INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 38, 1597–1610 (1995)

AN EFFICIENT MULTILEVEL SOLUTION SCHEME FOR LARGE SCALE NON-LINEAR SYSTEMS

J. FISH, M. PANDHEERADI AND V. BELSKY

Department of Civil Engineering and Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.

SUMMARY

We present a new approach for the fast, efficient solution of large systems of non-linear equations arising from the finite element discretization. The proposed non-linear solver builds on the advantages of the popular methods of solution that are currently being employed, while eliminating most of their undesirable features. It combines the well-known BFGS method with the FAS version of the multigrid method, introduced by Brandt,[1] to form a fast, efficient solution method for non-linear problems. We present numerical performance studies that are indicative of the convergence properties as well as the stability of the new method.

KEY WORDS: multi level; non-linear solution; FAS; multigrid

1. INTRODUCTION

There are popular alternatives to Newton's method for the solution of large systems of non-linear equations. They include the BFGS quasi-Newton method, the secant-Newton method, and multigrid methods. A careful investigation of the advantages and disadvantages of classical approaches to the solution of non-linear system of equations serves as a motivation for the introduction of the new solution approach presented in this paper. Our observations are predominantly based on our own experience apart from, of course, a thorough investigation of most of the pioneering work published in the literature that is related to the subject of our paper.

Since the pioneering work of Matthies and Strang,[2] that outlined a practical implementation of BFGS algorithm for the solution of non-linear finite element equations, there have appeared a series of related works (see Reference 3 for a review). The apparent robustness and stability of BFGS method make it an excellent choice for fast solution of general non-linear systems. The efficient implementation of the algorithm discussed by these authors, which preserves the skyline storage of the stiffness matrix by storing a series of update vectors and the initial factorized stiffness instead of the updated jacobian itself, renders it a particularly attractive choice in situations where storage requirements are also a concern. Our experience with problems involving material non-linearity shows that reliability of the BFGS method is greatly affected by the starting approximation to the jacobian matrix. More specifically, using the identity matrix with scaling or the diagonal of the jacobian greatly inhibits convergence, more often resulting in no convergence at all. Of course, there are other alternatives for the starting approximation in between the extremes of the full matrix and diagonal. One of these choices, the complete SSOR matrix (see Reference 4), has been investigated here for comparison purposes.

CCC 0029–5981/95/101597–14
© 1995 by John Wiley & Sons, Ltd.

Received 14 February 1994
Revised 26 August 1994

Another class of methods where the approximation to the Jacobian is updated on the basis of the quasi-Newton equation are the secant-Newton methods. But in this case the updates are not cumulative, and are always restarted from some approximation to the current jacobian which acts as a preconditioner. In this respect the secant-Newton algorithms have features similar to those of non-linear preconditioned conjugate gradient methods.[5–9] Our experience indicates that if the preconditioner has close resemblance to the tangent stiffness matrix, the secant-Newton methods perform well.

For large problems application of quasi-Newton methods or secant-Newton methods may become expensive because of the computational labour involved in the factorization of the initial Jacobian matrix for the classical BFGS method or in the application of a suitable preconditioner that is close to the Jacobian matrix for secant-Newton methods. In such a situation one of the most popular alternatives is multigrid methods. Here we discuss two different versions of multigrid methods for non-linear problems.[1]

A class of multigrid methods for non-linear problems where a linear multigrid method is used within the Newton method has been analysed by Kacou and Parsons,[10] where the method is applied for the solution of elastoplastic problems in a parallel computing environment. Our experience indicates that unless the strain-hardening slope is small, the Newton-multigrid method performs well and the Newton iterations retain their quadratic convergence rate even if the linear system is not solved exactly at every Newton step. Three major disadvantages to this approach are worth noting:

(a) The method runs into serious difficulties due to poor conditioning of the linear system for problems with significant amount of plastic flow and small hardening slope that make the solution of the linear system through the multigrid method even to a moderate tolerance prohibitively slow.
(b) The sensitivity of the standard Newton's method to starting approximation is inherited by multigrid-Newton method resulting in divergence unless a line search is applied in the case of reasonably large load steps.
(c) The method requires storage of the full jacobian matrix as in the standard Newton's method.

Another form of multigrid method, termed as Full Approximation Storage (FAS), was introduced by Brandt.[1] Its name is derived from the fact that instead of storing a correction on the auxiliary grid to the fine grid approximation, the full current approximation, namely the sum of the correction and its base approximation, is stored. In this algorithm an explicit global linearization is not required and the basic algorithm is made up of suitable non-linear relaxations or smoothing iterations performed on the finest and all intermediate levels and the near-exact solution of a non-linear problem on the coarsest grid at every multigrid iteration.

There are many conventional choices for the relaxation procedure to be employed, including non-linear Jacobi, non-linear Gauss–Seidel and conjugate gradient methods. In practice, these smoothing techniques make the FAS algorithm inefficient because of the enormous computation time that is spent on finer grids in computation of at least the diagonal terms of the Jacobian, especially for large scale history-dependent problems. Nevertheless, FAS has excellent convergence properties in addition to the fact that it provides considerable savings in storage. The only storage needed on finer levels is a few vectors as opposed to the storage for the full Jacobian matrix for multigrid-Newton method and the initial approximation to the Jacobian and the series of accumulated update vectors for the BFGS method. The total storage for all levels can be shown to be of order n where n is the problem size.[1] We will expand on our comments on the four methods discussed in this section later in the paper, with the help of numerical examples.

Keeping in mind the relative advantages and disadvantages of each method discussed in the previous section, we aim to develop a non-linear solver that builds on the advantages of multigrid- and BFGS-like methods and at the same time eliminates the undesirable characteristics of each to the maximum extent possible. The preliminary results on the numerical performance of our algorithm are encouraging and we believe the method is extremely useful for solving large scale non-linear systems.

The organization of the rest of the paper is as follows: In Section 2 we define the problem and then summarize the basic principles of solving it for each of the methods discussed in the introduction. The basic idea of the FAS-BFGS method and the detailed description of this new solution approach are described in Section 3. Numerical results and our conclusions appear in Sections 4 and 5, respectively.

2. PROBLEM STATEMENT AND CLASSICAL NON-LINEAR SOLVERS

Consider a discretized system of non-linear equations

$$r \equiv f_{\text{ext}} - f_{\text{int}}(d, \eta) = 0 \tag{1}$$

where r is the gradient or residual vector, f_{ext} is the external force vector, f_{int} is the vector of internal forces, d the solution vector, and η refers to the set of all variables that influence f_{int} apart from the solution vector d. It follows that η may include history variables for problems that are path or history-dependent. For example, for problems involving material non-linearity like plasticity, η represents the state of stress σ, accumulated plastic strain $\bar{\varepsilon}^{\text{pl}}$, etc. Computation of the current values of the history variables represented by η usually involves the numerical integration of the appropriate constitutive relations to obtain the stress state from the strain or deformation history up to the current time step. For simplicity of notation, we omit η from the argument list from now on.

The non-linear system (1) will invariably have to be solved by some iterative scheme that may or may not involve global linearization. In any case, the general iterative scheme for solving the non-linear system (1) can be represented in the following way:

$$d_{i+1} = d_i + \alpha_i p_i \tag{2}$$

where p is a 'search direction', α a scalar parameter usually obtained through a 'line search'. For non-linear problems an exact line search is seldom done since it may involve numerous, expensive gradient vector evaluations. One of the methods commonly adopted is to iteratively solve the line search equation $p_i^{\mathrm{T}} r(d_i + \alpha p_i) = 0$ until the criterion

$$|p_i^{\mathrm{T}} r(d_i + \alpha p_i)| \leqslant \mu |p_i^{\mathrm{T}} r_i| \tag{3}$$

is satisfied. Here μ is the line search tolerance parameter which is normally dependent on the class of non-linear problem and the particular solution method adopted. The typical value of μ normally ranges from 0·25 to 0·5.[11] The characterization of a particular method is dependent mainly on how the search directions p_i are formulated.

We start by presenting a brief summary of conventional solution methods. The purpose of this is two-fold: (i) to identify the strong points of various classical methods and (ii) to construct a hybrid method that combines select features of these methods to perform certain sub-tasks within the proposed method.

2.1. BFGS quasi-Newton method

In quasi-Newton method p_i in (2) is computed as $p_i = H_i r_i$ with the matrices H_i satisfying the secant or quasi-newton equation $H_{i+1}^{-1} u_i = v_i$ where $u_i = d_{i+1} - d_i$ and $v_i = r_{i+1} - r_i$. The quasi-Newton update proven to be most successful is the BFGS update—a rank 2 update—which preserves the symmetry and positive definiteness of the Jacobian approximation. It may be expressed in the following form that gives the inverse H_{i+1} of the Jacobian approximation directly in terms of H_i (Reference 2):

$$H_{i+1} = (I - \rho_i u_i v_i^{\mathrm{T}}) H_i (I - \rho_i v_i u_i^{\mathrm{T}}) + \rho_i u_i u_i^{\mathrm{T}} \tag{4}$$

where

$$\rho_i = \frac{1}{u_i^{\mathrm{T}} v_i} \tag{5}$$

We refer to Reference 12 for a detailed discussion of quasi-Newton methods.

2.2. Secant-Newton method

The expression of p_i in secant-Newton method is similar to that in BFGS method except that H_{i+1} in (4) is now computed from a preconditioner $\bar{H}$ rather than from H_i:

$$H_{i+1} = (I - \rho_i u_i v_i^{\mathrm{T}}) \bar{H} (I - \rho_i v_i u_i^{\mathrm{T}}) + \rho_i u_i u_i^{\mathrm{T}} \tag{6}$$

As discussed before, the choice of $\bar{H}$ influences the properties of secant-Newton method significantly.

2.3. Newton-Multigrid method

Computation of p_i in this case, involves the solution of the linearized system of equations

$$K_i p_i = r_i \tag{7}$$

where $K_i = [\nabla(f_{\mathrm{int}})]_{d_i, \eta_i}$ is the Jacobian matrix.

A standard linear multigrid method is applied to obtain the solution of the linear system (7) to some tolerance. Assuming two grids G^0 and G^1 with mesh sizes $h^0 > h^1$ one can summarize one linear multigrid cycle for the solution of

$$r^1 - K^1 p^1 = 0 \tag{8}$$

on grid G^1 as follows.

(i) *Pre-smoothing.* Apply m_1 relaxation steps on (8) to obtain $p^1_{m_1}$ and $\bar{r}^1_{m_1} \equiv r^1 - K^1 p^1_{m_1}$.

(ii) *Coarse grid correction.* Solve the linear system on grid G^0 given by $r^0 - K^0 (\Delta p)^0 = 0$ where $r^0 \equiv I_1^0 \bar{r}^1_{m_1}$, for the correction $(\Delta p)^0$. I_1^0 is the restriction operator between the two grids. K^0 may be formed either explicitly after appropriately restricting the history variables, if any, corresponding to the solution d_i from the fine grid or directly by restricting K_i^1, i.e. $K^0 = I_1^{0\mathrm{T}} K_i^1 I_1^0$.

(iii) *Post smoothing.* Perform m_2 relaxation steps on (8) starting from $p^1_{m_1+1}$ which is given by $p^1_{m_1+1} = p^1_{m_1} + I_0^1 (\Delta p)^0$, where I_0^1 is the prolongation operator between the two grids, with $I_0^1 = I_1^{0\mathrm{T}}$.

2.4. *FAS method*

Let us assume a sequence of grids $G^0, G^1, \ldots, G^M$ with mesh sizes $h^0 > h^1 > \cdots > h^M$ each grid approximating the same model geometry on the domain defined by Ω. The objective is to solve directly the discretized version of the non-linear boundary value problem on the finest grid, defined by the equation:

$$r^M \equiv f_{\text{ext}}^M - f_{\text{int}}^M(d^M) = 0 \tag{9}$$

To illustrate the basic idea of FAS method in a simple manner, the case of two grids, say G^0 and G^1, is considered. As in standard multigrid method one basic FAS multigrid cycle is made up of relaxation steps on fine grid G^1, and solution of the approximated residual equation on coarse grid G^0, also known as coarse grid correction.

2.4.1. Pre-smoothing. Perform m_1 non-linear relaxation steps on $f_{\text{ext}}^1 - f_{\text{int}}^1(d^1) = 0$ to obtain $d_{m_1}^1$ and the resulting residual $r_{m_1}^1$.

2.4.2. Coarse grid correction. Solve the non-linear system $f_{\text{ext}}^0 - f_{\text{int}}^0(d^0) = 0$ on grid G^0 for the 'full approximation' d^0 where $f_{\text{ext}}^0 \equiv I_1^0 r_{m_1}^1 + f_{\text{int}}^0(\bar{I}_1^0 d_{m_1}^1)$ is a pseudo-external force for the coarse grid consisting of two parts: (a) the restricted fine grid residual and (b) the internal force corresponding to the injected fine grid displacements. $\bar{I}_1^0$ and I_1^0 are the injection and restriction operators respectively. The computation of part (b) is to be preceded by the formation of current history variables on coarse grid G^0.

2.4.3. Post-smoothing. Perform m_2 non-linear relaxation steps on $f_{\text{ext}}^1 - f_{\text{int}}^1(d^1) = 0$ starting from $d_{m_1+1}^1$ which is the corrected fine grid approximation given by $d_{m_1+1}^1 = d_{m_1}^1 + I_0^1 (d^0 - \bar{I}_1^0 d_{m_1}^1)$.

3. THE FAS-BFGS METHOD FOR NON-LINEAR PROBLEMS

3.1. *The basic idea*

The three basic characteristics of the proposed approach can be briefly summarized as follows.

3.1.1. Full approximation storage. Solve the non-linear system directly, instead of solving a sequence of linearized problems within Newton method, using the FAS philosophy.

3.1.2. Continuous inter- and intra-cycle BFGS updates. We perform continuous BFGS updates on all grids in the basic FAS algorithm. In other words, the conventional non-linear relaxation procedure on the finest and all the intermediate grids is now replaced by a continuous BFGS procedure which exploits the information from previous cycles. This is schematically illustrated in Figure 1. The continuous BFGS smoothing procedures on the finest and all the intermediate grids are started from the diagonal of the Jacobian matrix for the particular grid. A restart of the BFGS procedures is done periodically after a fixed number of cycles. The handling of the solution in the coarsest grid is also different in our method. We perform continuous BFGS updates on the coarsest grid as well. We avoid the computation and factorization of the coarsest grid Jacobian matrix at every multigrid cycle by using the BFGS-updated inverse Jacobian approximation from the previous cycle as the starting approximation for the current cycle and continuing the updates

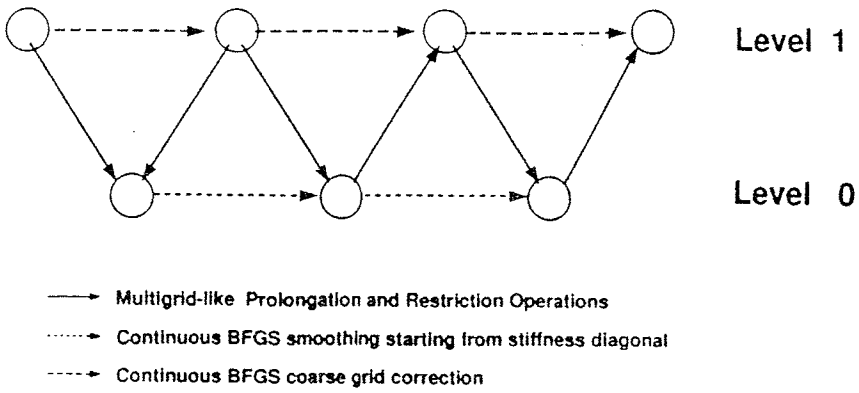

Figure 1. Schematic illustration of continuous BFGS updates

from that point. It should, however, be mentioned that the updates on all grids are not strictly standard BFGS updates in the sense that the current gradient vector and the solution vector may not correspond to the current approximation to the Jacobian matrix on which the update is performed. Mathematically, the equation

$$H_i r_i = \alpha_i (d_{i+1} - d_i) \tag{10}$$

is not satisfied in general. Specifically (10) holds only for the smoothing updates that are performed immediately after a BFGS smoothing iteration on a fine grid. Note that (10) is not a necessary condition for the validity of the BFGS update as long as the secant equation is satisfied.

3.1.3. Consistency of history data. Update the history variables on each grid level directly using their values for the previous load (time) step and the displacements from the current step by performing the integration of the appropriate constitutive relations.

This is a departure from the commonly adopted approach which employs intergrid interpolation to form the auxiliary grid history. Our experience with plasticity problems clearly indicates that consistent formation of the history data on all auxiliary grid levels, which exactly satisfies plasticity-based constrained equations, is essential for the successful application of FAS method to such problems. From the computational standpoint, stress updates at all mesh levels are not a significant overhead, since for 3-D problems the number of Gauss points in the auxiliary grids is reduced by a factor ranging from 7 to 15.

3.2. Formulation of FAS-BFGS approach

3.2.1. Two-grid FAS-BFGS method. As a prelude to the definition of multigrid version of FAS-BFGS method, we now describe one iteration of the two-grid FAS-BFGS method.

Again, for notational simplicity we introduce the following forms:
A concise notation to symbolize a BFGS update, namely,

$$U_{\mathrm{BFGS}}(H_j, u_i, v_i) = (I - \rho_i u_i v_i^{\mathrm{T}}) H_j (I - \rho_i v_i u_i^{\mathrm{T}}) + \rho_i u_i u_i^{\mathrm{T}} \tag{11}$$

where $\rho_i = 1/u_i^{\mathrm{T}} v_i$.

Now to symbolize m BFGS iterations on the system $r \equiv f_{\text{ext}} - f_{\text{int}}(d) = 0$ we use another concise notation $\text{BFGS}^m(H_j, r_i, d_i)$ which stands for m BFGS iterations starting from the initial approximations to the inverse jacobian, H_j, residual r_i and solution d_i that has the following expanded form:

do for $l = 1, 2, \ldots, m$

$$d_{i+l} = H_{j+l-1} r_{i+l-1} + d_{i+l-1} \tag{12}$$

$$r_{i+l} = f_{\text{ext}} - f_{\text{int}}(d_{i+l}) \tag{13}$$

$$H_{j+l} = U_{\text{BFGS}}(H_{j+l-1}, u_{i+l-1}, v_{i+l-1}) \tag{14}$$

$$l = l + 1 \tag{15}$$

Again, for convenience an abbreviated form $\text{UPD}(\eta, d_i)$ is adopted to represent the procedure of integration of the constitutive relations which gives the current values of history variables represented by η_i from the solution increment d_i and the history from the previous step denoted by η. For rate-independent plasticity with isotropic hardening, for example, we may write

$$(\sigma_i, \bar{\varepsilon}_i^{\text{pl}}) = \text{UPD}(\sigma, \bar{\varepsilon}^{\text{pl}}, d_i) \tag{16}$$

where $\text{UPD}(\sigma, \bar{\varepsilon}^{\text{pl}}, d_i)$ stands for the radial-return algorithm[13, 14] to perform the elastoplastic stress update.

Our aim is, as before, to solve $r^1 \equiv f_{\text{ext}}^1 - f_{\text{int}}^1(d^1) = 0$ on grid G^1. A typical FAS-BFGS iteration step for this problem starting from H_i^1, r_i^1, d_i^1 would proceed as follows.

Pre-smoothing. Perform m_1 BFGS relaxation steps on grid G^1.

$$d_{i+m_1}^1 := \text{BFGS}^{m_1}(H_i^1, r_i^1, d_i^1) \tag{17}$$

At the end of the above relaxation we also have the residual

$$r_{i+m_1}^1 := f_{\text{ext}}^1 - f_{\text{int}}^1(d_{i+m_1}^1)$$

Coarse grid correction. Restrict residual, $\bar{r}_i^0 := I_1^0 r_{i+m_1}^1$; inject solution, $d_i^0 = \bar{I}_1^0 d_{i+m_1}^1$; form the current history on coarse grid, $\eta_i^0 = \text{UPD}(\eta^0, d_i^0)$. Solve the non-linear system:

$$f_{\text{int}}^0(d^0) = \bar{r}_i^0 + f_{\text{int}}^0(d_i^0) \tag{18}$$

starting with d_i^0 as the initial approximation.

The right-hand side in (18) can be thought of as a pseudo-external force for the coarse grid, f_{ext}^0. Obviously, if $\bar{r}_i^0 = 0$, $d^0 = d_i^0$, the solution injected from the fine grid. A BFGS procedure is adopted to solve the coarse grid non-linear system:

do for $l = 1, 2, \ldots$ *until convergence*

$$d_{i+l}^0 = H_{i+l-1}^0 r_{i+l-1}^0 + d_{i+l-1}^0$$

$$r_{i+l}^0 = f_{\text{ext}}^0 - f_{\text{int}}^0(d_{i+l}^0)$$

$$H_{i+l}^0 = U_{\text{BFGS}}(H_{i+l-1}^0, u_{i+l-1}^0, v_{i+l-1}^0)$$

$$l = l + 1$$

We denote the converged solution as d^0. The correction that we now will interpolate back to G^1 is $w^0 := d^0 - d_i^0$. Compute the corrected approximation on G^1:

$$d_{i+m_1+1}^1 := d_{i+m_1}^1 + I_0^1 w^0 \tag{19}$$

Perform a BFGS update on G^1 with the new vectors available:

$$H^1_{i+m_1+1} := U_{\text{BFGS}}(H^1_{i+m_1}, u^1_{i+m_1}, v^1_{i+m_1}) \tag{20}$$

Post-smoothing. Perform m_2 BFGS relaxation steps

$$d^1 := \text{BFGS}^{m_2}(H^1_{i+m_1+1}, r^1_{i+m_1+1}, d^1_{i+m_1+1}) \tag{21}$$

thus completing one FAS-BFGS iteration of the two-grid case.

3.2.2. Multigrid FAS-BFGS method. As in the case of FAS method we assume a series of $(M+1)$ grids $G^0, G^1, \ldots, G^M$ with grid sizes $h^0 > h^1 > \cdots > h^M$. We summarize one $(M+1)$-grid step for the solution of the non-linear system $f^M_{\text{ext}} - f^M_{\text{int}}(d^M) = 0$ in the following.
Pre-smoothing. Perform m_1 BFGS relaxation steps on grid G^M.

$$d^M_{i+m_1} := \text{BFGS}^{m_1}(H^M_i, r^M_i, d^M_i) \tag{22}$$

The residual corresponding to this solution is

$$r^M_{i+m_1} := f^M_{\text{ext}} - f^M_{\text{int}}(d^M_{i+m_1})$$

We now approximate the full solution on the coarser grid $(M-1)$ and interpolate back to the fine grid the correction corresponding to the smoothed out part of the error in the following steps.
Coarse grid correction. Restrict residual, $\bar{r}^{M-1}_i := I^{M-1}_M r^M_{i+m_1}$; inject solution, $d^{M-1}_i := \bar{I}^{M-1}_M d^M_{i+m_1}$; form the current values of η on grid G^{M-1}, $\eta^{M-1}_i = \text{UPD}(\eta^{M-1}, d^{M-1}_i)$.

Update the inverse Jacobian approximation on grid G^{M-1} using solution at the end of last cycle:

$$H^{M-1}_i = U_{\text{BFGS}}(H^{M-1}_{i-1}, u^{M-1}_{i-1}, v^{M-1}_{i-1}) \tag{23}$$

Next we have to solve the non-linear system on grid $(M-1)$

$$f^{M-1}_{\text{ext}} - f^{M-1}_{\text{int}}(d^{M-1}) = 0 \tag{24}$$

where f^{M-1}_{ext} is the sum of the restricted residual and the internal force corresponding to the solution injected from grid M:

$$f^{M-1}_{\text{ext}} = \bar{r}^{M-1}_i + f^{M-1}_{\text{int}}(d^{M-1}_i) \tag{25}$$

We achieve this by applying $\gamma \geqslant 1$ steps of M-grid FAS-BFGS method on the grid sequence $G^0, G^1, \ldots, G^{M-1}$. The obvious choice for the initial approximation for d^{M-1} will be d^{M-1}_i. It can be easily verified that, by analogy to the two-grid case, the approximate solution would be arrived at after $m_1 + m_2 + 1$ modifications to d^{M-1}_i which we denote by d^{M-1}. The correction to be interpolated back to grid M is $w^{M-1} := d^{M-1} - d^{M-1}_i$.
Now compute the corrected approximation on G^M:

$$d^M_{i+m_1+1} := d^M_{i+m_1} + I^M_{M-1} w^{M-1} \tag{26}$$

Perform one BFGS update

$$H^M_{i+m_1+1} = U_{\text{BFGS}}(H^M_{i+m_1}, u^M_{i+m_1}, v^M_{i+m_1}) \tag{27}$$

Post-smoothing. Perform m_2 BFGS relaxation steps on grid G^M

$$d^M := \text{BFGS}^{m_2}(H^M_{i+m_1+1}, d^M_{i+m_1+1}, r^M_{i+m_1+1}) \tag{28}$$

which completes one cycle of $M+1$ grid FAS-BFGS method.

It should be emphasized that the history variables represented by η are to be computed directly on each grid by integrating the appropriate constitutive relations. This is one of the key principles

of our solution approach. The only information transferred between the grids are the displacements d and the gradient r. Also note that the updates represented in (20) for two-grid method and (23) and (27) for the multigrid version are characteristic of FAS-BFGS approach, highlighting another basic feature of the method, that is, continuous BFGS updates on all grids. These updates play a major role in the convergence and efficiency of the FAS-BFGS method.

We suggest the following as initial approximations for the case of $(M+1)$ grids:

$$H_0^k = [\text{diag}\nabla(f_{\text{int}}^k)]^{-1}_{d_0^k, \eta_0^k}, \quad k = M, M-1, \ldots, 1 \tag{29}$$

$$H_0^0 = [\nabla(f_{\text{int}}^0)]^{-1}_{d_0^0, \eta_0^0} \tag{30}$$

$$d_0^M = 0, \; r_0^M = f^M \tag{31}$$

Our experience shows that one can accumulate the updates for up to five cycles on the finest and all intermediate grids and up to ten cycles on the coarsest grid without adversely affecting performance. In other words, a restart of BFGS procedure should be done every five cycles for the finest and intermediate grids and every ten cycles for the coarsest grid. This is one of the aspects that should be further investigated in order to develop a mathematically sound criterion for update accumulation that will provide an optimal performance.

4. PERFORMANCE STUDY

In this section we discuss the numerical performance of FAS-BFGS approach as well as a comparison of this new approach with standard non-linear solution methods for some select model problems. The numerical results presented are for small strain, rate-independent elasto-plastic problems although the new solution approach itself is quite general and applicable to other types of nonlinearities. In all the examples the material is assumed to have linear behaviour in the elastic range with Young's modulus E and isotropic linear hardening with hardening modulus H in the plastic zone. Von-Mises yield criterion together with path-independent stress update procedure based on the radial return algorithm is used in the analysis.[15] Consistent or algorithmic tangent stiffness coefficients are employed.[16] The finite element meshes used in the analysis were generated automatically.[18] Unstructured grids have been employed for all the problems solved through multigrid methods in this paper. The intergrid transfer operators were formed according to Reference 17 that uses an efficient search procedure which requires $O(n)$ operations. Our experience indicates that by this technique the cpu time required for the formation of topology between the grids is less than ten percent of the total cpu time for the solution of the corresponding *linear* problem.

4.1. Solution parameters

In order to meaningfully differentiate various methods we specified a demanding tolerance of $1{\cdot}0 \times 10^{-6}$ for the L_2 norm of the normalized gradient vector as the termination criterion. For FAS and FAS-BFGS methods the non-linear system on the coarsest grid level was solved up to a tolerance of $1{\cdot}0 \times 10^{-2}$ at every cycle. In all cases, the full load was applied in one step. The numerical values of $1{\cdot}0 \times 10^6$ and 0·25 were assumed for the Young's modulus E and Poisson's ratio ν, respectively, for all the examples. H/E was assumed to be 0·0155. The fraction, ρ, of numerical integration points that are plastic in the final solution is taken to be indicative of the extent of nonlinearity in the problem domain. All CPU time estimates presented here are for Sun SPARC station 10.

In the continuous BFGS smoothing process for FAS-BFGS method, a restart of BFGS procedure was done every five cycles. The continuous BFGS solution procedure on the coarsest grid was restarted from the current tangent stiffness matrix every 10 cycles.

Unless explicitly stated otherwise, the BFGS solutions for all the problems were started from inverse of the initial stiffness matrix.

The *regula falsi* bisection method was applied to the line search equation (see Section 2) until the criterion (3) is satisfied with $\mu = 0{\cdot}5$. Line search was not attempted if $\alpha = 1$ satisfied the search criterion.

4.2. *Two-dimensional model problem*

Our two-dimensional model problem is a rectangular plate with three holes in plane stress fixed on one of the shorter edges and with a uniform load on the other. The loading results in the development of plastic zones around the three holes due to stress concentration. The yield stress was varied in such a way that a wide range of values of ρ is obtained. The plate was modelled by 8314 bilinear quadrilateral elements with a total of 17636 dofs (see Figure 2). For analysis through Newton-multigrid and FAS-BFGS methods one auxiliary grid was used. The CPU estimates for this problem are given in Table I. The results demonstrate the robustness of FAS-BFGS algorithm. The convergence of the method even in the presence of very high non-linearity (approximately 70 per cent of plasticity and very small hardening slope) is encouraging. The deterioration of the performance of Newton-multigrid method in the case of increased plasticity is also obvious from the results. It is worth mentioning that using the diagonal of the initial stiffness matrix or the complete SSOR matrix as H_0^{-1} with appropriate recomputation and restart for BFGS method failed to converge for all test cases considered here. It can be seen that secant-Newton method with initial tangent stiffness matrix as preconditioner converged in less number of iterations than classical BFGS method without recomputation or restart. However in the secant-Newton method a line search was always necessary at every iteration to satisfy the criterion discussed in Section 2 and this should explain the similar performance of both methods. The large CPU times for FAS method illustrate the extent of computational effort required for smoothing in the case of history-dependent problems. This is further evident from the results for the sample 3-D problems discussed below.

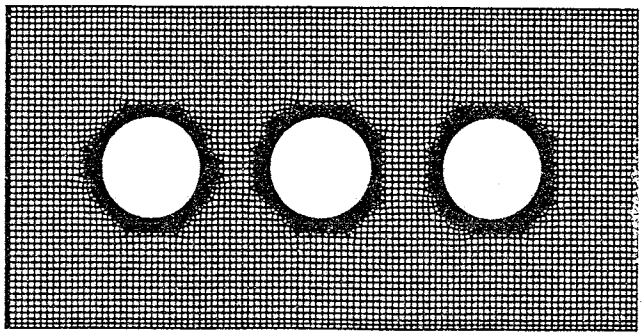

Figure 2. Finest finite element grid for the plate problem

Table I. Performance results for 2-D model problem

Method	ρ	No. of iterations	CPU (sec)	Remarks
FAS-BFGS	0·09	32	557	
FAS-BFGS	0·12	33	587	
FAS-BFGS	0·45	40	995	
FAS-BFGS	0·71	47	1098	
Newton	0·09	7	1087	
Newton	0·45	18	2880	
Newton-multigrid	0·09	9	481	Linear system tolerance 0·1
Newton-multigrid	0·12	8	428	"
Newton-multigrid	0·45	20	1231	"
Newton-multigrid	0·71	**	**	No convergence
FAS	0·09	180	6500	Jacobi smoothing
FAS	0·09	53	2450	Conjugate gradient smoothing
FAS	0·45	200/0·001	10000	". No convergence
BFGS	0·09	55	573	
BFGS	0·45	87	934	
BFGS	0·71	51	950	Restarted from current tangent stiffness matrix after 50 updates
SSOR-BFGS	0·09	200/0·128	1280	Started from initial SSOR matrix, restarted from current SSOR matrix every 10 updates. No convergence.
Secant-Newton	0·09	200/0·615	1893	Initial SSOR preconditioner. No convergence.
Secant-Newton	0·09	44	566	Initial tangent stiffness matrix as preconditioner
Secant-Newton	0·45	74	996	"
Secant-Newton	0·71	81	1161	"

4.3. Three-dimensional example problems

The results presented for the three-dimensional model problems clearly establish the superiority of FAS-BFGS solution method over the popular classical methods for large non-linear systems. The major drawback of the BFGS method stems from the fact that for large 3-D problems factorization of the initial stiffness matrix becomes too expensive, dominating the total solution time. At the same time other choices for H_0^{-1} such as diagonal of the Jacobian matrix or even the complete SSOR matrix resulted in non-convergence under reasonable CPU time. The secant-Newton method with complete SSOR matrix as preconditioner also did not converge for the two examples presented. In the case of Newton-multigrid method, most of the computational labor is spent in the solution of the linear system at every Newton step which becomes prohibitively slow in the presence of even moderate extent of plastic zone in the problem domain.

4.3.1. The problem of two cylinders. Our first three-dimensional example consists of two cylinders meeting at right angles as shown in Figure 3. We fixed all the degrees of freedom on one of the plane faces and applied uniform loading along all the three directions on the remaining three plane faces. The loading takes the solution well beyond elastic limit. The domain was modelled with 4-node tetrahedral elements with 63918 degrees of freedom (Figure 3). We used two auxiliary grids for both FAS-BFGS and Newton-multigrid methods. Analysis through

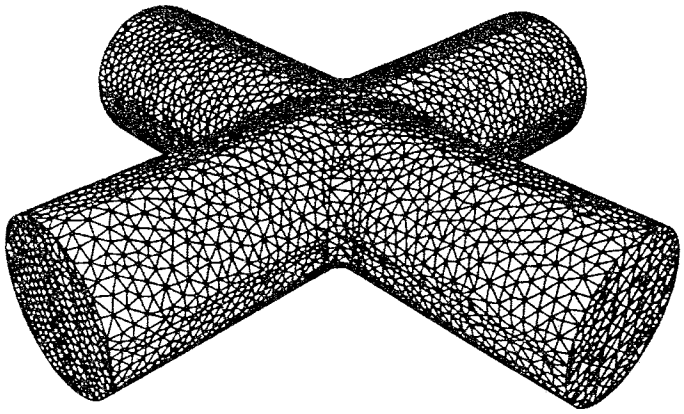

Figure 3. Finest mesh for the two-cylinder problem

classical BFGS and Newton methods would, of course, require only the finest grid. The specified loading and yield stress resulted in approximately 10 per cent of plastic zone in the problem domain. Table II summarizes the results. The advantage of the FAS-BFGS method for this relatively large problem is obvious from the CPU estimates presented.

4.3.2. The problem of Block with cylindrical hole. Next we present the CPU time and storage estimates obtained from the elastoplastic analysis of a solid block with a cylindrical hole, that is fixed on one face and loaded on the opposite face along all the three directions, the plane of the loading being parallel to the axis of the hole. The loading results in stress concentrations around the hole causing development of plastic zones. The source (finest) grid for this problem has 82383 degrees of freedom (Figure 4). As in the first model problem we used three grids for analysis through the two multigrid methods. There was approximately 9 per cent of plastic zone in the model due to the loading. The results are given in Table III. The superior performance of FAS-BFGS method for this problem is evident from the results presented, compared to the existing choices for the analysis of large non-linear systems.

Table IV provides a comparison of the approximate storage requirements for various methods for the Block problem. As expected, FAS and FAS-BFGS methods required less storage than all the other methods. The storage for FAS-BFGS method is slightly higher than that for FAS because of the memory required for the update vectors during continuous BFGS smoothing.

Table II. Comparison of CPU times for the two-cylinder problem

Method	No. of iterations	CPU (h)	Remarks
Newton	8	48	Projected time
FAS	60	3·80	Conjugate gradient smoothing
BFGS	60	7	Projected time, if started with initial tangent stiffness matrix
Newton-multigrid	16	5·18	Linear system tolerance 0·1
Newton-multigrid	9	4·37	Linear system tolerance 0·01
FAS-BFGS	60	1·82	

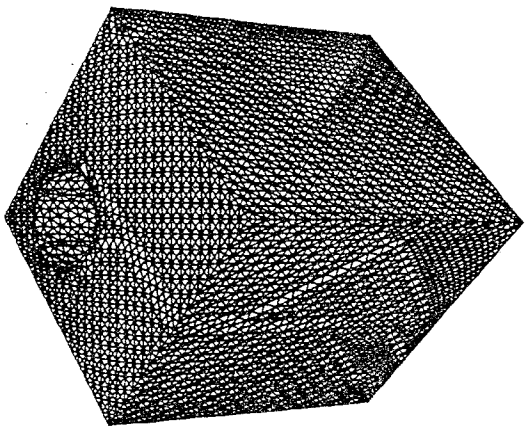

Figure 4. Finest mesh for the block problem

Table III. Comparison of CPU times for the Block problem

Method	No. of iterations	CPU (h)	Remarks
Newton	8	96	Projected time
FAS	72	6·81	Conjugate gradient smoothing
BFGS	40	15	Projected time, if started with initial tangent stiffness matrix
SSOR-BFGS	200/0·0467	5	Started from initial SSOR matrix, restarted from current SSOR matrix every 10 updates. No convergence.
Secant-Newton	200/0·1503	6·5	SSOR preconditioner, recomputed every 5 iterations. No convergence.
Newton-multigrid	8	3·67	Linear system tolerance 0·01
FAS-BFGS	36	1·32	

Table IV. Comparison of storage requirements for the block problem

Method	Memory used (in MB)	Remarks
Newton	500	Projected memory
BFGS	500	Projected memory
SSOR-BFGS	100	Compact storage (only non-zero terms of jacobian)
Newton-multigrid	100	"
Secant-Newton	100	SSOR preconditioner with compact storage
FAS	50	
FAS-BFGS	60	

5. CONCLUSIONS

A new approach has been presented for the solution of non-linear history-dependent finite element equations. It combines FAS multigrid method with BFGS procedure to form an efficient solution technique for large scale non-linear systems. The performance of the method for two-dimensional problems is shown to be comparable to that of the popular methods employed for the solution of such problems. On the other hand, for large three-dimensional systems the numerical results presented clearly demonstrate the advantages of FAS-BFGS method over existing techniques. Furthermore, unlike the common trend in the solution of large non-linear systems where there is a trade-off between speed of convergence and memory occupied by the solution method, the proposed method requires minimal storage without sacrificing the speed of convergence.

ACKNOWLEDGEMENT

The support of the National Science Foundation NYI award ECS-9257203 is gratefully acknowledged.

REFERENCES

1. A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *Math. Comput.*, **31**, 333–390 (1977).
2. H. Matthies and G. Strang, 'The solution of nonlinear finite element equations', *Int. j. numer. methods eng.*, **14**, 1613–1626 (1979).
3. M. Papadrakakis and V. Balopoulos, 'Improved quasi-Newton methods for large non-linear problems', *J. Eng. Mech.*, **117**, 1201–1219 (1991).
4. D. J. Evans, 'The use of preconditioning in iterative methods for solving linear equations with symmetric positive definite matrices', *J. Inst. Math. Appl.*, **4**, 295–314 (1967).
5. M. Papadrakakis and C. J. Gantes, 'Preconditioned conjugate- and secant-Newton methods for nonlinear problems', *Int. j. numer. methods eng.*, **28**, 1299–1316 (1989).
6. B. Irons and A. Elsawaf, 'The conjugate-Newton algorithm for solving finite element equations', in K. J. Bathe *et al.* (eds), *Proc. U. S.–German Symp. on Formulation and Algorithms in Finite Element Analysis*, MIT, 1977, pp. 656–672.
7. M. A. Crisfield, 'Incremental/iterative solution procedures for nonlinear structural problems', in C. Taylor *et al.* (eds.), *Numerical Methods for Nonlinear Problems*, Pineridge Press, Swansea, 1980, pp. 261–290.
8. D. F. Shanno, 'Conjugate gradient methods with inexact searches', *Math. Oper. Res.*, **3**, 244–256 (1978).
9. M. Papadrakakis and P. Ghionis, 'Conjugate gradient algorithms in nonlinear structural analysis problems', *Comput. Methods Appl. Mech. Eng.*, **59**, 11–27 (1986).
10. S. Kacou and I. D. Parsons, 'A parallel multigrid method for history-dependent elastoplasticity computations', *Comput. Methods Appl. Mech. Eng.*, **108**, 1–21 (1993).
11. H. Kardestuncer *et al.* (eds.), *Finite Element Handbook*, McGraw-Hill, New York, 1987.
12. J. E. Dennis and J. Moré, 'Quasi-Newton methods: motivation and theory', *SIAM Rev.*, **19**, 46–89 (1977).
13. M. L. Wilkins, 'Calculation of elastic-plastic flow', in B. Alder *et al.* (eds.), *Methods of Computational Physics, Vol. 3*, Academic Press, New York, 1964.
14. R. D. Krieg and S. W. Key, 'Implementation of a time-independent plasticity theory into structural computer programs', in J. A. Stricklin and K. J. Saczalski (eds.), *Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects*, AMD-20, ASME, New York, 1976, pp. 125–137.
15. T. J. R. Hughes, 'Numerical implementation of constitutive models: rate-independent deviatoric plasticity', in S. Nemat-Nasser *et al.* (eds.), *Theoretical foundation for large-scale computations of nonlinear material behavior: Proceedings of the Workshop on the Theoretical Foundation for Large-Scale Computations of Nonlinear Material Behaviour*, Northwestern University, 1983, pp. 29–57.
16. J. C. Simo and R. L. Taylor, 'Consistent tangent operators for rate-independent elastoplasticity', *Comput. Methods Appl. Mech. Eng.*, **48**, 101–118 (1985).
17. R. Lohner and K. Morgan, 'An unstructured multigrid method for elliptic problems', *Int. j. numer. methods eng.*, **24**, 101–115 (1987).
18. M. S. Shephard and M. K. Georges, 'Automatic three-dimensional mesh generation by the Finite Octree Technique', *Int. j. numer. methods eng.*, **32**, 709–749 (1991).