on an *element-by-element* basis. Recall, for example, that in the core-congruential formulation of the Total Lagrangian (TL) description,

$$\mathbf{f} = \int_{V_0} \mathbf{G}^T s_i\,\mathbf{b}_i\,dV_0, \tag{19.10}$$

where V_0 is the reference volume, $\mathbf{G}$ the transformation between displacement gradients and physical degrees of freedom, s_i are PK stresses in the current configuration, and $\mathbf{b}_i$ are the vectors defined in §8.5. Note that expression does not explicitly involve material properties, and is consequently applicable to problems with material nonlinearities. More precisely: the only requirement for using (19.9) is the availabilibity of the stresses s_i in the current configuration whereas the procedure by which such stresses are obtained is irrelevant.

Remark 19.3. In finite element work the synonyms *initial force method*, *initial stress method* and *initial strain method* have been associated with restricted versions of what we call here pseudo-force methods. These names focus attention on various physical interpretations of the calculation of the $\mathbf{f}$ term. A heated controversy as to which version was the best took place in the late sixties; from current perspective such arguments have only historical interest.

§19.3. Integration of Pseudo-force Rate Equation

In this subsection we assume that the calculation of $\mathbf{f}$, given the necessary ingredients to apply (19.5), is more practical than that of $\mathbf{f}'$. An estimation of this rate is done through finite difference approximations. This leads to very simple and fast implementations at the cost of numerical reliability and accuracy. A more expensive but reliable alternative technique for evaluating $\mathbf{f}'$ is discussed later.

§19.3.1. Forward Euler Integration under Load Control

The simplest incremental algorithm results on treating (19.5) by the forward Euler method with a backward-difference estimation of $\mathbf{f}'$ and assuming that the increments of λ are prescribed:

$$\boxed{\begin{aligned}
\mathbf{n}_n &= \mathbf{f}_n - \mathbf{K}_{ref}(\mathbf{u}_n - \mathbf{u}_{ref}),\\
\mathbf{n}'_n &= \frac{\mathbf{n}_n - \mathbf{n}_{n-1}}{\Delta\lambda_{n-1}},\\
\Delta\mathbf{u}_n &= \mathbf{K}_{ref}^{-1}\,(\mathbf{q}_n - \mathbf{n}'_n)\,\Delta\lambda_n = \mathbf{v}_n^f\,\Delta\lambda_n,\\
\mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta\mathbf{u}_n, \quad \lambda_{n+1} = \lambda_n + \Delta\lambda_n.
\end{aligned}} \tag{19.11}$$

where as usual we denote

$$\mathbf{f}_n \equiv \mathbf{f}(\mathbf{u}_n), \qquad \mathbf{n}_n = \mathbf{n}(\mathbf{u}_n), \qquad \text{etc.}$$

The pseudo incremental velocity $\mathbf{v}_n^f = \mathbf{K}_{ref}^{-1}\,(\mathbf{q}_n - \mathbf{n}'_n)$ plays the role of the incremental velocity vector, as can be seen by comparing the advancing equations (17.7) for purely incremental methods.

The scheme (19.11) applies if $n \geq 1$ and as long as $\mathbf{K}_{ref}$ and $\mathbf{u}_{ref}$ are kept fixed. At the start $n = 0$, $\mathbf{n}_{n-1} = \mathbf{n}_{-1}$ is not known. But if $\mathbf{K}_{ref} = \mathbf{K}_0$ and $\mathbf{u}_{ref} = \mathbf{u}_0$, $\mathbf{n}' = \mathbf{0}$. The same condition is applicable when $\mathbf{K}_{ref}$ and $\mathbf{u}_{ref}$ are reset, if one chooses the tangent stiffness at $\mathbf{u}_{ref}$ as reference stiffness.

This advancing scheme has poor accuracy characteristics unless nonlinearities are mild (say within $\pm 20\%$ of the reference response). There are three ways of improving accuracy: pseudo-force extrapolation, iteration and resetting the reference stiffness. The first two are described in subsections below.

Example 19.1. Solve the residual equation

$$r = 5u - u^3 - \lambda = 0, \qquad \text{for} \quad \lambda = 4, \tag{19.12}$$

using the pseudo-force method (19.11) with two $\Delta\lambda$ increments of 2.0, $K_{ref} = K_0 = 5$ and $u_{ref} = u_0 = 0$. The exact solution on the fundamental path is $v(\lambda) = v(4) = 1$. Since $q = -\partial r/\partial\lambda = 1$, and $n = 5u - u^3 - 5u = -u^3$, the rate form is $r' = 5u' - (u^3)' - 1 = 0$.

The first increment, with $\lambda_0 = 2$ specified, is

$$n_0' = 0, \quad \Delta u_0 = K_{ref}^{-1}(q_0 - n_0')\Delta\lambda_0 = 0.4, \quad u_1 = u_0 + \Delta u_0 = 0.4, \quad \lambda_1 = \lambda_0 + \Delta\lambda_0 = 2.$$

The second increment, with $\Delta\lambda_1 = 2$ again, is

$$n_1' = -(u_1^3 - u_0^3)/\Delta\lambda_0 = -0.032, \quad \Delta u_1 = K_{ref}^{-1}(q_1 - n_1')\Delta\lambda_0 = 0.4128,$$
$$u_2 = u_1 + \Delta u_1 = 0.8128, \quad \lambda_2 = \lambda_1 + \Delta\lambda_1 = 4.$$

Repeating these computations with 4, 8 and 17 equal increments of λ gives 0.8461, 0.8841 and 0.9190, respectively. As can be observed the accuracy attained is low. Table 19.1 compares these values with those obtained with other methods.

§19.3.2. Pseudo-Force Extrapolation

Accuracy improves if the first of (19.10) is replaced by a central difference estimator:

$$\mathbf{n}_n' = \frac{\mathbf{n}_{n+1} - \mathbf{n}_{n-1}}{\Delta\lambda_n + \Delta\lambda_{n-1}} \tag{19.13}$$

Since $\mathbf{n}_{n+1}$ is not known, it has to be predicted by extrapolation. The simplest extrapolator is

$$\mathbf{n}^{n+1} \approx \mathbf{n}_{n+1}^P = \mathbf{n}(\mathbf{u}_{n+1}^P), \quad \text{with} \quad \mathbf{u}_{n+1}^P = \mathbf{u}_n + (\mathbf{u}_n - \mathbf{u}_{n-1})\frac{\Delta\lambda_n}{\Delta\lambda_{n-1}}. \tag{19.14}$$

For constant $\Delta\lambda$, the predicted $\mathbf{u}_{n+1}^P$ is simply $2\mathbf{u}_n - \mathbf{u}_{n-1}$. The advancing algorithm is identical to (19.11) with the second equation replaced by (19.13) and (19.14).

The result of applying this technique to the example equation (19.12) with fixed λ increments is presented in Table 17.1. The accuracy obtained now is similar to that of the conventional purely incremental method with Forward Euler. Also given there are the results of using the midpoint rule (17.8), which as can be seen delivers higher accuracy.

§19.3.3. Iterative Improvement

Another way to improve accuracy while avoiding the reset of $\mathbf{K}_{ref}$ is to iterate on $\mathbf{u}_{n+1}$ while keeping $\Delta\lambda_n$ fixed. To derive an iteration scheme, write the residual form (19.4) at $n+1$ and n, subtract, and solve for $\Delta\mathbf{u}_n$:

$$\Delta\mathbf{u}_n = \mathbf{K}_{ref}^{-1}\,(\mathbf{n}_{n+1} - \mathbf{n}_n) - \mathbf{q}_n\,\Delta\lambda_n \tag{19.15}$$

Let k be an iteration step index and $\mathbf{u}_{n+1}^1$ be the value obtained from the increment equation. The resulting iterative scheme is

$$\left.\begin{aligned} \Delta\mathbf{u}_n^k &= \mathbf{K}_{ref}^{-1}(\mathbf{n}_{n+1}^k - \mathbf{n}_n) - \mathbf{q}_n\,\Delta\lambda_n, \\ \mathbf{u}_{n+1}^{k+1} &= \mathbf{u}_n + \Delta\mathbf{u}_n^k. \end{aligned}\right\} \; k = 1, \ldots \tag{19.16}$$

Table 19.1 Computed Incremental Solutions for (19.12)
Load Control, Equal λ Increments

Steps	PFI-FE	PFI-FE-X	PI-FE	PI-MR
1	0.8000	0.8000	0.8000	0.8850
2	0.8128	0.8512	0.8425	0.9360
4	0.8461	0.8835	0.8884	0.9760
8	0.8841	0.9139	0.9276	0.9888
17	0.9190	0.9407	0.9565	0.9964
32	0.9470	0.9617	0.9755	0.9990
64	0.9675	0.9768	0.9868	0.9997
128	0.9812	0.9867	0.9931	0.9999

PFI-FE: Pseudo-force incremental with Forward Euler (19.11)
PFI-FE-X: PFI-FE with extrapolation (19.13)-(19.14) for $\mathbf{n}'_n$
PI-FE: Purely incremental with Forward Euler (17.7)
PI-FE: Purely incremental with Midpoint Rule (17.8)

Using the fact that $\mathbf{q}$ is independent of $\mathbf{u}$, this can be rewritten in the "subincremental" form

$$\mathbf{u}_{n+1}^{k+1} = \mathbf{u}_{n+1}^{k} - \mathbf{K}_{ref}^{-1}\mathbf{r}_{n+1}^{k}. \tag{19.17}$$

If $\mathbf{K}_{ref} \equiv \mathbf{K}_n$, this is precisely the modified Newton-Raphson (MNR) method with unit steplength. This shows that the *the iterated pseudo-force incremental method is a MNR method with an arbitrary selection of reference stiffness*. The properties of these methods are investigated in Chapters that deal with Newton-Raphson corrective methods.

§19.4. Numerical Stability

The purely incremental tangent-stiffness methods studied in Chapters 17-18 are highly stable if some mild precautions are heeded. On the other hand, pseudo-force methods are much less robust. For a single degree of freedom, the homogeneous model equation corresponding to (17.10) is

$$v' = K_{ref}^{-1} n' = K_{ref}^{-1} \frac{\partial n}{\partial v} v' = A v', \tag{19.18}$$

where $A = K_{non}/K_{ref}$. Unlike (17.10) this is no longer a differential equation in v, v' but a *difference equation* in v', with A as amplification number. The iteration process to solve this equation is stable if

$$|A| < 1. \tag{19.19}$$

This condition is *independent of the stepsize* $\Delta\lambda$. It is seen that the key for numerical stability is that the *reference stiffness "dominates" the nonlinear stiffness* in the sense (19.20). That is, from that standpoint it is better to overestimate K_{ref}.

The generalization of (19.19) to N degrees of freedom is

$$\mathbf{u}' = \mathbf{K}_{ref}^{-1}\mathbf{u}' = \mathbf{K}_{ref}^{-1}\frac{\partial \mathbf{n}}{\partial \mathbf{u}}\mathbf{u}' = \mathbf{K}_{ref}^{-1}\mathbf{K}_{non} = \mathbf{A}\mathbf{u}'. \tag{19.20}$$

Stability is controlled by the $N \times N$ amplification matrix

$$\mathbf{A} = \mathbf{K}_{ref}^{-1} \frac{\partial \mathbf{n}'}{\partial \mathbf{u}} = \mathbf{K}_{ref}^{-1} \mathbf{K}_{non}. \tag{19.21}$$

Assume that this amplification matrix $\mathbf{A}$ has eigenvalues $\mu_i, i = 1, \ldots N$, and let $\mu = \max_i |\mu_i|$. The condition for numerical stability is

$$\mu < 1. \tag{19.22}$$

This condition holds *regardless of the stepsize* $\Delta\lambda$. Likewise, the iteration (19.17) converges only if the condition (19.22) holds.

A practical estimator for this eigenvalue is

$$\mu \approx \frac{|\mathbf{K}_{ref}^{-1}(\mathbf{n}_n - \mathbf{n}_{n-1})|}{|\mathbf{u}_n - \mathbf{u}_{n-1}|}. \tag{19.23}$$

Remark 19.4. This result is another aspect of the close relationship between pseudo-force incremental methods and modified-Newton corrective methods. If nonlinearities are substantial, the method diverges regardless of the increment length used.

Remark 19.5. It is beneficial from a stability standpoint to have $\mathbf{K}_{ref}$ "dominate" $\mathbf{K}_{non}$. This happens in *softening* structures when the elastic stiffness is selected as $\mathbf{K}_{ref}$, and explains the success of the method in plasticity analysis. On the other hand, if the structure hardens as λ increases (examples: cable and pneumatic structures), the stability condition is easily violated.

Remark 19.6. The simplest cure to numerical instability is to recompute the reference stiffness. Another (as yet unexplored) possibility is to correct $\mathbf{K}_{ref}$ with a rank-one matrix.

§19.5. Accuracy Control

Accuracy control may be effected as in the case of conventional incremental methods if one substitutes $\mathbf{v}^f$ for $\mathbf{v}$.

§19.6. Secant Estimation of $\mathbf{n}'$

The finite difference estimators for $\mathbf{n}'$ described in §17.3 are easy to implement but decidedly suffer from a lack of robustness unless the problem is only mildly nonlinear. An alternative estimate of $\mathbf{n}'$ can be obtained through the following "secant approximation" technique. This estimate is computationally slower but more reliable. Recall that

$$\mathbf{n} = \mathbf{p} - \mathbf{K}_{ref}(\mathbf{u} - \mathbf{u}_{ref}). \tag{19.24}$$

$$\mathbf{n}' = \mathbf{q} - \mathbf{K}_{ref}\mathbf{u}' = \mathbf{q} - \mathbf{K}_{ref}(\mathbf{K}^{-1}\mathbf{q}) = (\mathbf{I} - \mathbf{K}_{ref}\mathbf{K}^{-1})\mathbf{q}. \tag{19.25}$$

We now replace the exact inverse of the tangent matrix by a *secant approximation*:

$$\mathbf{K}^{-1} \approx \mathbf{F}_s. \tag{19.26}$$

$\mathbf{F}_s$ is a low-rank correction (typically rank one or two) of $\mathbf{F}_{ref} = \mathbf{K}_{ref}^{-1}$ that is constructed on the basis of the following increments:

$$\Delta\mathbf{u}_s = \Delta\mathbf{u}_{n-1} = \mathbf{u}_n - \mathbf{u}_{n-1}, \qquad \Delta\mathbf{r}_s = \mathbf{r}(\mathbf{u}_n, \lambda_n) - \mathbf{r}(\mathbf{u}_{n-1}, \lambda_n), \tag{19.27}$$

Note that the residual $\Delta\mathbf{r}_s$ is computed *by holding* λ *constant and equal to* λ_n and is *not* $\mathbf{r}_n - \mathbf{r}_{n-1}$. Two choices for the inverse stiffness secant approximant are the Davidon-Fletcher-Powell (DFP) rank-two update formula

$$\mathbf{F}_s = \mathbf{F}_{ref} + \frac{\Delta\mathbf{u}_s \Delta\mathbf{u}_s^T}{\Delta\mathbf{u}_s^T \Delta\mathbf{r}_s} - \frac{\mathbf{F}_{ref}\Delta\mathbf{r}_s \Delta\mathbf{r}_s^T \mathbf{F}_{ref}}{\Delta\mathbf{r}_s^T \mathbf{F}_{ref}\Delta\mathbf{r}_s}, \tag{19.28}$$

and Davidon's rank-one update formula

$$\mathbf{F}_s = \mathbf{F}_{ref} + \frac{(\Delta\mathbf{u}_s - \mathbf{F}_{ref}\Delta\mathbf{r}_s)(\Delta\mathbf{u}_s - \mathbf{F}_{ref}\Delta\mathbf{r}_s)^T}{(\Delta\mathbf{u}_s - \mathbf{F}_{ref}\Delta\mathbf{r}_s)^T \Delta\mathbf{r}_s}. \tag{19.29}$$

These formulas are collectively called *Quasi-Newton* updates in the numerical analysis literature, although to be more precise what we have shown above is just the first member of such updates.

§19.7. General Increment Control

The preceding developments assume that $\Delta\lambda_n$ is prescribed. But we can readily extend the pseudo-force technique to general increment control by following the procedures discussed in Chapter 18. For that we must replace $\mathbf{v}_n$ by

$$\mathbf{v}_n^f = \mathbf{K}_{ref}^{-1}\,(\mathbf{q}_n - \dot{\mathbf{n}}_n). \tag{19.30}$$

where the superposed dot denotes derivative respect to the pseudotime parameter t pertinent to the increment strategy chosen. Of particular importance is *arclength control*, in which t becomes the arclength s. Criteria for stability and accuracy are readily converted to this case.