Krylov Subspace Accelerated Newton Algorithm: Application to Dynamic Progressive Collapse Simulation of Frames

Michael H. Scott[1] and Gregory L. Fenves[2]

Abstract: An accelerated Newton algorithm based on Krylov subspaces is applied to solving nonlinear equations of structural equilibrium. The algorithm uses a low-rank least-squares analysis to advance the search for equilibrium at the degrees of freedom (DOFs) where the largest changes in structural state occur; then it corrects for smaller changes at the remaining DOFs using a modified Newton computation. The algorithm is suited to simulating the dynamic progressive collapse analysis of frames where yielding and local collapse mechanisms form at a small number of DOFs while the state of the remaining structural components is relatively linear. In addition, the algorithm is able to resolve erroneous search directions that arise from approximation errors in the tangent stiffness matrix. Numerical examples indicate that the Krylov subspace algorithm has a larger radius of convergence and requires fewer matrix factorizations than Newton-Raphson in the dynamic progressive collapse simulation of reinforced concrete and steel frames.

DOI: 10.1061/(ASCE)ST.1943-541X.0000143

CE Database subject headings: Algorithms; Failures; Nonlinear analysis; Progressive collapse; Steel frames.

Author keywords: Algorithms; Collapse; Nonlinear analysis; Progressive failure.

Introduction

Simulating the response of a structural system to extreme events poses significant computational challenges particularly when there are drastic changes in the load-resisting characteristics of the structural components. A problem that has been the focus of recent research is the dynamic response of a structure to sudden failure of one or more members, which will cause redistribution of resisting forces and in turn lead to possible failure of adjacent members. Dynamic progressive collapse can be difficult to simulate using implicit solution methods because of yielding, fracture, geometric instability, impact forces, and joint failures taking place simultaneously in a single time step. Kaewkulchai and Williamson (2004) showed that a static analysis may not provide conservative estimates of the collapse potential for a frame and that incorporating dynamic effects leads to large increases in critical damage parameters such as plastic rotation. Menchel et al. (2009) assessed various progressive collapse simulation techniques by comparison with large displacement plastic hinge finite-element analysis. Experimental and analytical evaluation of progressive collapse has been carried out by Sasani and Sagiroglu (2008) in order to determine how well predicted changes in load distribution compare to measured structural response after removal of vertical load carrying members.

The standard simulation approach for dynamic progressive collapse, as well as other nonlinear structural dynamic problems, is to time discretize the governing equations by Newmark time integration then solve them via the Newton-Raphson algorithm (Bathe 1996; Zienkiewicz and Taylor 2005). Although its local rate of convergence is quadratic, a small time step that must be used to ensure the Newton-Raphson algorithm will converge when there are multiple nonlinear events occurring simultaneously. Furthermore, nonlinear response typically occurs at a small fraction of the structural degrees of freedom (DOFs) while the state of the remaining DOFs is relatively unchanged. This can lead to excessive computations since the stiffness of the entire system must be recomputed at each iteration of a time step in order to update the equilibrium search directions at all DOFs. When using direct equation solvers, the continual formation and factorization of the stiffness matrix, or Jacobian, to become a bottleneck during analyses of large structural systems.

The modified Newton method has a lower computational cost per iteration than Newton-Raphson; however, its local rate of convergence is linear. The stiffness matrix at the first iteration of a time step is held constant over the time step, making repeated use of the matrix factorization. When one or more nonlinear events occur during a simulation time step, a significant discrepancy between the state of the structure and that represented by the stiffness matrix at the start of the time step arises, resulting in possible nonconvergence due to poor search directions that lead the iterations away from equilibrium. For both Newton-Raphson and modified Newton, line search techniques (Crisfield 1991) can improve the search directions when the stiffness matrix is positive definite, which may not always be the case with degrading materials and large displacement analysis.

Quasi-Newton methods seek a compromise between the Newton-Raphson and modified Newton algorithms by modifying the tangent stiffness matrix, or its factorization, with low-rank updates during the search for equilibrium, resulting in a superlinear rate of convergence. The rank-two BFGS quasi-Newton procedure (Broyden 1970; Fletcher 1970; Goldfarb 1970; Shanno 1970) is appropriate for the symmetric positive definite systems

[1]Assistant Professor, School of Civil and Construction Engineering, Oregon State Univ., Corvallis, OR 97331 (corresponding author). E-mail: michael.scott@oregonstate.edu

[2]Dean, Cockrell School of Engineering, Univ. of Texas, Austin, TX 78712. E-mail: dean@engr.utexas.edu

Note. This manuscript was submitted on September 1, 2008; approved on November 2, 2009; published online on November 4, 2009. Discussion period open until October 1, 2010; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Structural Engineering***, Vol. 136, No. 5, May 1, 2010. ©ASCE, ISSN 0733-9445/2010/5-473–480/$25.00.

typically encountered in structural mechanics. For nonsymmetric systems, such as those that arise from nonassociative plasticity, the rank-one procedure of Broyden (1967) is more appropriate. The main drawback to quasi-Newton methods is that the updates of the tangent stiffness matrix tend to increase the matrix bandwidth while driving the iterations away from equilibrium (Crisfield 1991).

Accelerated Newton methods also seek a balance between the Newton-Raphson and modified Newton algorithms. Rather than modify the tangent stiffness matrix, as in quasi-Newton methods, accelerated Newton algorithms use matrix-vector operations to increase the convergence rate of the modified Newton algorithm. Crisfield (1984) developed a secant-based accelerated Newton algorithm based on the BFGS procedure of Matthies and Strang (1979) and demonstrated this algorithm to be more efficient than the modified Newton algorithm for material and geometric nonlinear problems in structural mechanics.

Carlson and Miller (1998) developed an accelerated Newton algorithm for applications in gradient weighted moving finite elements. In these applications, it is computationally inefficient to solve the governing equations by conventional Newton algorithms when sharp fronts develop in the finite-element mesh. The Jacobian matrix changes significantly at only a few DOFs when a front develops, while it remains largely unchanged at the remaining DOFs. The Carlson-Miller algorithm accelerates the convergence rate of the modified Newton algorithm by solving low-rank least-squares problems in Krylov subspaces that coincide with the largest change in the system state. An analogous situation arises in simulating the response of structural systems to the sudden failure of one or more members, where redistribution of load can cause significant changes in the tangent stiffness matrix at a small number of DOFs while the rest of the structure remains linear.

The objective of this paper is to develop the Krylov subspace accelerated Newton algorithm for solving nonlinear equilibrium equations in dynamic progressive collapse simulations. The paper begins with a review of the governing equations of structural equilibrium, followed by their solution using the Newton-Raphson and modified Newton algorithms. Next is the development of the Krylov acceleration algorithm and a description of its applicability to nonlinear structural analysis. Through examples, the numerical properties of the Krylov algorithm are compared to those of the conventional Newton algorithms. The paper concludes with recommendations for use of the Krylov acceleration algorithm in nonlinear structural analysis.

Equations of Structural Equilibrium

The equations of nodal equilibrium for the nonlinear dynamic response of a structural system are written in residual form

$$\mathbf{R}(\mathbf{U}(t)) = \mathbf{P}_f(t) - \mathbf{P}_r(\mathbf{U}(t)) - \mathbf{C}\dot{\mathbf{U}}(t) - \mathbf{M}\ddot{\mathbf{U}}(t) \tag{1}$$

where $\mathbf{P}_f$=time-varying vector of applied loads and $\mathbf{P}_r$=vector of resisting forces, which is a nonlinear function of the nodal displacement vector, $\mathbf{U}$. The matrices $\mathbf{M}$ and $\mathbf{C}$ represent the mass and damping, respectively, of the system, while $\dot{\mathbf{U}}$ and $\ddot{\mathbf{U}}$ are the nodal velocities and accelerations, which are related to the nodal displacements by the time integration method. With the nodal displacements as the primary unknowns after time discretization, the statement of equilibrium is that the residual force vector equals zero

$$\mathbf{R}(\mathbf{U}) = \mathbf{0} \tag{2}$$

The root of Eq. (2) represents the state of the structure in which equilibrium is satisfied for the externally applied loads.

Iterative Root-Finding Algorithms

To solve Eq. (2), an iterative root-finding algorithm begins at an initial displacement vector, $\mathbf{U}_0$, then computes successive displacement increments, $\mathbf{V}$, which advance the trial state of the structure

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \mathbf{V}_{k+1} \tag{3}$$

The subscript k counts the number of iterations within a time step. The search for equilibrium terminates successfully when a specified convergence criterion, e.g., that the norm of the residual force vector decreases below either absolute or relative tolerance, is satisfied. After convergence, the trial state of the structure is committed as part of the solution path and the simulation proceeds to the next time step.

Newton-Raphson Algorithm

The Newton-Raphson algorithm is based on a first-order (linear) approximation of the residual vector near the root of Eq. (2)

$$\mathbf{R}_{k+1} = \mathbf{R}_k + \left.\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right|_{\mathbf{U}_k} \mathbf{V}_{k+1} \tag{4}$$

The Jacobian, $\partial \mathbf{R}/\partial \mathbf{U}$, is the tangent stiffness matrix, $\mathbf{K}_T$, of the structure

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}} = -\mathbf{K}_T \tag{5}$$

which contains contributions from the static tangent stiffness matrix, $\partial \mathbf{P}_r/\partial \mathbf{U}$, and the mass and damping matrices. It is assumed that the applied load vector, $\mathbf{P}_f$, does not depend on $\mathbf{U}$.

The Jacobian defined in Eq. (5) is inserted in Eq. (4) giving the following linear correction equation:

$$\mathbf{R}_{k+1} = \mathbf{R}_k - \mathbf{K}_T \mathbf{V}_{k+1} \tag{6}$$

After setting the correction equation equal to zero, the following linear system of equations is obtained for the displacement increment:

$$\mathbf{K}_T \mathbf{V}_{k+1} = \mathbf{R}_k \tag{7}$$

The residual vector and the tangent stiffness matrix are computed at every iteration for the current value of the displacement vector, $\mathbf{U}_k$.

Convergence of the Newton-Raphson algorithm depends on the initial displacement vector, $\mathbf{U}_0$, and the second derivative of the residual force vector. Consequently, difficulties are encountered with the Newton-Raphson algorithm when using large time steps or when there are sharp changes in the tangent stiffness matrix during a time step. The quadratic rate of convergence also depends on the ability to assemble a numerically consistent tangent stiffness matrix (Simo and Taylor 1985), which is often a difficult task for complex constitutive models.

Modified Newton Algorithm

To reduce the computational expense of the Newton-Raphson algorithm, the modified Newton algorithm holds the stiffness matrix constant within a time step. The tangent stiffness matrix in Eq. (7) is replaced with the stiffness matrix from the first iteration in the time step and repeated use is made of the matrix factorization until equilibrium is found. This computational benefit can be outweighed by the increased number of iterations required to reach equilibrium because the local rate of convergence is linear (Stoer and Bulirsch 1993). The modified Newton algorithm generally has a larger radius of convergence than Newton-Raphson.

Shamanskii (1967) proposed a generalization of the Newton-Raphson and modified Newton algorithms where the tangent stiffness matrix is updated every three to six iterations with modified Newton steps taken at the intermediate iterations. The rate of convergence for this periodic Newton algorithm is superlinear (Kelley 1995); however, similar to Newton-Raphson, it has convergence difficulties when there are sharp changes in the residual force vector. The optimal number of intervening modified Newton iterations depends on the computational cost of the tangent stiffness matrix formation and factorization relative to the cost of forming the residual force vector. These costs are derived from the complexity of the element constitutive models and the number of floating point operations required for matrix factorization (Demmel 1997). By performing low-cost matrix-vector operations in Krylov subspaces at the intervening modified Newton steps, the convergence of the periodic Newton algorithm is accelerated and many of the convergence difficulties of the conventional Newton algorithms are overcome.

Krylov Subspace Acceleration Algorithm

Krylov subspaces form the basis for many iterative algorithms in numerical linear algebra, including eigenvalue and linear equation solvers (Golub and Van Loan 1996; Demmel 1997; Trefethen and Bau 1997). A Krylov subspace of dimension m, denoted $\mathcal{K}_m$, is the span of vectors formed by the repeated multiplication of a vector, $\mathbf{b}$, by a matrix, $\mathbf{A}$

$$\mathcal{K}_m = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{m-1}\mathbf{b}\} \tag{8}$$

What makes Krylov subspace methods attractive for numerical algorithms is that the matrix $\mathbf{A}$ is never formed explicitly. Instead, its effect is obtained by taking the difference of vectors.

Carlson and Miller (1998) showed that a Krylov subspace can be formed from selected information about a finite-element model. Then the subspace vectors can be used to accelerate the modified Newton iteration (Miller 2005). At each time step, the Krylov acceleration algorithm seeks the solution to the system of preconditioned residual equations

$$\mathbf{r}(\mathbf{U}) = \mathbf{K}_0^{-1}\mathbf{R}(\mathbf{U}) = \mathbf{0} \tag{9}$$

where $\mathbf{K}_0$ = tangent stiffness matrix at the first iteration of the time step. For a nonsingular $\mathbf{K}_0$, the solution to Eq. (9) is equivalent to that of Eq. (2).

Similar to the Newton-Raphson algorithm, the Krylov acceleration algorithm uses a linear approximation of Eq. (9) near the root; however, the Jacobian matrix in Eq. (6) is replaced by a matrix, $\mathbf{A}$, to give the following linear correction equation:

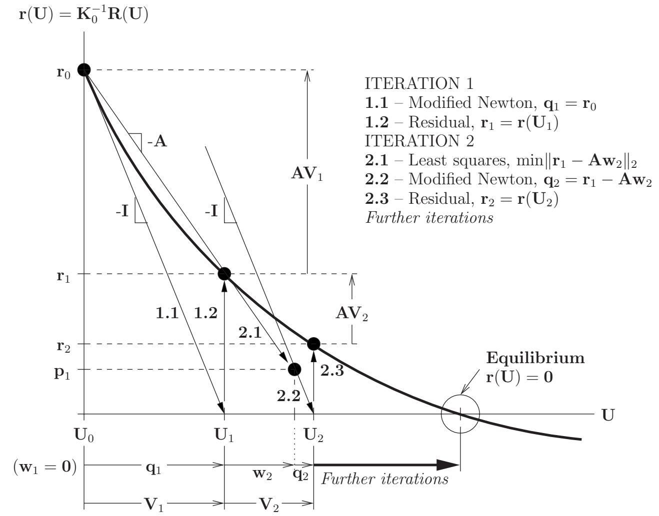

Fig. 1. Graphical representation of the Krylov subspace acceleration method

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{A}\mathbf{V}_{k+1} = \mathbf{0} \tag{10}$$

The matrix $\mathbf{A}$ is assumed to be constant and nonsingular. In the context of Eq. (10), the modified Newton algorithm makes repeated use of the assumption that $\mathbf{A}$ is equal to the identity, i.e., there is no change in the Jacobian during a time step. On the other hand, the Newton-Raphson algorithm uses $\mathbf{A} = \mathbf{K}_0^{-1}\mathbf{K}_T$, where the Jacobian is updated at each iteration.

To solve Eq. (10), the Krylov acceleration algorithm decomposes the displacement increment into two vectors

$$\mathbf{V}_{k+1} = \mathbf{w}_{k+1} + \mathbf{q}_{k+1} \tag{11}$$

where $\mathbf{w}_{k+1}$ = acceleration component and $\mathbf{q}_{k+1}$ = standard modified Newton component of the displacement increment. For the total displacement increment in Eq. (11), Eq. (10) takes the form

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{A}\mathbf{w}_{k+1} - \mathbf{A}\mathbf{q}_{k+1} = \mathbf{0} \tag{12}$$

The first step in satisfying Eq. (12) is to minimize the norm of the vector $\mathbf{r}_k - \mathbf{A}\mathbf{w}_{k+1}$. To this end, $\mathbf{w}_{k+1}$ is specified as a linear combination of the vectors from the subspace of displacement increments, $\mathcal{V}_m = \text{span}\{\mathbf{V}_1, \ldots, \mathbf{V}_m\}$, accumulated during the equilibrium iteration

$$\mathbf{w}_{k+1} = c_1\mathbf{V}_1 + \cdots + c_m\mathbf{V}_m \tag{13}$$

On the first iteration, the subspace is empty and $\mathbf{w}_1$ is zero, as shown in Fig. 1. On subsequent iterations, the vector $\mathbf{A}\mathbf{w}_{k+1}$ is equal to

$$\mathbf{A}\mathbf{w}_{k+1} = c_1\mathbf{A}\mathbf{V}_1 + \cdots + c_m\mathbf{A}\mathbf{V}_m \tag{14}$$

where each term measures the change in the residual at the previous m iterations according to Eq. (10)

$$\mathbf{A}\mathbf{V}_k = \mathbf{r}_{k-1} - \mathbf{r}_k \tag{15}$$

The vector $\mathbf{r}_k - \mathbf{A}\mathbf{w}_{k+1}$ in Eq. (12) thus represents an overdetermined system of equations for the unknown coefficients $c_1, \ldots, c_m$ in Eq. (14). The norm of this vector is minimized by least-squares analysis (Golub and Van Loan 1996) and the minimizing coefficients give the first component of the displacement increment according to the linear combination in Eq. (13).

At the intermediate solution point, $\mathbf{U}_k + \mathbf{w}_{k+1}$, the residual $\mathbf{r}_k$ has been reduced by $\mathbf{A}\mathbf{w}_{k+1}$, giving the *least-squares* residual,

$\mathbf{p}_k=\mathbf{r}_k-\mathbf{A}\mathbf{w}_{k+1}$, shown in Fig. 1. The correction Eq. (12) now takes the form

$$\mathbf{r}_{k+1} = \mathbf{p}_k - \mathbf{A}\mathbf{q}_{k+1} = \mathbf{0} \tag{16}$$

The vector $\mathbf{q}_{k+1}$ remains unspecified at this point, thus a further computation is required to solve Eq. (16). Similar to the modified Newton algorithm, the matrix $\mathbf{A}$ is assumed to be the identity matrix and the value of $\mathbf{q}_{k+1}$ that solves Eq. (16) is the least-squares residual, $\mathbf{q}_{k+1}=\mathbf{p}_k$. At this point, both components of the displacement increment, $\mathbf{V}_{k+1}$, are defined in order to update the trial state of the structure. The computations taken by the Krylov acceleration algorithm during one simulation time step are summarized in Fig. 1.

Carlson and Miller (1998) showed by mathematical induction that the subspace $\mathcal{V}_m=\text{span}\{\mathbf{V}_1,\ldots,\mathbf{V}_m\}$ generated during the search for equilibrium coincides with the Krylov subspace $\mathcal{K}_m=\text{span}\{\mathbf{r}_0,\mathbf{A}\mathbf{r}_0,\ldots,\mathbf{A}^{m-1}\mathbf{r}_0\}$. At the first iteration, the subspace $\mathcal{V}_m$ is empty and the algorithm produces the modified Newton displacement increment, $\mathbf{V}_1=\mathbf{r}_0$, as shown in Fig. 1. On subsequent iterations, the least-squares computation advances the solution and the factorization of the tangent stiffness matrix from the first iteration is reused, giving a computational savings over Newton-Raphson and faster rate of convergence than modified Newton. This algorithm, and the problem it solves, is completely different from standard "nonlinear Krylov" and "inexact Newton" methods (Brown and Saad 1990; Kelley 1995) in that it forms Krylov subspaces as part of the root-finding algorithm rather than as part of an iterative solution to the linear system equations formed at each Newton iteration.

Implementation and Interpretation of the Algorithm

Pseudocode for the Krylov acceleration algorithm is shown in Fig. 2 using MATLAB (2007) matrix-vector notation. For general finite-element analysis, the algorithm has been implemented in the OpenSees software framework (McKenna et al. 2000) using the *Strategy* software design pattern (Gamma et al. 1995; McKenna et al. 2010). The implementation requires the storage of up to $m_{\max}$ vectors, $\mathbf{V}_i$, in order to compute $\mathbf{w}_{k+1}$ by Eq. (13). In addition, the least-squares solution requires the same amount of storage for the vectors, $\mathbf{AV}_i$, bringing the total storage requirement for the Krylov algorithm to $2m_{\max}N_{\text{DOF}}$, where N_{DOF} is the number of DOFs. To keep storage to a minimum, the parameter $m_{\max}$ is typically between three and six. A small value of $m_{\max}$ also keeps to a minimum the computational cost of the least-squares solution. When equilibrium is not found after $m_{\max}$ iterations within a time step, the algorithm zeros the subspace vectors and recomputes the tangent stiffness matrix for use on the next $m_{\max}$ iterations, as indicated in the code of Fig. 2.

The two components of the displacement increment, $\mathbf{V}_{k+1}$, each have a distinct physical interpretation. The first component, $\mathbf{w}_{k+1}$, advances the solution with a bias, determined by least-squares analysis, toward the DOFs where the largest changes in state occur, as measured by the change in nodal displacements at previous iterations. This is what the Newton-Raphson algorithm does at every DOF by continually updating the tangent stiffness matrix. The second component of the displacement increment, $\mathbf{q}_{k+1}$, uses a modified Newton computation to advance the iteration further toward equilibrium at DOFs where there are smaller changes in the residual.

When yielding, geometric instability, and other significant changes in state occur in a structural system, the tangent stiffness matrix changes drastically at only a few DOFs and remains relatively unchanged elsewhere. In this case, the matrix $\mathbf{A}$ is significantly different from the identity on a subspace of small dimension. The Krylov acceleration method is able to solve efficiently for such low-rank changes in the structural stiffness, as demonstrated in the following examples.

```
U = Uo;
R = residual(U);
m = mmax+1;  % Subspace dimension

% Main loop
while (norm(R) > tol)

   % Refresh tangent and clear subspace
   if (m > mmax)
      K = jacobian(U);
      [l,u] = lu(K);
      m = 0;
   end

   % Backsolve
   r = u \ (l \ R);
   AV(:,m+1) = r;

   % Least squares analysis
   if (m > 0)
      AV(:,m) = AV(:,m) - r;
      c = AV(:,1:m) \ r;
      r = r + V(:,1:m)*c;  % w
      r = r - AV(:,1:m)*c; % q
   end

   % Update state of structure
   U = U + r;
   R = residual(U);
   m = m+1;
end
```

Fig. 2. MATLAB code for Krylov subspace acceleration algorithm

Numerical Examples

The first two examples demonstrate the behavior of the Krylov algorithm for a two-DOF nonlinear structural system using a consistent tangent matrix, and an inconsistent tangent. In the second set of examples, comparisons of the convergence behavior of the Krylov and Newton-Raphson algorithms are made by simulating the progressive collapse of a small reinforced concrete frame and a moderately sized steel frame.

Two-DOF Spring System

A two-DOF example illustrates the differences between the Krylov acceleration algorithm and the Newton-Raphson and modified Newton algorithms. The system consists of three uniaxial springs with bilinear force-deformation relationships shown in Fig. 3. Using the compatibility matrix shown in the figure, the consistent tangent stiffness matrix is obtained from the basic spring stiffnesses

$$\mathbf{k}_b = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \rightarrow \mathbf{K}_T = \mathbf{A}^T\mathbf{k}_b\mathbf{A} = \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2+k_3 \end{bmatrix} \tag{17}$$

A load vector of $\mathbf{P}_f=[6\ \ 12]^T$ is applied to the system in one time step and only static response is considered. The convergence cri-

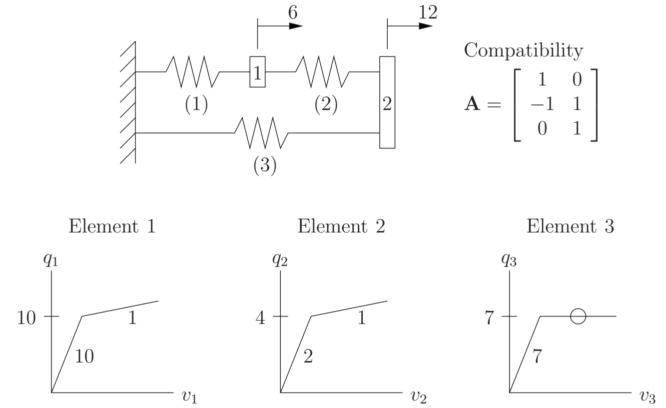

Fig. 3. Two-DOF spring system

terion is that the norm of the residual vector reduces from its initial value by four orders of magnitude.

The Newton-Raphson algorithm finds the equilibrium solution, at nodal displacement vector $\mathbf{U}=[2\ \ 5]^T$, in three iterations (three residual evaluations and three matrix factorizations), as shown in Fig. 4 (from initial point to A, A to B, then B to the solution). The modified Newton algorithm requires 140 iterations (140 residual evaluations and one matrix factorization) to find the solution point, where it is observed in Fig. 4 that the rate of convergence is slow as the solution point is approached. The Krylov accelerated Newton algorithm (with $m_{\max}=1$) requires five iterations (five residual evaluations and three matrix factorizations). As is typically the case for small structural models under monotonic loading with simple strain-hardening constitutive models, Newton-Raphson is the most efficient algorithm. There are however more complicated instances where the Newton-Raphson algorithm is less robust, as demonstrated in the following examples.

Two-DOF Spring System with Inconsistent Tangent

The efficiency of the Newton-Raphson algorithm relies on the use of a numerically consistent tangent stiffness matrix. For complex constitutive models, a consistent tangent can be difficult to develop and implement, and the result of using an inconsistent tangent or one with approximation errors is often nonconvergence of the Newton-Raphson algorithm, as demonstrated in this example.

To mimic an error in the tangent calculation of the element state determination, an artificial coupling of Springs 1 and 2 is introduced, leading to a tangent stiffness matrix that is inconsistent with the residual vector. The basic stiffness matrix of the system contains off-diagonal terms, leading to the following tangent stiffness matrix:

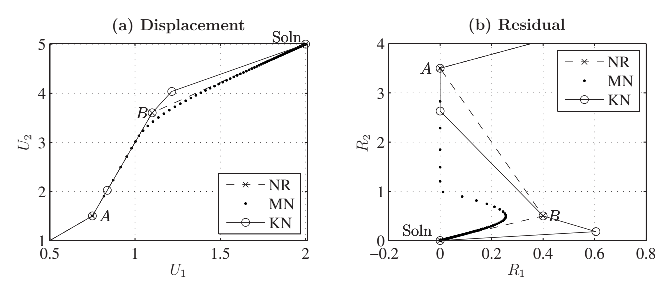

Fig. 4. Trace of equilibrium search for two-DOF spring example: (a) displacement vector; (b) residual vector

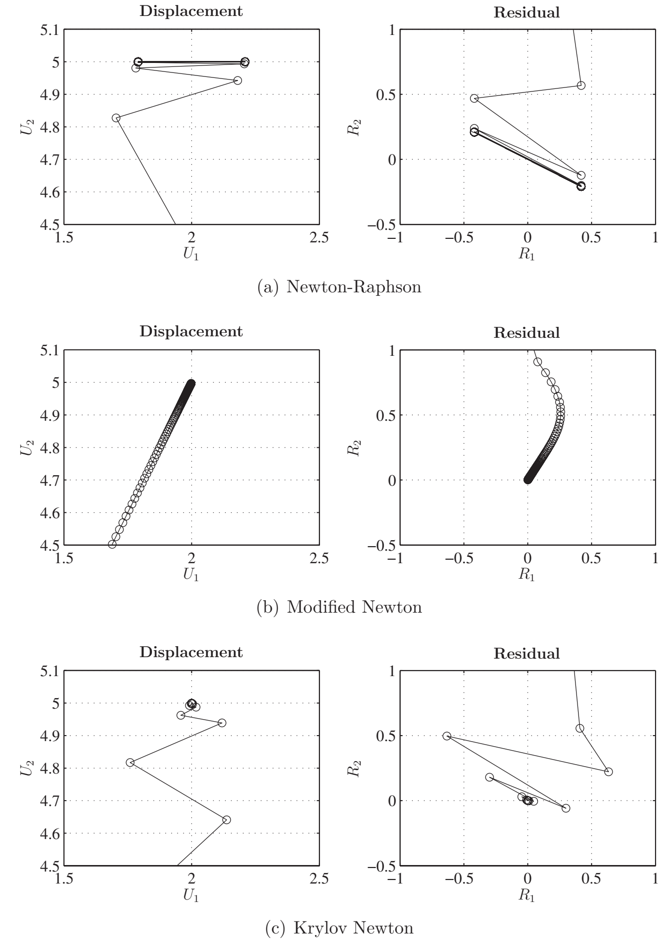

Fig. 5. Trace of equilibrium search for two-DOF spring example with inconsistent tangent: (a) Newton-Raphson; (b) modified Newton; and (c) Krylov-Newton

$$\mathbf{k}_b=\begin{bmatrix} k_1 & 0.5 & 0 \\ 0.5 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \rightarrow \mathbf{K}_T=\mathbf{A}^T\mathbf{k}_b\mathbf{A}=\begin{bmatrix} k_1+k_2-1 & -k_2+0.5 \\ -k_2+0.5 & k_2+k_3 \end{bmatrix} \tag{18}$$

As shown in Fig. 5(a), the Newton-Raphson algorithm approaches the solution point then begins to oscillate or flip-flop indefinitely between $[1.8\ \ 5]^T$ and $[2.2\ \ 5]^T$ with corresponding changes in the residual vector. The modified Newton algorithm converges to the solution after 143 iterations of applying the inconsistent tangent to the residual, with the slow convergence shown in Fig. 5(b). On the other hand, the Krylov accelerated Newton algorithm is able to find the solution after 14 iterations. As shown in Fig. 5(c), the Krylov algorithm begins to show the potential for residual flip-flop as the iterations approach the solution point, but it is able to converge to the solution by using least-squares computations rather than relying solely on the inconsistent tangent.

While the matrix in Eq. (18) was contrived to demonstrate the effect of an inconsistent tangent, the resulting behavior of the Newton-Raphson algorithm is representative of that which occurs when there are sharp changes in the residual, e.g., stiff unloading at a load reversal in a dynamic analysis. The Krylov acceleration

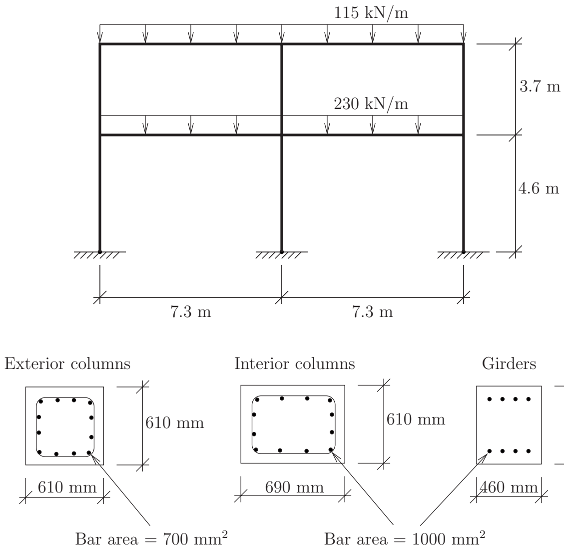

Fig. 6. Reinforced concrete frame example

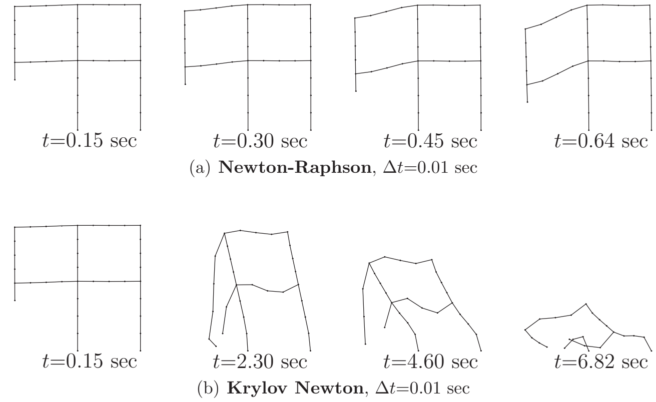

Fig. 7. Configuration of reinforced concrete frame after initiating progressive collapse by removal of exterior first story column (no display magnification)

algorithm can resolve efficiently such flip-flop behavior when convergence cannot be achieved with the Newton-Raphson algorithm.

Reinforced Concrete Frame

Predicting the progressive collapse potential of reinforced concrete frames requires solution algorithms that can solve for the structural response with large time steps, especially when simulating fine scale phenomena at the constitutive level such as crushing, strain softening, and axial-moment interaction via computationally intense fiber models. Progressive collapse simulation of the two-bay two-story reinforced concrete frame shown in Fig. 6 demonstrates the convergence behavior of the Krylov subspace acceleration algorithm.

Each frame member is discretized with four beam-column finite elements using the corotational transformation for geometric nonlinearity (Crisfield 1991) and a displacement-based formulation (linear curvature approximation) of material nonlinearity using fiber discretizations of the cross sections shown in Fig. 6. The model has 108 equations of nodal equilibrium (DOFs). Reinforcing steel is assumed bilinear with f_y=420 MPa, E =200,000 MPa, and 2% kinematic hardening. The nominal concrete strength is 28.0 MPa, with confining effects of column transverse reinforcement (Mander et al. 1988) increasing the core strength to 36.0 MPa. Fiber models simulate the cross section response with 20 concrete fibers through the section depth. To simulate loss of load carrying capacity, fibers are assumed to fail (carry zero stress and provide zero tangent) when ultimate strain levels are reached. More realistic macromodels of reinforced concrete member behavior during progressive collapse have been proposed by Bao et al. (2008). The Newmark average acceleration method is used for time integration and the convergence tolerance is 10^{-8} on the norm of the displacement increment vector, $\mathbf{V}_{k+1}$.

To ensure complete collapse of the frame upon removal of an exterior column (Member 1 in Fig. 6), heavy gravity loads are applied to the girders. When a node reaches the ground, it, along with its connected elements and associated member loads, is removed from the structural model and the analysis proceeds with the resulting smaller structural system. This allows the collapse simulation to proceed as long as possible and test the solution algorithms; however, it ignores phenomena such as impact forces (Kaewkulchai and Williamson 2006).

The results of the progressive collapse simulation are summarized in Fig. 7, which shows the final displaced shape of the frame, as well as selected intermediate configurations, at the point where the equilibrium solution fails to converge. Using a time step of Δt=0.01 s with the Newton-Raphson algorithm, the final converged state of the frame is at t=0.64 s, as shown in Fig. 7(a). At time t=0.65 s, the equilibrium iteration flip-flops between displacement increment norms are on the order of 10^9. To reach the final converged state, the Newton-Raphson algorithm requires 313 evaluations for both the residual vector and tangent stiffness matrix. Although there is a significant downward displacement (about 1.5 m) of the second story, which is indicative of severe structural damage, it is not clear from the final displaced shape if the second bay will be able to continue to resist loads. In an attempt to keep the simulated structural response within the radius of convergence of the Newton-Raphson algorithm, a time step one order of magnitude smaller (Δt=0.001 s) is used. In this case, the simulation proceeds slightly further to t=0.754 s with no discernible difference in the final state of the frame. It is noted that the modified Newton algorithm was not able to find an equilibrium solution beyond t=0.1 s for either Δt=0.01 or 0.001 s.

Returning to the larger time step (Δt=0.01 s) but employing the Krylov acceleration algorithm with $m_{\max}$=3, the simulation converges at each time step until t=6.82 s, where the frame is completely collapsed, as shown in Fig. 7(b). It is noted that to reach t=0.64 s, the final converged state of the Newton-Raphson algorithm using Δt=0.01 s, the Krylov algorithm requires fewer matrix evaluations (145) but more residual evaluations (492) than the 313 required for each with Newton-Raphson.

Steel Frame

A moderately sized moment-resisting steel frame is the final example. The frame dimensions and member sizes are shown in Fig. 8 and are adapted from collapse simulations performed by Khandelwal and El-Tawil (2007). The finite-element model used herein consists of three displacement-based beam elements per member (672 equations of nodal equilibrium), corotational geometric transformations, and fiber discretizations of all cross sections (eight fibers through the depth of each web and two fibers through each flange). The steel material behavior is assumed elastic-

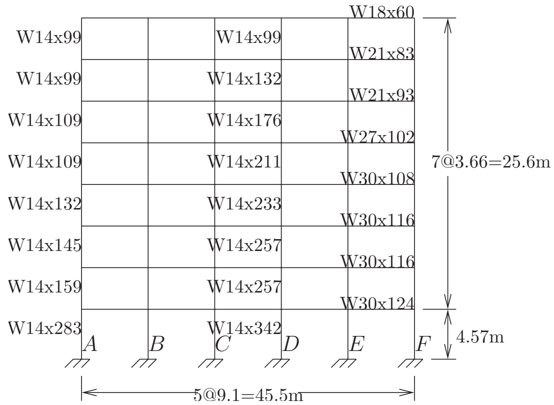

Fig. 8. Steel frame example

perfectly plastic with elastic modulus $E=200,000$ MPa and yields stress $\sigma_y=240$ MPa. As in the previous example, members that reach the ground are removed from the model and steel fibers are assumed to fail when ultimate strain is reached. Accurate constitutive models of steel frames under progressive collapse have been developed by Khandelwal et al. (2008). The combined dead and live load is 58.4 kN/m for all girder members and the frame is analyzed for static equilibrium of the gravity loads before initiating dynamic progressive collapse with the Newmark average acceleration method and convergence tolerance of 10^{-8} on the norm of the displacement increment vector.

After obtaining equilibrium for gravity loads, the frame is able to reach a stable dynamic solution due to the removal of only column *A* or column *B*. Removing both columns *A* and *B* simultaneously produces complete collapse. The Newton-Raphson algorithm with a time step of $\Delta t=0.01$ s is able to track progressive collapse mechanisms to $t=2.17$ s, at which point the equilibrium solution fails with the deformed shape shown in Fig. 9(a). At this point, the Newton-Raphson algorithm fails to converge where the norm of the residual force vector stagnates at 1264.96 while the norm of the displacement increment vector stays in the range of $10^{11}-10^{13}$. The algorithm requires 973 residual evaluations and matrix factorizations to reach the final converged state. Using a smaller time step, $\Delta t=0.001$ s, does not improve the Newton-Raphson simulation as nonconvergence occurs at $t=2.184$ s in this case. Although the algorithm is able to track the equilibrium path up to this point, it is not clear if the remaining portion of the structure will resist collapse were the simulation able to continue

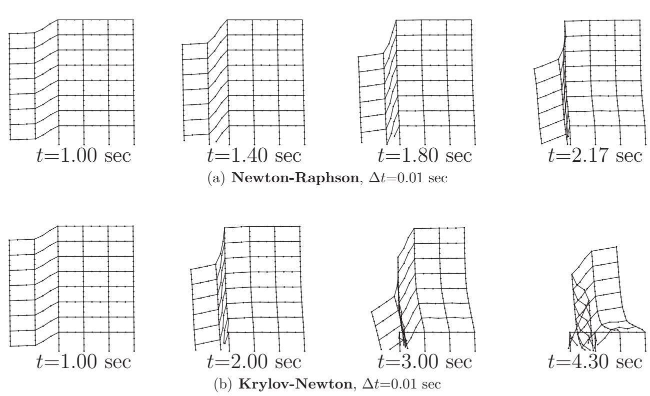

Fig. 9. Displaced shape of steel frame after removal of two first floor columns (no display magnification)

past the last converged state. For both the large and small time steps, the Newton-Raphson algorithm fails to find a solution when large strains and sudden changes in resisting force dominate the girder response in bay *BC*. For analysis time steps, $\Delta t=0.01$ and 0.001 s, the modified Newton algorithm fails at $t=0.21$ and 0.277 s, respectively.

As shown in Fig. 9(b), simulation with the Krylov acceleration algorithm and the larger time step, $\Delta t=0.01$ s, continues well past the point of nonconvergence for the Newton-Raphson algorithm. A larger maximum subspace dimension, $m_{\max}=6$, is used since the structural model has a larger number of DOFs than the previous example. Up to simulation time $t=2.17$ s, the last converged state found via Newton-Raphson with $\Delta t=0.01$ s, the Krylov algorithm requires fewer matrix factorizations (437) and more residual evaluations (2169) than Newton-Raphson (973). However, the Krylov algorithm is able to track subsequent failures, including buckling of a second story column and the resulting second story collapse, up to simulation time $t=3.79$ s, at which point the removed elements and nodes result in a singular stiffness matrix. Although the sequence of events shown in Fig. 9(b) is unlikely due to fracture and connection failures in bay *BC* and the omission of impact forces in the analysis (Kaewkulchai and Williamson 2006), it demonstrates the ability of the Krylov acceleration algorithm to track severe nonlinear phenomena in an implicit dynamic progressive collapse simulation.

Conclusions

This paper has presented an accelerated Newton algorithm based on computations in Krylov subspaces for solving the equations of dynamic equilibrium in nonlinear structural analysis. The algorithm uses a low-rank least-squares analysis and a modified Newton computation at each iteration during a simulation time step. The least-squares analysis searches for equilibrium at the DOFs where the most significant nonlinear response occurs, while the modified Newton computation advances the equilibrium search at DOFs where the structural response is relatively linear. The example applications show that the Krylov algorithm is able to converge with an error in the tangent stiffness matrix and it is also able to track progressive collapse through more failure mechanisms when compared to the Newton-Raphson algorithm. Further research will focus on the scalability of the Krylov acceleration algorithm to large structural systems and on an implementation of the algorithm in a parallel computing environment.

Although many advantages have been demonstrated for the Krylov acceleration algorithm, it is by no means a "silver bullet," as there are situations where Newton-Raphson, modified Newton, or another solution algorithm will outperform the Krylov algorithm in one or more metrics. An important metric that will be the focus of future research is the computational cost of matrix factorizations relative to residual evaluations in progressive collapse simulations as the Krylov algorithm generally requires fewer matrix factorizations than Newton-Raphson but more residual evaluations. For structural models with expensive residual evaluations, e.g., when using nonlinear beam elements with fiber cross sections, it is recommended to begin a simulation with the Newton-Raphson algorithm, then switch to the Krylov acceleration algorithm when there are convergence difficulties and return to Newton-Raphson after a sufficient number of Krylov time steps. Switching between solution algorithms is handled easily in scriptable finite-element analysis packages such as OpenSees (Mazzoni et al. 2006).

Regardless of computational cost, an important feature of the Krylov algorithm is its ability to solve equilibrium equations when using complex constitutive models for which a numerically consistent tangent may be difficult to implement. The least-squares component of the Krylov acceleration algorithm can correct for errors in the tangent stiffness that would otherwise lead to slow convergence or nonconvergence of the Newton-Raphson and modified Newton algorithms.

Acknowledgments

This work and its development in OpenSees have been supported by the Pacific Earthquake Engineering Research Center under Grant No. EEC-9701568 from the National Science Foundation to the University of California, Berkeley. The writers would like to thank Professor Keith Miller in the Department of Mathematics at the University of California, Berkeley for discussions and guidance regarding the Krylov subspace acceleration algorithm.

References

Bao, Y., Kunnath, S. K., El-Tawil, S., and Lew, H. S. (2008). "Macromodel-based simulation of progressive collapse: RC frame structures." *J. Struct. Eng.*, 134(7), 1079–1091.

Bathe, K. J. (1996). *Finite element procedures*, Prentice-Hall, Upper Saddle River, N.J.

Brown, P. N., and Saad, Y. (1990). "Hybrid Krylov methods for nonlinear systems of equations." *SIAM J. Sci. Comput. (USA)*, 11, 450–481.

Broyden, C. G. (1967). "Quasi-Newton methods and their application to function minimisation." *Math. Comput.*, 21, 368–381.

Broyden, C. G. (1970). "The convergence of a double-rank minimisation. 2: The new algorithm." *J. Inst. Math. Appl.*, 6, 222–231.

Carlson, N., and Miller, K. (1998). "Design and application of a 1D GWMFE code." *SIAM J. Sci. Comput. (USA)*, 19, 766–798.

Crisfield, M. A. (1984). "Accelerating and damping the modified Newton-Raphson method." *Comput. Struct.*, 18, 395–407.

Crisfield, M. A. (1991). *Non-linear finite element analysis of solids and structures*, Vol. 1, Wiley, New York.

Demmel, J. W. (1997). *Applied numerical linear algebra*, SIAM, Philadelphia.

Fletcher, R. (1970). "A new approach to variable metric algorithms." *Comput. J.*, 13, 317–322.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*, Addison-Wesley, Reading, Mass.

Goldfarb, D. (1970). "A family of variable-metric methods derived by variational means." *Math. Comput.*, 24(109), 23–26.

Golub, G. H., and Van Loan, C. F. (1996). *Matrix computations*, 3rd Ed., Johns Hopkins University Press, Baltimore, Md.

Kaewkulchai, G., and Williamson, E. B. (2004). "Beam element formulation and solution procedure for dynamic progressive collapse analysis." *Comput. Struct.*, 82(7–8), 639–651.

Kaewkulchai, G., and Williamson, E. B. (2006). "Modeling the impact of failed members for progressive collapse analysis of frame structures." *J. Perform. Constr. Facil.*, 20(4), 375–383.

Kelley, C. T. (1995). *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia.

Khandelwal, K., and El-Tawil, S. (2007). "Collapse behavior of steel special moment resisting frame connections." *J. Struct. Eng.*, 133(5), 646–655.

Khandelwal, K., El-Tawil, S., Kunnath, S. K., and Lew, H. S. (2008). "Macromodel-based simulation of progressive collapse: Steel frame structures." *J. Struct. Eng.*, 134(7), 1070–1078.

Mander, J. B., Priestley, M. J. N., and Park, R. (1988). "Theoretical stress-strain model for confined concrete." *J. Struct. Eng.*, 114(8), 1804–1826.

MATLAB. (2007). Version 7.4, Natick, Mass.

Matthies, H., and Strang, G. (1979). "The solution of non-linear finite element equations." *Int. J. Numer. Methods Eng.*, 14, 1613–1626.

Mazzoni, S., McKenna, F., Scott, M. H., and Fenves, G. L. (2006). *Open system for earthquake engineering simulation user command-language manual (version 1.7.3)*. Univ. of Calif., Berkeley, Calif., ⟨http://opensees.berkeley.edu/OpenSees/manuals/usermanual/⟩ (Sept. 1, 2008).

McKenna, F., Fenves, G. L., and Scott, M. H. (2000). *Open system for earthquake engineering simulation*, Univ. of Calif., Berkeley, Calif., ⟨http://opensees.berkeley.edu⟩ (Sept. 1, 2008).

McKenna, F., Scott, M. H., and Fenves, G. L. (2010). "Nonlinear finite element analysis software architecture using object composition." *J. Comput. Civ. Eng.*, 24(1), 95–107.

Menchel, K., Massart, T. J., Rammer, Y., and Bouilliard, P. (2009). "Comparison and study of different progressive collapse simulation techniques for *RC* structures." *J. Struct. Eng.*, 135(6), 685–697.

Miller, K. (2005). "Nonlinear Krylov and moving nodes in the method of lines." *J. Comput. Appl. Math.*, 183(2), 275–287.

Sasani, M., and Sagiroglu, S. (2008). "Progressive collapse resistance of Hotel San Diego." *J. Struct. Eng.*, 134(3), 478–488.

Shamanskii, V. E. (1967). "A modification of Newton's method." *Ukr. Mat. Zh.*, 19, 133–138.

Shanno, D. F. (1970). "Conditioning of quasi-Newton methods for function minimisation." *Math. Comput.*, 24, 322–334.

Simo, J. C., and Taylor, R. L. (1985). "Consistent tangent operators for rate independent elasto-plasticity." *Comput. Methods Appl. Mech. Eng.*, 48, 101–118.

Stoer, J., and Bulirsch, R. (1993). *Introduction to numerical analysis*, 2nd Ed., Springer, New York.

Trefethen, L. N., and Bau, D., III (1997). *Numerical linear algebra*, SIAM, Philadelphia.

Zienkiewicz, O. C., and Taylor, R. L. (2005). *The Finite element method for solid and structural mechanics*, 6th Ed., Butterworth-Heinman, Stoneham, Mass.