Iterative Solution Techniques for the Quasi-continuum Method

Yu Liang, [1] Ramdev Kanapady [2] and Peter Chung [3]

Summary

Multi-scale approaches in computational mechanics have recently received much attention in several branches of the physical sciences. The quasi-continuum (QC) method, in particular, furnishes a computational scheme for seamlessly bridging the small atomistic scale systems at zero temperature to a deforming continuum, which permits a reduction of the full set of atomistic degrees of freedom. The present paper surveys iterative solution techniques to determine the stable equilibrium configurations of a deforming crystalline material. The objective is to study solvers in order to optimize computational performance. Due to the implicit solution technique employed by the QC method and the strongly non-convex nature of the potential energy surface with vast numbers of metastable configurations, one of the keys to its success is the use of an effective and efficient iterative solution technique. The two iterative solution techniques studied presently are the nonlinear conjugate gradient method and the Newton method. The performance of various preconditioners employing approximate and exact Hessians, and various line search methods such as backtracking are illustrated for a nano-indentation problem in 2-D and 3-D situations.

Introduction

The QC method is commonly employed to solve a wide variety of multi-scale problems. Typically problems of interest are those involving a small number of defects such as dislocations, a crack, or grain boundary interactions. In this paper, iterative solution techniques are explored for the QC methods in 2-D and 3-D situations. The fully non-local QC method with variable clusters [2] is employed in the 3-D situations and the local/non-local variation [1] is employed for the 2-D situations. These two formulations are briefly described first before detailing out the iterative techniques to minimize the total potential energy arising out of these formulations. Readers are referred to the original papers [2,3] for further details.

Brief Overview of the QC method: The key aspect of accurate representation of real materials by interatomic potentials is that the interaction energies extend beyond nearest neighbor atoms. Thus interatomic potentials are non-local. Given the atomic potentials as a function of the atomic coordinates $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, the total potential energy admits additive decomposition of the energy of individual atoms,

$$E_a = \sum_i E_i \quad \Rightarrow \quad \mathbf{f}_i = -\frac{\partial E_a(\{\mathbf{x}_1,\ldots,\mathbf{x}_N\})}{\partial \mathbf{x}_i} \tag{1}$$

where $\mathbf{f}_i$ is the force on each atom in the absence of externally applied forces. For practical implementation, potentials possessing a cutoff radius over the first few neighboring atoms is preferred. In atomistic calculations there are no definitions of the continuum concepts of strain or displacement. The motion of individual atoms are tracked without reference to their original positions. On the other hand in continuum mechanics it assumes that a strain energy density functional W exists. Hence, the energy in an incremental volume dV around a point $\mathbf{X}$ is $W(\mathbf{X})dV$. In addition, the QC method is allowed because the constitutive behavior is assumed to remain unchanged regardless of representation by atoms or continuum.

The QC formulation that is entirely "non-local" was proposed recently by Knap and Ortiz [2] in that the locality of finite element (FE) computations is effectively removed. This method appears promising as it is truly seamless since it has eliminated the effects of the atomistic to continuum transition regions. The key is to employ FE concepts to kinematically constrain some atomic positions to node positions through the Cauchy-Born rule and to determine forces from a fully non-local atomistic description in all regions of space. The forces are calculated from clusters containing representative atoms that correspond to FE nodes, which are chosen to be the degrees of freedom of the

[1] yliang@cs.umn.edu, Army High Performance Computing Center (AHPCRC), Ph: 612-626-8605

[2] ramdev@me.umn.edu, AHPCRC, University of Minnesota, 111 Church St. S.E. Minneapolis, MN 55455, USA., Ph: 612-626-8101, Fax: 612-626-1596

[3] pchung@arl.army.mil, U.S Army Research Laboratory, Ph:410-278-6027 Fax: 410-278-4983

Proceedings of the 2004 International Conference on Computational & Experimental Engineering & Science
26-29 July, 2004, Madeira, Portugal

problem. Small clusters of atoms surrounding the representative atoms are constructed by interpolating the displacements of the FE nodes and deforming the perfect lattice accordingly. If $\mathbf{X}$ is the reference positions of an atom and $\mathbf{U}_j$ is its interpolated displacements, employing FE interpolation functions N_j allows for the positions of all the atoms $\mathbf{x}$ to be computed. Thus via derivatives of energy (Eq. 1), the force on node j becomes

$$\mathbf{f}_j \equiv -\frac{\partial E^a}{\partial \mathbf{U}_j} = -\sum_{i=1}^{N} \frac{\partial E_i(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{U}_j} \tag{2}$$

Employing the identity $\frac{\partial \mathbf{u}}{\partial \mathbf{U}_j} = N_j \mathbf{I}$ and the displacement interpolation $\mathbf{u}(\mathbf{X}) = \sum_{j=1}^{N} \mathbf{U}_j N_j(\mathbf{X})$, and the above summation is approximated by summation over a smaller cluster around each representative atom (i.e., node)

$$\mathbf{f}_j = -\sum_{i=1}^{N} \frac{\partial E_i(\mathbf{u})}{\partial \mathbf{u}} N_j \approx -\sum_{i}^{N} w_i \left[\sum_{c \in C_i} \mathbf{p}_c N_j(\mathbf{X}_c) \right]; \quad \mathbf{p}_c = \frac{\partial E^a}{\partial \mathbf{u}_c} \tag{3}$$

where C_j refers to the set of atoms in the cluster around node j, $\mathbf{p}_c$ is the atomic-level force experienced by cluster atom c in displacement field $\mathbf{u}$, and w_i is a appropriate weight function for node i to account for the variable density of nodal points throughout the model. The appropriate Hessian can be computed accordingly.

The 2-D QC formulation employed in this study involves “local” and “non-local” regions separated by a transition region. The non-local region is the atomistic region in which every atom is explicitly represented. These atoms are treated using interatomic potentials. In the local region, there is a FE mesh in which, typically, some of the nodes are mapped on atomic lattice sites. There is a one-to-one correspondence between atoms and nodes on the FE mesh at the interface of the transition region. In the continuum region, the FE nodes become sparse and elements become larger so as to completely fill the physical space. In addition, on the continuum side of the interface there exists so-called corrective ghost forces due to loss of central force symmetry across the interface. The total force and the Hessian are derived by explicit differentiation of the energy functional Eq. 1 which are then modified by the FE approximation and corrected with ghost forces such that ultimately [3]

$$\mathbf{f}_\alpha = \frac{\partial \Pi_h}{\partial \mathbf{u}_\alpha} = \sum_{e=1}^{M} \upsilon_e P(\mathbf{F}_e) \nabla N_\alpha(\mathbf{X}_e) - \sum_{\beta=1}^{R_{NL}} \left[\sum_{j=1}^{m_\beta} \varphi_\beta^j N_\alpha(\mathbf{X}_\beta^j) \right] + \sum_{j=1}^{m_\beta} \varphi_\beta^j - n_\alpha(\bar{\mathbf{f}}_\alpha + \mathbf{f}_\alpha^G) \tag{4}$$

$$\begin{aligned}\mathbf{H}_{\alpha\beta} = {} & \sum_{e=1}^{M} \upsilon_e \mathbf{C}(\mathbf{F}_e) \nabla N_\alpha(\mathbf{X}_e) \nabla N_\beta(\mathbf{X}_e) + \sum_{\gamma=1}^{R_{NL}} \left[\sum_{k=1}^{m_\gamma} \sum_{l=1}^{m_\gamma} \mathbf{K}_\gamma^{kl} N_\alpha(\mathbf{X}_\gamma^k) N_\beta(\mathbf{X}_\gamma^l) \right] \\ & - \sum_{k=1}^{m_\alpha} \sum_{l=1}^{m_\alpha} \mathbf{K}_\alpha^{kl} N_\beta(\mathbf{X}_\alpha^l) - \sum_{k=1}^{m_\beta} \sum_{l=1}^{m_\beta} \mathbf{K}_\beta^{kl} N_\beta(\mathbf{X}_\beta^k) + \delta_{\alpha\beta} \sum_{k=1}^{m_\alpha} \sum_{l=1}^{m_\alpha} \mathbf{K}_\alpha^{kl}\end{aligned} \tag{5}$$

where $\mathbf{C} = \frac{\partial^2 E}{\partial \mathbf{F}^2}$ is Lagrangian tangent stiffness tensor, $\mathbf{K}_\beta^{kl} = \frac{\partial^2 E_\beta}{\partial \mathbf{r}_\beta^k \mathbf{r}_\beta^l}$ is the atomic level stiffness matrix, $\mathbf{P} = \frac{\partial E}{\partial \mathbf{F}}$ is the first Piola-Kirchhoff stress tensor and $\mathbf{X}_e$ is the element centroid. The design of both the total energy and the forces in these QC formulations permit application of gradient-type iterative solution methods to minimize the total energy to obtain the static, zero-temperature equilibrium atomic configuration.

Solvers for Unconstrained Minimization

In order to obtain the equilibrium configuration of the solid, unconstrained minimization of the total potential energy $\Pi(\mathbf{u})$ needs to be performed. Let $\mathbf{g} = \nabla\Pi(\mathbf{u})$, $\mathbf{H} = \nabla^2\Pi(\mathbf{u})$, and $\mathbf{B} \approx \mathbf{H}^{-1}$ be the gradient, the Hessian and approximate inverse of the Hessian, respectively. Then a general iterative solver framework for unconstrained minimization can be described by Algorithm 1 below. It is evident from Algorithm 1 that if only the gradient $\nabla\Pi(\mathbf{u})$ is available then Quasi-Newton [4], steepest descent, and non-linear conjugate gradient [5,6] are applicable. It should be noted that, as an approximation to $\mathbf{H}^{(k)^{-1}}$, $\mathbf{B}^{(k)}$ is constructed based on the gradient. While the Newton-Raphson method has better convergence properties the computation of an exact Hessian matrix is time-consuming and may require large amounts of storage for large scale problems. In Algorithm 1, convergence is said to occur if $\|\mathbf{g}_k\| < \varepsilon$ is achieved.

Algorithm 1: General iterative solver framework

```
Initialization: $\mathbf{u}_0$ is given;
Relax Iteration:
    $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k,\ k = 0, \cdots$ , where $\mathbf{d}_k$ for:
        Newton-Raphson: $\mathbf{d}_k = -(\mathbf{H}_k)^{-1}\mathbf{g}_k$
        Quasi-Newton: $\mathbf{d}_k = -\mathbf{B}_k \mathbf{g}_k$
        Steepest-descent: $\mathbf{d}_k = -\mathbf{g}_k$
        Conjugate Gradient: $\mathbf{d}_k = -\mathbf{g}_k + \beta \mathbf{d}_k$ where $\beta$ for:
            Fletcher-Reeves method: $\beta_k^{FR} = \left(\frac{\|\mathbf{g}_k\|_2}{\|\mathbf{g}_{k-1}\|_2}\right)^2$
            Polak-Ribiere method: $\beta_k^{PR} = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{g}_{k-1}, \mathbf{g}_{k-1} \rangle}$
            or Hestenes-Stiefel method: $\beta_k^{HS} = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{d}_{k-1}, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}$
```

Preconditioned Non-linear CG: Nonlinear conjugate gradient methods are motivated by the success of the linear conjugate gradient method in minimizing quadratic functions with positive definite Hessians. Nonlinear conjugate gradient methods are of the form

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k \tag{6}$$

where $\alpha_k > 0$ is the step-length and $\mathbf{d}_k$ is search direction. Normally the search direction at the first iteration is the steepest descent direction, namely, $\mathbf{d}_0 = -\mathbf{g}_0$, the other search directions can be defined recursively:

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k. \tag{7}$$

For nonlinear problems, performance is problem dependent, but these methods have the advantage that they require only gradient evaluations and memory requirements are minimal making this a popular class of algorithms for large-scale optimization. These algorithms can be derived as extensions of the conjugate gradient algorithm or as specializations of limited-memory quasi-Newton methods. Among Fletcher-Reeves, Polak-Ribiere and Hestenes-Stiefel method, the Polak-Ribiere non-linear conjugate method appears to have the best convergence performance and is described in Algorithm 2. An efficient preconditioner of non-linear preconditioned conjugate gradient (PCG) should approximate the inverse of the Hessian. For its construction, one can refer to the Quasi-Newton method described subsequently. For the Broyden-Fletcher-Goldfarb (BFGS) method with preconditioner $\mathbf{B}_{k+1}(\mathbf{B}_0 = \mathbf{I})$ at $k+1$, iterations involve the following preconditioning operation:

$$\mathbf{q}_{k+1} = \mathbf{B}_{k+1}\mathbf{r}_k \tag{8}$$

which can be computed via

$$\mathbf{q}_{k+1} = \mathbf{r}_k + \frac{\langle \mathbf{r}_k, \mathbf{s}_k \rangle \mathbf{y}_k + \langle \mathbf{r}_k, \mathbf{y}_k \rangle s_k}{\langle \mathbf{s}_k, \mathbf{y}_k \rangle} - \frac{\langle \mathbf{r}_k, \mathbf{s}_k \rangle}{\langle \mathbf{s}_k, \mathbf{y}_k \rangle}(1 + \frac{\langle \mathbf{y}_k, \mathbf{y}_k \rangle}{\langle \mathbf{s}_k, \mathbf{y}_k \rangle})\mathbf{s}_k. \tag{9}$$

The bracket notation refers to the standard scalar product of two vectors.

Algorithm 2: Polak-Ribiere Conjugate Gradient with given tolerance ε:

```
(1) Initialize $\mathbf{u}_0$; $\mathbf{d}_0 = -\mathbf{g}_0$;
(2) FOR ($k = 1, \cdots, k_{\max}$) DO
(3) Compute $\alpha$ such that $\min \Pi(\mathbf{u}_{k-1} + \alpha \mathbf{d}_{k-1})$; /* line search */
(4) $\mathbf{u}_k = \mathbf{u}_{k-1} + \alpha \mathbf{d}_{k-1}$;
(5) IF ($\|\mathbf{g}_k\| \leq \varepsilon$) THEN exit;
(6) $\beta_k = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{g}_{k-1}, \mathbf{g}_{k-1} \rangle}$; /* Polak-Ribiere method */
(7) $\mathbf{d}_k = -\mathbf{B}_k \mathbf{g}_k + \beta_k \mathbf{d}_{k-1}$;
(8) ENDFOR
```

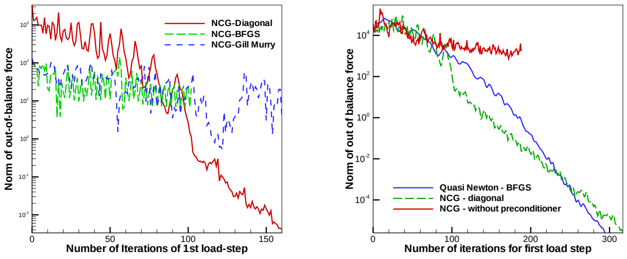

Figure 1: Performance of preconditioned nonlinear CG for 3-D nanoindentation problem.

Line-search Algorithm: Define $\Pi(\alpha) = \Pi(\mathbf{u}+\alpha\mathbf{d})$ where $\mathbf{d}$ and $\mathbf{u}$ are given. This potential energy can be redefined as a univariate function as $\phi(\alpha) = \Pi(\mathbf{u}_k + \alpha\mathbf{d}_k)$ which transforms $\min_{\mathbf{x}} \Pi(\mathbf{x})$ into the minimizing problem $\min_{\alpha\in[\ell,\hbar]} \phi(\alpha)$. Two main line-search methods are Newton-Raphson (where $\mathbf{H}_k$ is needed) approach and backtracking approach. In backtracking method the step length control in the line search algorithm is mainly governed by the Wolfe Conditions which requires two conditions to be satisfied. First, requires that to avoid over-approximating the step length $\Pi(\mathbf{u}_k+\alpha\mathbf{d}_k) \le \Pi(\mathbf{u}_k) + c_1\alpha\langle\mathbf{g}_k\mathbf{d}_k\rangle$ where $c_1 = 10^{-4}$ which is based on the sufficient decrease condition. Secondly, to avoid under-approximating the step length, a condition is placed on the curvature and is given by $\langle g(\mathbf{u}_k+\alpha_k\mathbf{d}_k), \mathbf{d}_k\rangle \ge c_2\langle\mathbf{g}_k, \mathbf{d}_k\rangle$ where $c_2 = 0.9$. A backtracking line-search method is described in Algorithm 3.

Algorithm 3: Backtracking Line-Search: given $\alpha_0 > 0$ and γ_1, γ_2 satisfying $0 < \gamma_1 < \gamma_2 < 1$

```
(1) $k = 0$;
(2) While $\Pi(\mathbf{u}_k + \alpha_k\mathbf{d}_k) > \Pi(\mathbf{u}_k) + c_1\alpha_k\langle\mathbf{g}_k, \mathbf{d}_k\rangle$
(3) Compute $\alpha_{k+1} \in [\gamma_1\alpha_k, \gamma_2\alpha_k]$ such that Wolfe conditions are satisfied;
(4) $k = k+1$;
(5) ENDWhile
```

Statement (3) in Algorithm 3 is implemented via the bisection method, gold section section or polynomial interpolation method. In our study, three point cubic interpolation methods are employed which are described here briefly. Assume $\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d$ that satisfy $\phi_c(0) = \phi(0)$ $\phi_c'(0) = \phi'(0)$, $\phi_c(\alpha_k) = \phi(\alpha_{k-1})$ and $\phi_c(\alpha_k) = \phi(\alpha_k)$. Then, $\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \phi'(0)\alpha + \phi(0)$ where

$$\begin{bmatrix} a \\ b \end{bmatrix} = \theta \begin{bmatrix} & -\alpha_k^2 \\ -\alpha_{k-1}^3 & \alpha_k^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_k) - \phi(0) - \phi'(0)\alpha_k \\ \phi(\alpha_{k-1}) - \phi(0) - \phi'(0)\alpha_{k-1} \end{bmatrix}, \tag{10}$$

and $\theta = 1/(\alpha_{k-1}^2\alpha_k^2(\alpha_k - \alpha_{k-1}))$. Using $\phi_q'(\alpha) = 0$ such that $\phi_c(\alpha)$ is minimized, it follows that $\alpha_{k+1} = \frac{b-\sqrt{b^2-3a\phi'(0)}}{3a}$.

Remarks: The motivation for this paper comes from the results of Figure 1. The performance of the presently employed nonlinear CG method [2] with various preconditioners was evaluated using a large test specimen made of FCC gold with a spherical indentor size of 70 nm. The specimen is significantly larger than that used in the numerical experiments section later in the paper to demonstrate the large iteration numbers required for typical QC problems over a single load step. Clearly the preconditioning results are better with than without preconditioning. However, all of the preconditioning methods resulted in similar convergence results while a diagonal preconditioner produced marginally better results. Hence, the goal is to explore alternative iterative strategies to reduce the number of iterations by at least one order of magnitude. One such alternative is the Newton-Raphson technique which is described next.

Newton-Raphson Method : Approximating $\Pi(\mathbf{u}_{k+1})$ by Taylor series yields

$$\Pi(\mathbf{u}_k + \alpha_k\mathbf{d}_k) \approx \bar{\Pi}(\mathbf{d}_k) = \Pi(\mathbf{u}_k) + \alpha_k\mathbf{d}_k^T\mathbf{g}_k + \frac{\alpha_k^2}{2}\mathbf{d}_k^T\mathbf{H}_k d_k \tag{11}$$

Since $\mathbf{H}_k$ is symmetric by definition, the search direction $\mathbf{d}_k$ can be found as the minimizer of $\bar{\Pi}(\mathbf{d}_k) \Rightarrow \frac{\partial\bar{\Pi}}{\partial\mathbf{d}_k} = \alpha_k\mathbf{g}_k + \alpha_k^2\mathbf{H}_k\mathbf{d}_k$. Then, the optimal search direction is given by $\alpha_k\mathbf{d}_{k+1} = -\mathbf{H}_{k+1}^{-1}\mathbf{g}_k$ such that $\bar{\Pi}(\mathbf{d}_k)$ is minimized.

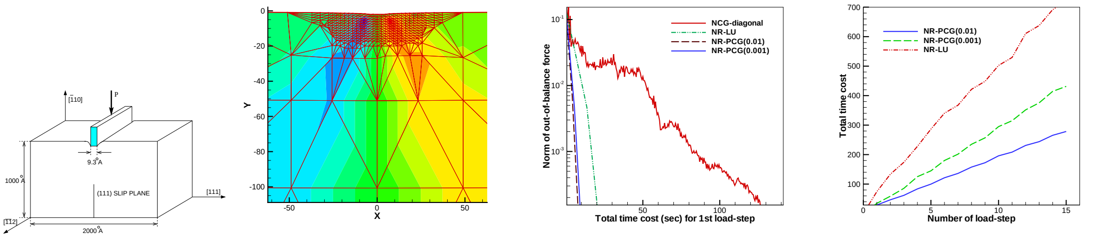

Figure 2: (a) Nanoindentation into single crystal aluminum with an EAM potential, (b) results for 10th load step, (c) – (d) performance of nonlinear CG and Newton-Raphson method.

The resulting Newton-Raphson method is described in Algorithm 4. The computationally intensive part of Algorithm 4 is the Statement (8). For large scale problems, due to the symmetry and sparsity of $\mathbf{H}$, a linear preconditioned conjugate gradient algorithm (PCG) is employed to solve $\mathbf{Hd} = \mathbf{g}$. Using the Hessian matrix, the optimal search direction can be obtained through the Newton-Raphson method. However, the $O(n^2)$ memory requirements and $O(n^3)$ associated with solving a linear system directly have restricted Newton-Raphson methods only to (1) small-scale problems, (2) problems with special sparsity patterns, or (3) if the initial guess is in the vicinity of the solution. As an alternative, Quasi-Newton and discrete Newton and truncated Newton methods can be employed. The key to the above methods is to obtain an approximate inverse of the Hessians matrix, i.e., $\mathbf{B} \approx \mathbf{H}^{-1}$.

Quasi-Newton Method: The Quasi-Newton method builds up an approximation to the Hessian by keeping track of the gradient differences along each step taken by the algorithm. Various conditions are imposed on the approximate Hessian. For example, its behavior along the step just taken is forced to mimic the behavior of the exact Hessian, and it is usually kept positive definite. Let $\mathbf{s}_k = \mathbf{u}_{k-1} - \mathbf{u}_k$ and $\mathbf{y}_k = \mathbf{g}_{k-1} - \mathbf{g}_k$, then four methods can be described in Table 1.

Algorithm 4:Newton-Raphson algorithm with given tolerance ε:

```
(1) Initialize $\mathbf{u}_0$;
(2) $\mathbf{d}_0 = -\mathbf{g}_0$;
(3) FOR ($k = 0, \cdots, k_{\max}$) DO
(4) $arg\min_\alpha \Pi(\mathbf{u}_k + \alpha\mathbf{d}_k)$; /* line search */
(5) $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha\mathbf{d}_k$;
(6) update $\mathbf{H}_{k+1}$ and $\mathbf{g}_{k+1}$;
(7) IF ($\|\mathbf{g}_k\| \leq \varepsilon$) THEN exit;
(8) $\mathbf{H}_{k+1}\mathbf{d}_{k+1} = -\mathbf{g}_{k+1}$; /* search direction */
(9) ENDFOR
```

Table 1: Approximate Hessian

Broyden-Fletcher-Goldfarb (BFGS):	$\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k + \frac{\mathbf{g}_k \times \mathbf{g}_k}{\langle \mathbf{g}_k, \mathbf{d}_k \rangle} + \frac{\mathbf{y}_k \times \mathbf{y}_k}{\langle \mathbf{y}_k, \mathbf{s}_k \rangle}$
Powell-Symmetric (PSB):	$\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k) \times \mathbf{s}_k + \mathbf{s}_k \times (\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)}{\langle \mathbf{s}_k, \mathbf{s}_k \rangle} + \frac{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k) \times \mathbf{s}_k}{\langle \mathbf{s}_k, \mathbf{s}_k \rangle^2} (\mathbf{s}_k \times \mathbf{s}_k)$
Davidson-Fletcher-Powell (DFP):	$\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k - \frac{(\mathbf{B}_k\mathbf{s}_k) \times (\mathbf{B}_k\mathbf{s}_k)}{\langle \mathbf{s}_k, \mathbf{s}_k \rangle_{\mathbf{B}_k}} + \frac{\mathbf{y}_k \times \mathbf{y}_k}{\langle \mathbf{y}_k, \mathbf{s}_k \rangle} + \langle \mathbf{s}_k, \mathbf{s}_k \rangle_{\mathbf{B}_k} (\mathbf{w}_k \times \mathbf{w}_k)$,
	where $\mathbf{w}_k = \frac{\mathbf{y}_k}{\langle \mathbf{y}_k, \mathbf{s}_k \rangle} - \frac{\mathbf{B}_k\mathbf{s}_k}{\langle \mathbf{s}_k, \mathbf{s}_k \rangle_{\mathbf{B}_k}}$
Gill and Murray:	$\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k + \frac{1}{\mathbf{y}_k^\mathsf{T}\mathbf{s}_k} \left(\left(-\mathbf{B}_k\mathbf{y}_k\mathbf{s}_k^\mathsf{T} + \mathbf{s}_k\mathbf{y}_k^\mathsf{T}\mathbf{B}_k \right) + \left(1 + \frac{\mathbf{y}_k^\mathsf{T}\mathbf{B}_k\mathbf{y}_k}{\mathbf{y}_K^\mathsf{T}\mathbf{s}_k} \right) \mathbf{s}_k\mathbf{s}_k^\mathsf{T} \right)$

Numerical Experiments

Two sets of numerical test calculations were performed. The first is 2-D using the QC method whose software was made available through a freely distributed package [1] with the example problem of a rigid flat-nosed knife indenter also contained therein. The material considered is single crystal aluminum modeled with an EAM potential indenting the $(\bar{1}10)$ surface. The second calculation is 3-D using the QC cluster-based method originally proposed in [2]. The material in that example is a [100] surface of unrelaxed single crystal EAM gold subjected to a spherical indenter.

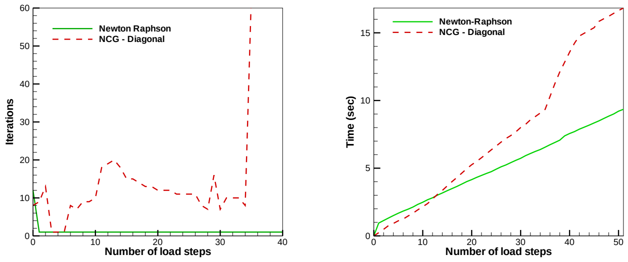

Figure 3: Performance of Newton-Raphson method with Hessian 3-D nano indentation of unrelaxed single crystal EAM gold problem.

Figure 2(a) depicts the 2-D problem of the rigid knife-like flat indentor into single crystal aluminum. The simulation results for the 10th load step are shown in Fig. 2 employing Newton-Raphson technique with preconditioned conjugate gradient method to solve search direction employing the exact Hessian (Eq. 5). The nonlinear conjugate gradient method [1,3] did not converge for the first load step in this case and its corresponding wall-clock time is therefore not reported in Figure 2d. The convergence results of the Newton-Raphson for 10 load steps are shown in Figure 2 (c)–(d). This reduces the number of iterations by order of magnitude for the local/non-local QC method for in 2-D situations. With this results the efforts were carried out for fully non-local QC method in 3-D situations.

The preliminary results with the appropriate computations of the Hessian from the force (Eq. 3) are shown in Fig. 3 for the 3-D unrelaxed single crystal problem. The search direction is recomputed in each iteration of the Newton-Raphson technique. From the figure it is clear that the number of iteration is reduced by an order of magnitude compared to the nonlinear CG with diagonal preconditioner [2]. However, because of the use of a simple unrelaxed test model, the results are as of yet inconclusive. Initial surface relaxations are underway, and additional results will be presented in the final paper.

Acknowledgement

The authors gratefully acknowledge J. Knap and M. Ortiz for numerous discussions on issues related to solver performance and for providing the necessary 3-D code to perform the performance tests. YL and RK are very pleased to acknowledge support in part by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under contract number DAAD19-01-2-0014. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Special thanks are due for additional support to the Computational & Information Sciences Directorate, High Performance Computing Division at the U. S. Army Research Laboratory (ARL), Aberdeen Proving Ground, Maryland. Other related support in the form of computer grants from the Minnesota Supercomputer Institute (MSI), Minneapolis, Minnesota is also gratefully acknowledged.

Reference

1. Quasicontinuum, http://www.qcmethod.com.
2. J. Knap and M. Ortiz, An Analysis of the Quasicontinuum Method. *J. Mech. Phys. Solids*, 49, 1899-1923, 2001.
3. V. B. Shenoy, R. Miller, E. B. Tadmor, D. Rodney R. Phillips, M. Ortiz, An Adaptive Finite Element Approach to Atomic-Scale Mechanics - The Quasicontinuum Method, *J. Mech. Phys. Solids*, 47. 611-642, 1998.
4. Jr. Dennis and J. J. More, Quasi-Newton methods, Motivation and Theory, *SIAM Rev. 19, 46–89*, 1974.
5. A. Buckley and A. LeNir, QN-like Variable Storage Conjugate Gradients, *Mathematical Programming*, 27, 155–175, 1983.
6. P. Concus, et. al., A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations, *Sparse Matrix Computation*, J.R.Bunch and D.J.Rose, Eds., Academic Press, New York, 1976