An Efficient Numerical Method With a Parallel Computational Strategy for Solving Arbitrarily Shaped Inclusions in Elastoplastic Contact Problems

The plastic zone developed during elastoplastic contact may be effectively modeled as an inclusion in an isotropic half space. This paper proposes a simple but efficient computational method to analyze the stresses caused by near surface inclusions of arbitrary shape. The solution starts by solving a corresponding full space inclusion problem and proceeds to the surface of the half space is chosen to validate the method. When the surface truncation domain is extended sufficiently and the grids are dense enough, the results based on the new approach are in good agreement with the exact solutions. When solving a typical elastoplastic contact problem, the present analysis is roughly two times faster than the image inclusion approach and six times faster than the direct method. In addition, the present work demonstrates that a significant enhancement in the computational efficiency can be achieved through the introduction of parallel computation. [DOI: 10.1115/1.4023948]

Keywords: eigenstress, eigenstrain, half space, fast Fourier transform, parallel computing

1 Introduction

Inhomogeneous and inelastic strains occur during a loading process and affect the elastic fields of contacting elements. Murakawa [1] notes that the inelastic strains can be treated as eigenstrains and the regions with eigenstrains termed as inclusions whose elastic moduli are the same as the matrix region [2]. Moreover, the stress disturbance due to inhomogeneities can be equivalent to the eigenstress resulting from corresponding inclusions with proper eigenstrains, known as the equivalent inclusion method (EIM) [2]. Understanding the elastic fields caused by inclusions provides an effective and convenient approach to solving complex contact problems that require a large amount of computation using conventional methods.

The elastic fields caused by the inclusions in a half space were investigated by a number of researchers involving several types of inclusions, such as spherical thermal inclusions [3], ellipsoidal thermal inclusions [4], cuboidal inclusions [5], and periodic distribution of dislocations [6,7]. For cases involving inclusions of more complicated geometry or structure, an analytical solution is generally difficult to obtain, and therefore a numerical solution is sought. A cuboidal inclusion solution is fundamental to conducting numerical computation since an arbitrarily shaped inclusion can be divided into several small cuboids, and the elastic fields calculated by superposing the contributions from each individual cuboidal inclusion. Chiu [5] employed the method of images to obtain closed form solutions to stresses and surface normal displacements. In his method, the relation between the eigenstress and eigenstrain is expressed as a recursive formula, without explicit demonstration of the convolution/correlation properties in the depth direction. Based on Chiu’s method, Jacq et al. [8] used a two-dimensional fast Fourier transform (2D-FFT) to calculate the residual stresses caused by the plastic strains, where the plastic zone was uniformly discretized into cuboidal inclusions. The stresses in their method were calculated layer by layer through the depth direction. This method has been widely used to analyze several elastoplastic contact problems [9–13]. Liu and Wang [14] demonstrated the explicit correlation properties in the depth direction with respect to the image inclusion; therefore, the discrete correlation using fast Fourier transform (DC-FFT) can be employed to solve for the stresses related to the image inclusion, while the stress arising from its own part, i.e., the inclusion in a full space, can be obtained by discrete convolution, using fast Fourier transform (DC-FFT). According to the convolution/correlation properties in the depth direction, Zhou et al. [15] used two three-dimensional fast Fourier transform (3D-FFT) procedures to solve the stresses due to the cuboidal inclusions and its image part in a full space, and one 2D-FFT procedure to solve the stress caused by surface normal traction. By using 3D-FFT, the method proposed by Zhou et al. [16] is significantly faster than the layer-by-layer 2D-FFT of Jacq et al. [8]; however the computational domain required appropriate extension to reduce the numerical error caused by normal traction cancellation.

Recently, Liu et al. [17] employing Galerkin vectors derived analytical solutions to the elastic fields caused by eigenstrains in a half space. Explicit closed-form results were obtained for the influence coefficients relating eigenstrains to displacements or stresses.
By using the influence coefficients, a numerical implementation based on 3D-FFT could be employed to speed up the computation, and the elastic field caused by arbitrarily shaped inclusions calculated. This method has been successfully used to analyze the contact involving joined quarter spaces [18]. The analytical solutions of influence coefficients can bypass the additional complexity of surface domain truncation but requires more computational time than the Zhou et al. [16] since four 3D-FFT schemes must be used.

Considering the rapid development of multicore processors in recent years, parallel computing has been increasingly used in solving eigenstrains and related problems, such as dislocation dynamics [19], crack propagation [20], and composite materials [21]. To employ parallel computing, a large problem is divided into smaller independent components, where each processor executes its own part of the algorithm simultaneously, leading to a significant increase in computational speed; and the theoretical maximum acceleration can be predicted by Amdahl’s law [22]. In the present problem, calculating influence coefficients and performing the FFT consume most of the numerical implementation time. Since influence coefficients are matrix components independent of each other, parallel computing can be employed to accelerate the computation speed. Moreover, the parallel version of FFT is now well developed and is adopted in this work.

The purpose of this paper is to develop a simpler and more rapid method to calculate the stresses caused by the presence of arbitrarily shaped inclusions in contact problems. Since the new method is only based on one full space solution, it is significantly faster and more efficient than previous methods. In addition, parallel computing is employed in the new solver to accelerate computation speed.

2 Theory of Inclusions in Materials Under Contact

2.1 Theoretical Basis of the New Method. Contact problems can be analyzed in a half space, as shown in Fig. 1. Here, the stresses are the sums of those of a full space solution and those caused by tractions required to create the free surface. The full space solutions can be obtained by using Chiu’s method [23] or Liu et al. [17].

2.1.1 Chiu’s Full Space Solution. The gradient of displacement components at a target point \( \mathbf{x} \) in a half space caused by a cuboidal inclusion of uniform eigenstrains \( \varepsilon^a_{ij} \) can be expressed as [23]

\[
2\mu\varepsilon^a_{ij}(\mathbf{x}) = \frac{(8\pi^3)}{1 - \nu} \sum_{n=1}^{k} (-1)^n \left[ \frac{1 - 2\nu}{1 - \nu} \varepsilon^a_{ik} : D_{ijmn}(\varepsilon_n) + 4\mu\varepsilon^a_{ik} : D_{ijmn}(\varepsilon_n) - \frac{2\mu}{1 - \nu} \varepsilon^a_{ik} : D_{ijmn}(\varepsilon_n) \right]
\]  

(1)

where \( \lambda \) and \( \mu \) are the Lamé constants, \( \nu \) is Poisson’s ratio, and \( \varepsilon_n \) are the vectors linking the eight corners of the cuboidal inclusion to the target point, the comma indicates partial differentiation, and \( D \) is as follows [23]:

\[
D_{ijmn}(\varepsilon_n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\exp(i\xi \cdot \varepsilon_n)}{(\xi_1^2 + \xi_2^2 + \xi_3^2)^{1/2}} d\xi_1 d\xi_2 d\xi_3
\]

(2)

Once the displacement gradients are calculated from Eq. (1), the eigenstress can be obtained from Hooke’s law,

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \frac{\mathbf{R}^f_{ij}(\mathbf{x}')}dx', \quad \mathbf{x} \notin \Omega
\]

(3a)

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \mathbf{R}^f_{ij}(\mathbf{x}'')dx'' - 2\mu\varepsilon^a_{ij} - \frac{2\mu}{1-\nu}\varepsilon^a_{ik}\delta_{ij}, \quad \mathbf{x} \in \Omega
\]

(3b)

where the cuboidal inclusion is in domain \( \Omega \).

2.1.2 Full Space Solution of Liu et al. The full space solution can be expressed as follows [17]:

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \mathbf{R}^f_{ij}(\mathbf{x}')dx', \quad \mathbf{x} \notin \Omega
\]

(4a)

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \mathbf{R}^f_{ij}(\mathbf{x}'')dx'' - 2\mu\varepsilon^a_{ij} - \frac{2\mu}{1-\nu}\varepsilon^a_{ik}\delta_{ij}, \quad \mathbf{x} \in \Omega
\]

(4b)

where \( \mathbf{x}' = [x_1', x_2', x_3', 2m, 2n, 2p]^T \), the semicolons are used to separate members in the vector; here the vector \( \mathbf{R}^f \) is defined as

\[
\mathbf{R}^f = \begin{bmatrix} \mathbf{R}^f_{11} & \mathbf{R}^f_{12} & \mathbf{R}^f_{13} & \mathbf{R}^f_{14} & \mathbf{R}^f_{15} & \mathbf{R}^f_{16} \\
\end{bmatrix}
\]

(5)

where the potential \( \mathbf{R}^f \) is introduced as follows:

\[
\mathbf{R}^f = |\mathbf{x} - \mathbf{x}'|
\]

(6)

Equation (4) is the general equation characterizing inclusions of arbitrary shape. When the domain \( \Omega \) is a cuboid containing uniform eigenstrains, integration can be performed and explicit relations between the eigenstresses and eigenstrains obtained.

Here Eq. (4) is used to obtain the full space solutions. The closed-form cuboidal solutions can be found in Liu et al. [17]. The stresses due to an arbitrarily shaped inclusion can be written as a summation of its cuboidal contributions, i.e.,

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \mathbf{R}^f_{ij}(\mathbf{x}')dx', \quad \mathbf{x} \notin \Omega
\]

\[
\sigma^a_{ij}(\mathbf{x}) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} \mathbf{R}^f_{ij}(\mathbf{x}'')dx'' - 2\mu\varepsilon^a_{ij} - \frac{2\mu}{1-\nu}\varepsilon^a_{ik}\delta_{ij}, \quad \mathbf{x} \in \Omega
\]

Fig. 1 New method: the elastic fields caused by an inclusion in a half space are the sum of results due to the inclusion and the surface tractions to create a surface free from their opposite tractions
where the asterisk symbolizes convolution; $T$ is the influence coefficient matrix, whose explicit expression is listed in Appendix A. It is worth mentioning that for a cuboidal inclusion with uniform eigenstrains inside, the internal stresses in $\Omega$ and external stresses outside $\Omega$ can be expressed by the same equation, analogous to the results of the two-dimensional problem [24,25].

After the full space solution is obtained, the half space problem is considered. Figure 2 shows the computational region, which is meshed into cuboidal elements of the same size, where the stresses at each element center and the surface tractions at the cuboidal surface center are determined. In addition, the surface tractions for each rectangular patch are treated as constant. The stress in a half space can be expressed as the contributions of each cuboidal inclusion as follows:

$$
\sigma_j(x, \beta, \gamma) = \frac{-\mu}{4\pi(1-\nu)} \sum_{\beta_0, \gamma_0=0}^{L-1} \sum_{\zeta=0}^{N-1} T_{ijkl}^{(0)}(x-\zeta, \beta-\eta, \gamma-\vartheta) \epsilon_{ij}^{(0)}(\zeta, \eta, \vartheta)
$$

(0 \leq x \leq M - 1, 0 \leq \beta \leq N - 1, 0 \leq \gamma \leq L - 1)

where $(\alpha, \beta, \gamma)$ or $(\zeta, \eta, \vartheta)$ is the central point of the cuboid. Equation (7), written in a matrix form, is as follows:

$$
\begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{33} \\
\sigma_{12}
\end{bmatrix}
= -\frac{\mu}{4\pi(1-\nu)}
\begin{bmatrix}
T_{1111}^{(0)} & T_{1112}^{(0)} & T_{1122}^{(0)} & T_{1222}^{(0)} \\
T_{2211}^{(0)} & T_{2212}^{(0)} & T_{2222}^{(0)} & T_{2232}^{(0)} \\
T_{3311}^{(0)} & T_{3312}^{(0)} & T_{3322}^{(0)} & T_{3332}^{(0)} \\
T_{1311}^{(0)} & T_{1312}^{(0)} & T_{1322}^{(0)} & T_{1332}^{(0)}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} \\
\epsilon_{22} \\
\epsilon_{33} \\
2\epsilon_{12}
\end{bmatrix}
$$

(8)

The first term of Eq. (9) shows three-dimensional discrete convolutions of influence coefficients and eigenstrains, and the others are two-dimensional discrete convolutions of influence coefficients and surface tractions. The discrete convolutions can be converted to circular discrete convolutions and the FFT can be employed to implement the computation. For the three-dimensional discrete convolution, when the size of the target domain is $M \times N \times L$, the extended computational domain must be $2M \times 2N \times 2L$ to avoid any FFT error. The calculation of influence coefficients is performed on the extended domain and implemented by the technique.
of wrap-around order while the eigenstrains in the extended eigenstrain need only zero padding [26]. The stresses can be obtained by taking an FFT of each sequence, i.e., influence coefficients and eigenstrains, multiplying point wise, and then performing an inverse FFT. Once completed, the stresses in the extended domain are discarded. Thus, the time to perform both FFT and inverse FFT in the first term of Eq. (9) is O\((864MNL  \ln 8MNL)\) and that for remaining terms is O\((216MNL \ln 4MNL)\).

Tractions at a surface point \((x, \beta, \gamma)\), shown in Fig. 2, can be obtained by

\[
\tau_{ij}(x, \beta, \gamma) = \frac{\mu}{4\pi(1-\nu)} \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{ijkl}(x-z, \beta - \eta, -\delta) e_{ijl}(\xi, \eta, \vartheta)
\]

(11a)

\[
\tau_{2i}(x, \beta, \gamma) = \frac{\mu}{4\pi(1-\nu)} \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{2ijk}(x-z, \beta - \eta, -\delta) e_{2ijl}(\xi, \eta, \vartheta)
\]

(11b)

\[
\tau_{3i}(x, \beta, \gamma) = \frac{\mu}{4\pi(1-\nu)} \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{3ijk}(x-z, \beta - \eta, -\delta) e_{3ijl}(\xi, \eta, \vartheta)
\]

(11c)

The DC–FFT in the x and y directions can be employed to calculate the shear tractions; computing Eq. (11) requires 3 \(\times\) \(L\) \(\times\) 6 \(\times\) 3 times 2D-FFT and inverse 2D-FFT. Thus the total time for performing FFT and inverse FFT in the computation for shear tractions is O\((216MNL \ln 4MNL)\). After obtaining the shear tractions, the stresses caused by shear tractions can be calculated by using the second, third, and fourth term of Eq. (9).

The time burden for performing the total FFT and inverse FFT in calculating stresses in a half space are listed in Table 1. The calculation of influence coefficients also requires considerable computing time. Because \(T_{ijkl}\) is a symmetric matrix; only 21 terms of \(T_{ijkl}\) need to be calculated in the new method.

### 2.2 Direct Method

Recently, Liu et al. [17] derived the complete set of elastic field solutions caused by an eigenstrain and further obtained the influence coefficients to relate eigenstrain to displacement or stresses. Eigenstress \(\sigma_{ij}^e\) due to eigenstrain in a half space can be expressed as [17]

\[
\sigma_{ij}^e(x) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} (\Theta_{ij} - \overline{R}_{ij} - \overline{R}_{ij} - 2\pi \overline{R}_{ij} + 2\pi \overline{R}_{ij}) [\varepsilon^e] dx', \quad x \notin \Omega
\]

(12a)

\[
\sigma_{ij}^e(x) = \frac{-\mu}{4\pi(1-\nu)} \int_{\Omega} (\Theta_{ij} - \overline{R}_{ij} - \overline{R}_{ij} - 2\pi \overline{R}_{ij} + 2\pi \overline{R}_{ij}) [\varepsilon^e] dx' - 2\mu \varepsilon_{ij} - \frac{2\mu}{1-\nu} \varepsilon_{ij} \delta_{ij}, \quad x \in \Omega
\]

(12b)

where vectors \(\Theta\), \(\overline{R}\), and \(\overline{R}\) are the combination of potentials. Similarly, an arbitrarily shaped inclusion can be divided into several small cuboidal inclusions with uniform eigenstrains inside of each one, where the total stresses are the superposition of the solutions caused by each cuboidal inclusion.

### 2.3 Image Inclusion Approach

The half space problem, shown in Fig. 3, can be solved by using the image method [5]. The cuboidal inclusion has uniform eigenstrains and its center is located at \((0, 0, z')\). Suppose the cuboidal inclusion has an image domain with its center located at \((0, 0, -z')\), and the eigenstrain components \(e_{ijkl}^0\), \(e_{ijkl}^1\), \(e_{ijkl}^2\), and \(e_{ijkl}^3\) in the two domains are equal while components \(e_{ijkl}^0\), \(e_{ijkl}^2\), and \(e_{ijkl}^3\) are equal but opposite. Then, the plane of symmetry is free from shear tractions, and the normal stress in this plane due to the two domains is double the stress caused by a single domain. The stresses in a half space can be obtained by adding the stresses caused by two cuboidal inclusions in a full space and the stresses caused by surface normal traction, which can make the surface free of normal traction.

An arbitrarily shaped inclusion is thus divided into several small cuboidal inclusions of the same size. The total stress caused by arbitrary inclusions can be calculated from the equation below [16].

\[
\sigma_{ij}^a(x, \beta, \gamma) = \frac{-\mu}{4\pi(1-\nu)} \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{ijkl}(x-z, \beta - \eta, -\delta) e_{ijl}(\xi, \eta, \vartheta)
\]

(11a)

\[
\times \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{ijkl}(x-z, \beta - \eta, -\delta) e_{ijl}(\xi, \eta, \vartheta)
\]

(11b)

\[
+ z \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{ijkl}(x-z, \beta - \eta, -\delta) e_{ijl}(\xi, \eta, \vartheta)
\]

(11c)

\[
+ 2 \sum_{l=1}^{L-1} \sum_{l=1}^{N-1} \sum_{l=1}^{M-1} T_{ijkl}(x-z, \beta - \eta, -\delta) e_{ijl}(\xi, \eta, \vartheta)
\]

(11d)

\[
(0 \leq x \leq M-1, 0 \leq \beta \leq N-1, 0 \leq \gamma \leq L-1)
\]

(13)

Here \(T_{ijkl}(0), T_{ijkl}(1), T_{ijkl}(2),\) and \(T_{ijkl}(3)\) are the influence coefficients relating eigenstrains to eigenstresses. More details on the influence coefficients \(T_{ijkl}(0), T_{ijkl}(1), T_{ijkl}(2),\) and \(T_{ijkl}(3)\) are available in Liu et al. [17], Appendix (A2). The first term of Eq. (13) includes 3D convolutions and the others include 3D convolution-correlation combinations. The convolution-correlation combination terms can be calculated by combining discrete convolution-discrete correlation and FFT (DC–DCTR–FFT) (DC–FFT in the x and y directions and DCTR–FFT in z direction). Likewise, 3D-FFT is employed to speed up the computation. Since the stress calculation needs 36 \(\times\) 4 \(\times\) 3 times 3D-FFT and inverse 3D-FFT, the total time for performing 3D-FFT is O\((3456MNL \ln 8MNL)\) (see Table 1).
where \( \gamma^{(0)}_{ijkl} \) are the influence coefficients relating eigenstrain to eigensplit discussed above; \( C_{ijkl}^{0} \) are the influence coefficients relating surface normal traction \( \tau_{zz} \) to stresses, and the explicit form of \( C_{ijkl}^{0} \) can be found in Appendix B. The image eigenstrain is \( \epsilon_{ijkl}^{a} \), where \( \epsilon_{11}^{a} = \epsilon_{12}^{a} = \epsilon_{13}^{a} = \epsilon_{22}^{a} = \epsilon_{23}^{a} = \epsilon_{33}^{a} = \epsilon_{12}^{b} = \epsilon_{13}^{b} = -\epsilon_{23}^{b} \), and \( \epsilon_{13}^{b} = -\epsilon_{23}^{b} \). The surface normal traction \( \tau_{zz} \) can be obtained by using Eq. (11a). In Eq. (14), the first term is a 3D convolution group; the second term is a 3D convolution-correlation combination; and the third term is a two-dimensional convolution set.

The 3D-FFT can be employed to accelerate the calculation of first and second terms of Eq. (14) \cite{15,16}, and the 2D-FFT for the third term \cite{16,27}. For the first and second terms in Eq. (14), the time burden to perform a set 3D-FFT or inverse 3D-FFT is \( O(8MNL \ln 8MNL) \). The total time consumption is listed in Table 1. In order to capture the surface normal traction \( \tau_{zz} \), accurately, the computational domain must be extended appropriately and covered by fine grids, which are discussed in Sec. 3 below.

3 Numerical Results and Discussion

3.1 Full Space Solutions. First, the stresses caused by a cuboidal inclusion in a full space are considered. The stresses in this space are influenced by the distance between the target surface and the inclusion. Figure 4 shows the stresses along the \( x \) axis caused by the cuboidal inclusion at different positions. The three sides of the cuboid are each of length \( 2a \) in the \( x \), \( y \), and \( z \) directions. The cuboid center is located at the \( z \) axis, and the \( xy \) plane is selected as the target plane, which is parallel to one of the cuboidal faces. Assuming uniform eigenstrains, \( \{e\} = \{e_{zz}; e_{zy}; e_{zy}; 2e_{zz}; 2e_{zy}; 2e_{zy}\} = 10^{-3} \times \{1, 1, 1, 1, 1, 1\} \) inside and zero outside of the cuboidal inclusion. If the inclusion touches the target surface (Fig. 4(b)), the values of surface stresses \( \sigma_{zz}, \sigma_{zy}, \) and \( \sigma_{zy} \) are large and dramatic variations at the left and right interfaces can be observed. As the inclusion is further from the target surface, the values of surface stresses decrease (Figs. 4(b)-4(d)). If three tractions \( \tau_{zz} = -\sigma_{zz}, \tau_{zy} = -\sigma_{zy}, \) and \( \tau_{zy} = -\sigma_{zy} \) are imposed on the target plane, the target plane becomes traction-free, and the half space solution can be obtained. In the new method, the stresses caused by tractions \( \tau_{zz}, \tau_{zy}, \tau_{zy} \) are calculated by using the numerical approach. Theoretically, tractions have nonzero values on the entire target plane; however, as they decrease drastically when the distance is further from the inclusion, the computational domain must be extended to reduce truncation errors caused by shear tractions. On the other hand, the computational domain is meshed by many cuboidal elements with uniform eigenstrains inside; thus, the mesh should be fine enough to represent arbitrarily shaped inclusions.

3.2 Half Space Solutions

3.2.1 A Single Inclusion. In this section half space problems are considered, where the extreme cases, i.e., the inclusion touching the surface, are chosen. The inclusion size and the eigenstrains values inside are the same as those mentioned in Sec. 3.1. Six cases with different computational domains and grid sizes are listed in Table 2 and are used as benchmarks, where the analytical solution by Liu et al. \cite{17} is employed.

3.2.1.1 Effect of computational domain. Cases 1 to 3 are analyzed, where the mesh size is held constant in the three directions, i.e., \((\Delta x, \Delta y, \Delta z) = (0.222, 0.222, 0.2)\). Figure 5(a) shows the von Mises stress along the line passing through point \((0, 0.1a, 0)\) and parallel to the \( x \) axis, and Fig. 5(b) shows the von Mises stress along the \( z \) axis. As the computational domain is increased, the results become more accurate and the computational domain of \([-3a, 3a] \times [-3a, 3a] \times [0, 3a]\) in three directions is seen to be large enough to obtain a stable and accurate solution. Note that the computational domain is affected by the position of the inclusion. The extreme case occurs as the inclusion touches the surface, and the extended computational domain is larger in the extreme case than that for the inclusion inside the body.

3.2.1.2 Effects of grid density. A set of simulations on a fixed computational domain \([-3a, 3a] \times [-3a, 3a] \times [0, 3a]\) with varying grid sizes is performed to study the effect of grid size. Figure 6 shows the von Mises stress along the two target lines, i.e., the line passing through point \((0, 0.1a, 0)\) and parallel to the \( x \) axis and a line along the \( z \) axis. As the grid size increases, the results from the new method agree well with the analytical solutions. In case 6, the peaks of the stress are captured by using fine grids (Fig. 6(a)).

3.2.1.3 Comparison of different methods. The results obtained by the new method are compared to those from the method of Zhou et al. \cite{16}, where case 2 is taken as an example. Figure 7 shows the stresses \( \sigma_{zz}, \sigma_{zy}, \) and \( \sigma_{zy} \) along the two target lines, similar to Figs. 5 and 6. The main source of error in the method of Zhou et al. \cite{16} arises from the elimination of the surface normal traction; in the new approach, error is caused by elimination of both the surface normal and the shear tractions. Note that the normal traction in method of Zhou et al. \cite{16} is twice that of the normal traction in the new method. In most cases, the method based on Zhou et al. \cite{16} can achieve a better solution when the inclusions are near to the surface (Fig. 7(a)). When the
computational domain is extended sufficiently large with grids dense enough along the $x$ and $y$ directions (as discussed above), the numerical results by using the two methods are almost the same as those obtained from the analytical solution.

### 3.2.2 Multiple Inclusions

A multi-inclusion problem is further considered, shown in Fig. 8(a), a half space having two layered inclusions with the top one touching the surface. Similarly, each inclusion size and the eigenstrains values inside the inclusions are the same as those mentioned in Sec. 3.1. The distance between two adjacent inclusions are all assumed to be $a$. A fixed computational domain $[-6a, 6a] \times [-6a, 6a] \times [0, 6a]$ with $128 \times 128 \times 64$ grids is used for the numerical solutions. The contours of the dimensionless von Mises stress in the $xz$ plane are shown in Figs. 8(b)–8(d), where those in Figs. 8(b)–8(d) are the results from the new method, the methods of Zhou et al. [16] and Liu et al. [17], respectively. Multi-inclusion problem can be solved by using the three methods and the results are in good agreement with each other.

### 3.3 Elastoplastic Contact

The contact of an elastoplastic sphere with elastic perfectly plastic behavior with a rigid plane is shown in Fig. 9 and is analyzed to evaluate the accuracy and efficiency of different methods. The loading conditions and the material properties of the sphere are listed in Table 3. In the numerical simulation, the loading path is divided into 10 steps. Here the plastic strains are treated as eigenstrains and the residual stresses as eigenstresses. The residual stresses due to plastic strains are calculated with an elastoplastic contact solver. In this section, the new method is compared with the image inclusion approach of Zhou et al. [16] and the direct method Liu et al. [17]. More details for the numerical approaches to solve the elastoplastic contact can be found in the related works [8–14].

Figure 10 plots the dimensionless pressure along the $x$ axis and the equivalent plastic strain along the $z$ axis, and Fig. 11 shows the contours of the dimensionless residual von Mises stress $\sigma_{\text{res}}/\sigma_0$ in the $xz$ plane. More computational results comparing the three different methods are listed in Table 4. Excellent agreement

### Table 2 Cases of different computational domains and grid sizes

<table>
<thead>
<tr>
<th>Case</th>
<th>Computational domain</th>
<th>Cuboidal elements</th>
<th>Grid intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[-a,a] \times [-a,a] \times [0,3a]$</td>
<td>$9 \times 9 \times 15$</td>
<td>(0.222, 0.222, 0.2)</td>
</tr>
<tr>
<td>2</td>
<td>$[-3a,3a] \times [-3a,3a] \times [-5a,5a]$</td>
<td>$27 \times 27 \times 15$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$[-5a,5a] \times [-5a,5a] \times [0,3a]$</td>
<td>$45 \times 45 \times 15$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$[-3a,3a] \times [-3a,3a] \times [0,3a]$</td>
<td>$3 \times 3 \times 15$</td>
<td>(2, 2, 0.2)</td>
</tr>
<tr>
<td>5</td>
<td>$[-3a,3a] \times [-3a,3a] \times [0,3a]$</td>
<td>$9 \times 9 \times 15$</td>
<td>(0.667, 0.667, 0.2)</td>
</tr>
<tr>
<td>6</td>
<td>$[-3a,3a] \times [-3a,3a] \times [0,3a]$</td>
<td>$243 \times 243 \times 15$</td>
<td>(0.025, 0.025, 0.2)</td>
</tr>
</tbody>
</table>

3.3 Elastoplastic Contact. The contact of an elastoplastic sphere with elastic perfectly plastic behavior with a rigid plane is shown in Fig. 9 and is analyzed to evaluate the accuracy and efficiency of different methods. The loading conditions and the material properties of the sphere are listed in Table 3. In the numerical simulation, the loading path is divided into 10 steps. Here the plastic strains are treated as eigenstrains and the residual stresses as eigenstresses. The residual stresses due to plastic strains are calculated with an elastoplastic contact solver. In this section, the new method is compared with the image inclusion approach of Zhou et al. [16] and the direct method Liu et al. [17]. More details for the numerical approaches to solve the elastoplastic contact can be found in the related works [8–14].

Figure 10 plots the dimensionless pressure along the $x$ axis and the equivalent plastic strain along the $z$ axis, and Fig. 11 shows the contours of the dimensionless residual von Mises stress $\sigma_{\text{res}}/\sigma_0$ in the $xz$ plane. More computational results comparing the three different methods are listed in Table 4. Excellent agreement
from the three methods is observed. Figure 12 shows the computational time consumed by different methods using a computer of a 64-bit Windows operating system with Intel® Core™ i7-920 Processor @ 2.67 GHz. As shown in Fig. 12, the computational speed of the new method is about twice faster than that of the image inclusion approach [16] and six times faster than the direct method [17].

4 Parallel Computing Scheme

The computational efficiency can be significantly increased by using a parallel computing technique. There are two types of parallel programming models, i.e., open multiprocessing (OpenMP) and message passing interface (MPI). The former refers to computing on shared-memory computers with multiprocessing, while the latter is for distributed memory computers with multiple independent processors, and the data on each processor are inaccessible to other processors except by explicit message passing.

The parallel strategy for calculating eigenstress is shown in Fig. 13. The influence coefficients matrix \( T \) is a symmetric matrix for the full space problem. \( T^{(0)} \) is a block matrix, where every component of \( T^{(0)} \) is independent of each other, and can be calculated simultaneously through parallel computing. After obtaining the influence coefficients, the discrete convolution of the eigenstrains and influence coefficients can employ the FFT algorithms via the circular convolution theorem. The eigenstresses can be obtained after performing \( 36 \times 3 \) times FFT or inverse FFT. In the present study, a free software package, i.e., FFTW [28], is employed for performing the FFT. FFTW is a C subroutine library for computing the FFT in one or more dimensions of an arbitrary input size, and for both real and complex data (http://www.fftw.org). Parallel computing based on OpenMP is used when performing the FFT. The source code in this paper is written in standard Fortran 90 and compiled using Intel® Visual Fortran Studio XE 2011 for Windows. The Intel® Core™ i7-920 Processor @ 2.67 GHz with four cores is used for speed comparison. In each of the examples below, all data types are in double precision.
The efficiency of parallel FFTW is examined first. Figure 14 shows the execution time for different 3D transform size with the double-precision complex data type. To get more accurate results, each execution time is obtained by repeating 3D-FFT transforms 10 times. The results show that when the processor number is increased, the execution time decreases dramatically. With the same array size, the computational efficiency by using four processors is almost three times faster than a single processor. The execution time for calculating the eigenstresses are shown in Fig. 15 for different grid sizes. Furthermore, the elastoplastic case described in the Sec. 3 is analyzed, and the execution time is shown in Fig. 16. For a finer grid, i.e., $2^5 \times 2^5 \times 2^5$, the time is 3295 s by using four processors and 10,043 s by one processor.

Note that the calculation of eigenstrains, eigenstresses, or the residual stresses, requires roughly 95% of the computation time during the whole contact analysis. Several other portions of elastoplastic code, such as the calculation of influence coefficients...
relating pressure displacement and pressure stress, 2D-FFT involving stress and displacement are also performed similarly with the parallel implementation strategy. Parallel computing is well suited for the SAMs based on the influence coefficients and FFT.

It is well worth mentioning that many researchers have studied the full space solutions for polyhedral inclusions with uniform eigenstrains [29–31]. The half space solution for tetrahedral or other types of inclusions can be calculated with a numerical method from the full space solution, as shown in Fig. 1. In general tetrahedral meshes are better for the arbitrary inclusions; however, FFT can only be employed for the cuboidal meshes of equal size (Fig. 2).

<table>
<thead>
<tr>
<th>Table 4 Results for elastoplastic contact with different methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>New method</td>
</tr>
<tr>
<td>Based on Zhou et al. [16]</td>
</tr>
<tr>
<td>Based on Liu et al. [17]</td>
</tr>
</tbody>
</table>

Fig. 12 Computational time with different methods: (a) new method, (b) based on Zhou et al. [16], and (c) based on Liu et al. [17]

Fig. 13 Parallel strategy for calculating stress caused by eigenstrains in the full space

\[ \sigma_i = \frac{\sigma_{eq}}{4\pi(1-\nu)} \]
5 Conclusions

A simple but efficient method is proposed to solve the stresses caused by inclusions in a half space, based on full space solutions. The verification calculation shows the solutions based on the new method agree well with the corresponding analytical solutions from the direct method of Liu [17]. The new method is also applicable to the extreme cases, i.e., when the inclusion is touching the surface. Here the computational domain along x and y directions must be sufficiently extended, e.g., three times larger than that of the inclusion domain, and the grid fine enough to discretize the inclusions. The computational speed based on the new method is about twice as fast as the imaging method of Zhou et al. [16] and six times as fast as the direct method by Liu et al. [17]. Furthermore, parallel computing is introduced to perform the FFT to obtain the influence coefficients. When solving for the eigenstresses in a half space, the computational efficiency is almost three times faster by using four processors than by using a single processor.

Acknowledgment

Z. Wang would like to express sincere gratitude to the support from the National Science Foundation of China under Grant No. 51105391 and the Ph.D. Programs Foundation of Ministry of Education of China under Grant No. 20110191120009. The authors would like to acknowledge supports from Center for Surface Engineering and Tribology at Northwestern University, USA, and State Key Laboratory of Mechanical Transmission, Chongqing University, China. In addition, the authors like to thank Professor Nenzi Wang at Chang Gung University for inspiratory discussion.

Nomenclature

\[ a = \text{Hertzian contact radius, mm} \]
\[ c_a = \text{vectors linking the eight corners of the cuboidal inclusion to the target point} \]
\[ C_{ijkl}^{(\text{in})}, C_{ijkl}^{(0)}, C_{ijkl}^{(s)} = \text{influence coefficients relating surface tractions } \tau_{xu}, \tau_{yu}, \text{and } \tau_{zu} \text{ to stresses} \]
\[ E = \text{Young’s modulus, GPa} \]
\[ p = \text{pressure, MPa} \]
\[ p_h = \text{maximum Hertzian pressure, MPa} \]
\[ R = \text{radius of the sphere, mm} \]
\[ T_{ijkl}^{(0)}, T_{ijkl}^{(1)}, T_{ijkl}^{(2)}, T_{ijkl}^{(3)} = \text{influence coefficients relating eigenstrain to stress} \]
\[ u_i = \text{displacement, mm} \]
\[ W = \text{applied normal load, N} \]
\[ x, y, z = \text{space coordinates, mm} \]
\[ \delta_{ij} = \text{Kronecker delta} \]
\[ \Delta_x, \Delta_y, \Delta_z = \text{grid size in the } x, y, \text{and } z \text{ direction, mm} \]
\[ e_{ij}^{(s)} = \text{eigenstrain} \]
\[ \nu = \text{Poisson’s ratio} \]
\[ \sigma_{ij} = \text{stress, MPa} \]
\[ \sigma_{ij}^{(e)} = \text{eigenstress due to eigenstrain, MPa} \]
\[ \mu = \text{shear modulus, MPa} \]
\[ \tau_{xz}, \tau_{xz}, \tau_{xy} = \text{surface tractions, MPa} \]

Special Symbols

\[ * = \text{convolution} \]

Appendix A: Influence Coefficients Relating Eigenstresses to Eigenstresses in a Full Space

Consider a cuboidal inclusion with uniform eigenstresses \( e_{ij}^{(0)} \) in a full space. The center of the cuboid is located at the origin of the Cartesian system \((x, y, z)\), and the three sides of the cuboid have length \( \Delta_x, \Delta_y, \Delta_z \) in the \( x, y, \text{and } z \) directions, respectively. The eigenstress at the target point \( Q(x, \beta, \gamma) \) can be calculated by

\[
\sigma_{ij}^{(e)}(x, \beta, \gamma) = \frac{-\mu}{4\pi(1 - \nu)} T_{ijkl}^{(0)}(x, \beta, \gamma) e_{ij}^{(0)}
\]  

(A1)

where

\[
T_{ijkl}^{(0)}(x, \beta, \gamma) = -f_{ijkl}(x_1, y_1, z_1) + f_{ijkl}(x_2, y_1, z_1) - f_{ijkl}(x_2, y_2, z_1) + f_{ijkl}(x_1, y_2, z_1)
+ f_{ijkl}(x_1, y_2, z_2) - f_{ijkl}(x_2, y_1, z_2) + f_{ijkl}(x_2, y_2, z_2)
- f_{ijkl}(x_1, y_1, z_2) + f_{ijkl}(x_2, y_1, z_2)
\]  

(A2)
and

\[ f_{1111}(\xi_1, \xi_2, \xi_3) = \frac{\xi_1 \xi_3}{r(r + \xi_1)} + \frac{\xi_1 \xi_2}{r(r + \xi_2)} + 2 \left[ \tan^{-1} \left( \frac{\xi_1 \xi_3}{r \xi_1} \right) + \tan^{-1} \left( \frac{\xi_1 \xi_2}{r \xi_2} \right) \right] \]

\[ f_{1122}(\xi_1, \xi_2, \xi_3) = \frac{\xi_1 \xi_2}{r(r + \xi_1)} + 2 \nu \tan^{-1} \left( \frac{\xi_1 \xi_2}{r \xi_1} \right) \]

\[ f_{1133}(\xi_1, \xi_2, \xi_3) = \frac{\xi_1 \xi_3}{r(r + \xi_2)} + 2 \nu \tan^{-1} \left( \frac{\xi_1 \xi_3}{r \xi_2} \right) \]

\[ f_{1122}(\xi_1, \xi_2, \xi_3) = 2 \nu \log(r + \xi_1) - \frac{\xi_1}{r} \]

\[ f_{1113}(\xi_1, \xi_2, \xi_3) = \log(r + \xi_2) - \frac{\xi_2}{r} \]

\[ f_{1111}(\xi_1, \xi_2, \xi_3) = \log(r + \xi_3) - \frac{\xi_3}{r} \]

\[ f_{2222}(\xi_1, \xi_2, \xi_3) = \frac{\xi_2 \xi_3}{r(r + \xi_3)} + \frac{\xi_2 \xi_1}{r(r + \xi_1)} + 2 \left[ \tan^{-1} \left( \frac{\xi_2 \xi_3}{r \xi_3} \right) + \tan^{-1} \left( \frac{\xi_2 \xi_1}{r \xi_1} \right) \right] \]

\[ f_{2333}(\xi_1, \xi_2, \xi_3) = \frac{\xi_1 \xi_3}{r(r + \xi_2)} + \frac{\xi_2 \xi_3}{r(r + \xi_1)} + 2 \left[ \tan^{-1} \left( \frac{\xi_1 \xi_3}{r \xi_2} \right) + \tan^{-1} \left( \frac{\xi_2 \xi_3}{r \xi_1} \right) \right] \]

where

\[ r = \sqrt{\frac{\xi_1^2}{\xi_1} + \frac{\xi_2^2}{\xi_2} + \frac{\xi_3^2}{\xi_3}} \]

and

\[ \tan^{-1} \left( \frac{\xi_1 \xi_3}{r \xi_3} \right) \bigg|_{\xi_1 = 0} = \frac{\pi}{2} \text{sign}(\xi_1 \xi_2) \]

\[ \tan^{-1} \left( \frac{\xi_2 \xi_3}{r \xi_1} \right) \bigg|_{\xi_2 = 0} = \frac{\pi}{2} \text{sign}(\xi_2 \xi_3) \]
\[
g^{n, f}_{j, j'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ -\tan \left( \frac{\xi_1 \xi_2}{r_{j, j'}} \right) \frac{r_{j, j'}}{r(r + \xi_j)} + \frac{r_{j, j'}}{r(r + \xi_1)} \right] (B6)
\]

\[
g^{n, f}_{j, j'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \frac{\xi_3}{r(r + \xi_j)} \right] (B7)
\]

\[
g^{n, f}_{i, j'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \frac{\xi_3}{r(r + \xi_j)} \right] (B8)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \frac{2(\nu - 1) \log(r + \xi_3) - \frac{\xi_3}{r}}{r} \right] (B9)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \frac{2\log(r + \xi_3) + \xi_3(1 - 2\nu)}{r} \times \left[ \frac{\xi_2}{r(r + \xi_3)} + \frac{r_{i, j'}}{r(r + \xi_3)} - \frac{2\nu \xi_2^2}{r} \right] \right] (B10)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \frac{2\log(r + \xi_3) - \xi_3(1 - 2\nu)}{r} \times \left[ \frac{\xi_2}{r(r + \xi_3)} + \frac{r_{i, j'}}{r(r + \xi_3)} - \frac{2\nu \xi_2^2}{r} \right] \right] (B11)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ -\frac{\xi_3^2}{r(r + \xi_j)} \right] (B12)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ -\frac{\xi_3^2}{r(r + \xi_j)} \right] (B13)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ -\frac{\xi_3^2}{r(r + \xi_j)} \right] (B14)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B15)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B16)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B17)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B18)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B19)
\]

\[
g^{n, f}_{i, i'}(\xi_1, \xi_2, \xi_3) = \frac{1}{2\pi} \left[ \log(r + \xi_3) - \xi_3(1 - 2\nu) \frac{\xi_1}{r(r + \xi_3)} - \frac{2\nu \xi_1^2}{r} \right] (B20)
\]

where

\[
r = \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2} \quad \text{and} \quad \tan^{-1} \left( \frac{\xi_1 \xi_2}{r_{j, j'}} \right) \bigg|_{\xi_3=0} = \frac{\pi}{2} \text{sign}(\xi_1 \xi_2) (B22)
\]

References