Title: Eshelby's problem

Subtitle: Analytical solution and FEM verification

Author: Jakub Mikula

Date: December 25, 2024

1 Introduction

The purpose of this post is to expound upon the fundamental tensors that arise within the Eshelby's problem, offering their concise closed forms. While the tedious derivations are omitted here, I supplement this with finite element simulations to validate the equations. Included are Matlab/Octave codes designed to compute the Eshelby's tensors for arbitrary ellipsoids and anisotropic materials. These codes may prove particularly valuable in coding self-consistent polycrystal homogenization approaches.
Eshelby's problem centers around an ellipsoidal inclusion embedded within an infinitely large homogeneous matrix. When the inclusion alters its dimensions or form through the imposition of eigen strains (such as thermal or phase-induced strains), it prompts stress formation both within the inclusion and its surroundings. Remarkably, Eshelby demonstrated that stress (and strain) within the ellipsoidal inclusion remains uniform.
The correlation between the applied eigen strain ε^0_{ij} within the ellipsoidal domain and the ensuing strain ε_{ij} generated within this domain is described by the equation:

$$\varepsilon_{ij} = S_{ijkl}\varepsilon^0_{kl} \tag{1}$$

where S_{ijkl} is the fourth order Eshelby's tensor. Beginning with the most general case of the Eshelby's tensor in the following section, I then proceed with simpler cases developed under certain specific assumptions.

1.1 Fully anisotropic material

Let's begin with the implicit equation of an ellipsoid:

$$\left(\frac{x_1}{a_1}\right)^2 + \left(\frac{x_2}{a_2}\right)^2 + \left(\frac{x_3}{a_3}\right)^2 = 1 \tag{2}$$

where x_i are the coordinates along the principal axes and a_i are the lengths of the semi-axes.
In the case of a fully anisotropic material, the Eshelby's tensor cannot be derived in a closed form but can be expressed in terms of elliptical integrals that can be solved numerically. This most general form is expanded in the text below.
Let us first define the Hill polarization tensor P_{ijkl} in the form presented by [1], [2].

$$P_{ijkl} = \frac{1}{4\pi}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} M_{ijkl}(\theta,\phi)\sin\theta d\theta d\phi \tag{3}$$

where

$$M_{ijkl} = \frac{1}{4}\left(A^{-1}_{jk}\xi_i\xi_l + A^{-1}_{ik}\xi_j\xi_l + A^{-1}_{jl}\xi_i\xi_k + A^{-1}_{il}\xi_j\xi_k\right) \tag{4}$$

Note that in the previous equation, A^{-1}_{jk} indicate the elements of the inverse matrix.
Let us then define a vector $\vec{\xi}$ given by:

$$\begin{aligned}\xi_1 &= \frac{\sin\theta\cos\phi}{a_1}\\ \xi_2 &= \frac{\sin\theta\sin\phi}{a_2}\\ \xi_3 &= \frac{\cos\phi}{a_3}\end{aligned} \tag{5}$$

and a tensor:

$$A_{ik} = C_{ijkl}\xi_j\xi_l \tag{6}$$

The Eshelby's tensor can then be obtained by the multiplication of the Hill polarization tensor P_{ijkl} and the tensor of elastic constants C_{ijkl}:

$$S_{ijkl} = P_{ijmn}C_{mnkl} \tag{7}$$

Having established the Eshelby's tensor, it is now possible to calculate the strain $\boldsymbol{\varepsilon}$ within the ellipsoid given its eigen strain $\boldsymbol{\varepsilon}^0$. Notice however, that for a fully anisotropic case, numerical integration needs to be employed to evaluate the previous equations. One possible implementation is discussed below.

1.2 Numerical Integration - Gauss-Legendre Quadrature

The integration of the previous equations can be approximated by the Gauss integration scheme:

$$\frac{1}{4\pi}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} M_{ijkl}(\theta,\phi)\sin\theta d\theta d\phi \approx \frac{1}{4\pi}\sum_i\sum_j M_{ijkl}(\theta_i,\phi_i)\sin(\theta_i)\, w_i w_j \det \boldsymbol{J} \tag{8}$$

where $\theta = \frac{\pi}{2}(\xi+1)$ and $\phi = \pi(\eta+1)$ and the Jacobian of the transformation is

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial\theta}{\partial\xi} & \frac{\partial\theta}{\partial\eta} \\ \frac{\partial\phi}{\partial\xi} & \frac{\partial\phi}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} & 0 \\ 0 & \pi \end{bmatrix} \tag{9}$$

therefore $\det \boldsymbol{J} = \frac{\pi^2}{2}$.
In the previous expressions, w_i are the Gauss weights and ξ and η are the Gauss node coordinates.
A Matlab code evaluating the Eshelby's tensor for arbitrary aspect ratio and anisotropy is given below:

Listing 1: Matlab example

```
% PURPOSE:
% For a given eigen strain compute the strain within the ellipsoid using the Eshelby's equations.

clear
clc

% Compute the Hill polarization tensor:
% Select the number of Gauss integration points:
ngp =10;

% Define the shape of the ellipsoid
a_1 = 2;
a_2 = 1;
a_3 = 3;

% Define the elastic constants:
% Isotropic elasticity:
E = 210.0e9;
v = 0.3;

c_11 = E*(1-v)/(1+v)/(1-2*v);
c_12 = E*v/(1+v)/(1-2*v);
c_13 = E*v/(1+v)/(1-2*v);
c_14 = 0;
c_15 = 0;
c_16 = 0;
c_22 = E*(1-v)/(1+v)/(1-2*v);
c_23 = E*v/(1+v)/(1-2*v);
c_24 = 0;
c_25 = 0;
c_26 = 0;
c_33 = E*(1-v)/(1+v)/(1-2*v);
```

```
c_34 = 0;
c_35 = 0;
c_36 = 0;
c_44 = E/2/(1+v);
c_45 = 0;
c_46 = 0;
c_55 = E/2/(1+v);
c_56 = 0;
c_66 = E/2/(1+v);

% Define eigen strain
e_0 = [0.023522, 0.076558, 0.019838, 0.068208, 0.059107, 0.031905];

C = [
    c_11/2 c_12   c_13   c_14   c_15   c_16
    0      c_22/2 c_23   c_24   c_25   c_26
    0      0      c_33/2 c_34   c_35   c_36
    0      0      0      c_44/2 c_45   c_46
    0      0      0      0      c_55/2 c_56
    0      0      0      0      0      c_66/2
];
C = C+C'

% Fully anisotropic material:
%C = [
%        1.9627       0.71094        1.2973        2.1394         1.908        1.1079
%       0.71094       0.88304       0.57145         1.129       0.85012       0.76032
%        1.2973       0.57145        1.7383        1.7461        1.5757       0.97334
%        2.1394         1.129        1.7461        2.8236        2.6732        1.5282
%         1.908       0.85012        1.5757        2.6732        2.9695        1.4951
%        1.1079       0.76032       0.97334        1.5282        1.4951        1.1461
%         ];

% Convert from Voigt notation to tensor notation:
for i=1:3
for j=1:3
for k=1:3
for l=1:3
    c(1, 1, 1, 1) = C(1,1);
    c(1, 1, 2, 2) = C(1,2);
    c(1, 1, 3, 3) = C(1,3);
    c(1, 1, 2, 3) = C(1,4);
    c(1, 1, 1, 3) = C(1,5);
    c(1, 1, 1, 2) = C(1,6);

    c(2, 2, 1, 1) = C(2,1);
    c(2, 2, 2, 2) = C(2,2);
    c(2, 2, 3, 3) = C(2,3);
    c(2, 2, 2, 3) = C(2,4);
    c(2, 2, 1, 3) = C(2,5);
    c(2, 2, 1, 2) = C(2,6);

    c(3, 3, 1, 1) = C(3,1);
    c(3, 3, 2, 2) = C(3,2);
    c(3, 3, 3, 3) = C(3,3);
    c(3, 3, 2, 3) = C(3,4);
    c(3, 3, 1, 3) = C(3,5);
    c(3, 3, 1, 2) = C(3,6);

    c(2, 3, 1, 1) = C(4,1);
    c(2, 3, 2, 2) = C(4,2);
    c(2, 3, 3, 3) = C(4,3);
    c(2, 3, 2, 3) = C(4,4);
```

```
    c(2, 3, 1, 3) = C(4,5);
    c(2, 3, 1, 2) = C(4,6);

    c(1, 3, 1, 1) = C(5,1);
    c(1, 3, 2, 2) = C(5,2);
    c(1, 3, 3, 3) = C(5,3);
    c(1, 3, 2, 3) = C(5,4);
    c(1, 3, 1, 3) = C(5,5);
    c(1, 3, 1, 2) = C(5,6);

    c(1, 2, 1, 1) = C(6,1);
    c(1, 2, 2, 2) = C(6,2);
    c(1, 2, 3, 3) = C(6,3);
    c(1, 2, 2, 3) = C(6,4);
    c(1, 2, 1, 3) = C(6,5);
    c(1, 2, 1, 2) = C(6,6);
end
end
end
end

% Apply minor symmetries:
for i=1:3
for j=1:3
for k=1:3
for l=1:3
    c(j, i, k, l) = c(i, j, k, l);
    c(i, j, l, k) = c(i, j, k, l);
    c(j, i, l, k) = c(i, j, k, l);
end
end
end
end



switch ngp
  case 1
    x = [0];
    w = [2];
  case 2
    x = [0.5773502692, -0.5773502692];
    w = [1,1];
  case 3
    x = [0.7745966692, -0.7745966692, 0.0];
    w = [0.5555555556, 0.5555555556, 0.8888888889];
  case 4
    x = [0.8611363116, -0.8611363116, 0.3399810436, -0.3399810436];
    w = [0.3478548451, 0.3478548451, 0.6521451549, 0.6521451549];
  case 5
    x = [0.9061798459, -0.9061798459, 0.5384693101, -0.5384693101, 0.0];
    w = [0.2369268851, 0.2369268851, 0.4786286705, 0.4786286705, 0.5688888889];
  case 6
    x = [0.9324695142, -0.9324695142, 0.6612093865, -0.6612093865,
        0.2386191861, -0.2386191861];
    w = [0.1713244924, 0.1713244924, 0.3607615730, 0.3607615730, 0.4679139346,
        0.4679139346];
  case 7
    x = [ 0.9491079123, -0.9491079123, 0.7415311856, -0.7415311856,
        0.4058451514, -0.4058451514, 0.0];
    w = [0.1294849662, 0.1294849662, 0.2797053915, 0.2797053915, 0.3818300505,
        0.3818300505, 0.4179591837];
  case 8
```

```
    x = [0.9602898565, -0.9602898565, 0.7966664774, -0.7966664774,
        0.5255324099, -0.5255324099, 0.1834346425, -0.1834346425];
    w = [0.1012285363, 0.1012285363, 0.2223810345, 0.2223810345, 0.3137066459,
        0.3137066459, 0.3626837834, 0.3626837834];
  case 9
    x = [0.9681602395, -0.9681602395, 0.8360311073, -0.8360311073,
        0.6133714327, -0.6133714327, 0.3242534234, -0.3242534234, 0.0];
    w = [0.0812743883, 0.0812743883, 0.1806481607, 0.1806481607, 0.2606106964,
        0.2606106964, 0.3123470770, 0.3123470770, 0.3302393550];
  case 10
    x = [0.9739065285, -0.9739065285, 0.8650633667, -0.8650633667,
        0.6794095683, -0.6794095683, 0.4333953941, -0.4333953941, 0.1488743390,
        -0.1488743390];
    w = [0.0666713443, 0.0666713443, 0.1494513492, 0.1494513492, 0.2190863625,
        0.2190863625, 0.2692667193, 0.2692667193, 0.2955242247, 0.2955242247];
  otherwise
    error('Invalid number of Gauss integration points. Process terminated!')
end

w

% Tensor M:
M = zeros(3,3,3,3);


for ii=1:ngp
for jj=1:ngp
  phi = pi*(x(ii) + 1);
  theta = pi/2*(x(jj) + 1);

  xi(1) = sin(theta)*cos(phi)/a_1;
  xi(2) = sin(theta)*sin(phi)/a_2;
  xi(3) = cos(theta)/a_3;


  A = zeros(3,3);
  for i=1:3
  for j=1:3
  for k=1:3
  for l=1:3
    A(i,k) = A(i,k) + c(i,j,k,l)*xi(j)*xi(l);
  end
  end
  end
  end

  invA = inv(A);

  for i=1:3
  for j=1:3
  for k=1:3
  for l=1:3
    M(i,j,k,l) = M(i,j,k,l) + 1/4*(invA(j,k)*xi(i)*xi(l) + invA(i,k)*xi(j)*xi(l
        ) + invA(j,l)*xi(i)*xi(k) + invA(i,l)*xi(j)*xi(k))*sin(theta)*w(ii)*w(jj
        )*pi*pi/2;
  end
  end
  end
  end

end
end

```

```
% Eshelby's tensor:
s = zeros(3,3,3,3);

for i=1:3
for j=1:3
for k=1:3
for l=1:3
for m=1:3
for n=1:3
    s(i,j,k,l) = s(i,j,k,l) + 1/4/pi*M(i,j,m,n)*c(m,n,k,l);
end
end
end
end
end
end

% Convert Eshelby's tensor into Voigt notation:

S = [
    s(1, 1, 1, 1),   s(1, 1, 2, 2),   s(1, 1, 3, 3),   s(1, 1, 2, 3),   s(1, 1, 3, 1),   s(1, 1, 1, 2)
    s(2, 2, 1, 1),   s(2, 2, 2, 2),   s(2, 2, 3, 3),   s(2, 2, 2, 3),   s(2, 2, 3, 1),   s(2, 2, 1, 2)
    s(3, 3, 1, 1),   s(3, 3, 2, 2),   s(3, 3, 3, 3),   s(3, 3, 2, 3),   s(3, 3, 3, 1),   s(3, 3, 1, 2)
  2*s(2, 3, 1, 1), 2*s(2, 3, 2, 2), 2*s(2, 3, 3, 3), 2*s(2, 3, 2, 3), 2*s(2, 3, 3, 1), 2*s(2, 3, 1, 2)
  2*s(3, 1, 1, 1), 2*s(3, 1, 2, 2), 2*s(3, 1, 3, 3), 2*s(3, 1, 2, 3), 2*s(3, 1, 3, 1), 2*s(3, 1, 1, 2)
  2*s(1, 2, 1, 1), 2*s(1, 2, 2, 2), 2*s(1, 2, 3, 3), 2*s(1, 2, 2, 3), 2*s(1, 2, 3, 1), 2*s(1, 2, 1, 2)
  ];
S

e = S*e_0';
e'


disp('Program has finished!')
```

2 Finite element simulation - verification of the semi-analytical implementation

Due to the intricate and error-prone nature of deriving the aforementioned equations, ensuring our confidence in their application necessitates a finite element simulation. This approach involves comparing the outcomes of a numerical finite element simulation solution with the semi-analytical solution outlined by the previous equations.
A finite element analysis is constructed encompassing three distinct cases of ellipsoids characterized by aspect ratios $a \times b \times c$: $1 \times 1 \times 1$, $1 \times 1 \times 2$, and $2 \times 1 \times 3$.
The finite element Representative Volume Elements (RVEs) are visually depicted below. It's worth noting that these RVEs possess finite dimensions, which stands in contrast to the Eshelby solutions derived for an infinitely large matrix. Consequently, this discrepancy may introduce some variations. However, as long as the matrix's dimensions are sufficiently large relative to the size of the ellipsoid, any resulting error should remain relatively minor.
Let us generate a random eigen strain tensor

$$\varepsilon_0 = \begin{bmatrix} 0.023522 & 0.076558 & 0.019838 & 0.068208 & 0.059107 & 0.031905 \end{bmatrix}^T \tag{10}$$

Given this eigen strain, the strain generated within the ellipsoid is predicted using the semi-analytical

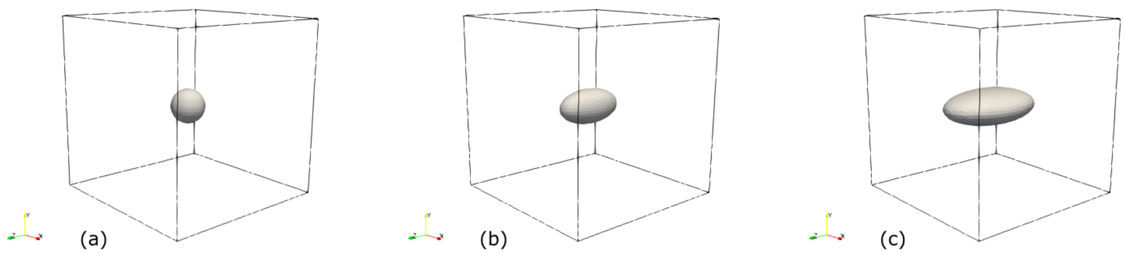

Figure 1: Ellipsoids of aspect ratios a) 1x1x1, b) 1x1x2, and c) 2x1x3.

solution (see Matlab code) and using the finite element simulation. The results are compared in the table below.

Table 1: Semi-analytical solutions and finite element simulation solutions of the ellipsoid strain components for the given eigen strain tensor.

	$\boldsymbol{\varepsilon}_0$	$\boldsymbol{\varepsilon}^{(1\times1\times1)}$		$\boldsymbol{\varepsilon}^{(1\times1\times2)}$		$\boldsymbol{\varepsilon}^{(2\times1\times3)}$	
		FEM	analytical	FEM	analytical	FEM	analytical
xx	0.023522	0.016544	0.016907	0.0197261	0.020092	0.0106386	0.009391
yy	0.076558	0.042158	0.042171	0.0507705	0.050056	0.0676905	0.067769
zz	0.019838	0.014737	0.015157	0.0064683	0.006845	0.0050078	0.0051844
yz	0.068208	0.032787	0.032480	0.0316119	0.030895	0.0429553	0.04018
zx	0.059107	0.028448	0.028146	0.0273814	0.026772	0.0220308	0.019031
xy	0.031905	0.015371	0.015187	0.0184856	0.018011	0.0208782	0.018065

An important observation to make is that in the isotropic scenario, the outcomes remain unaffected by the Young's modulus E, manifesting a dependency solely on the Poisson's ratio ν.
To illustrate the convergence of the semi-analytical solutions across varying numbers of Gauss integration points towards the numerical finite element solution, refer to the figures provided below. These figures distinctly demonstrate the alignment between the results. The dashed lines in the figures represent the finite element simulation solutions for the six components of the strain tensor, while the circled lines represent the semi-analytical solutions for varying number of Gauss integration points with increased number of integration points leading to a more precise result.

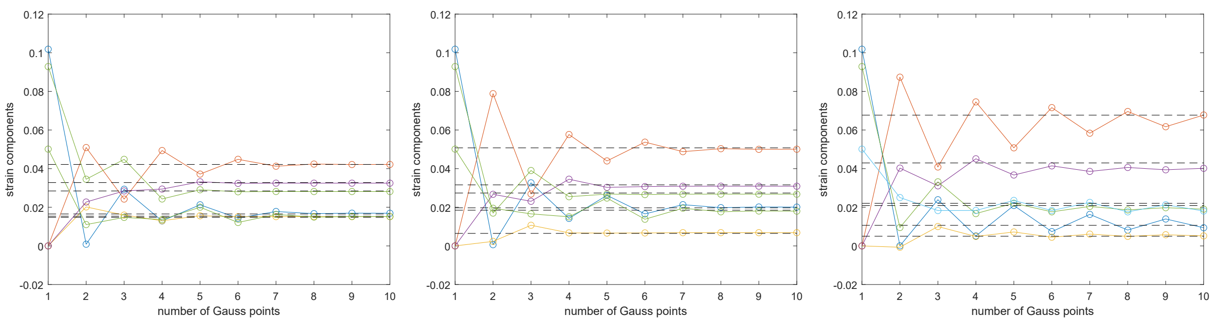

Figure 2: Sensitivity analysis demonstrating the convergence of numerical integration using the Gauss method (marked lines) and the solution obtained from the finite element simulation (dashed lines).

Below, the outcomes of the finite element simulations for the three outlined cases are displayed. These visualizations depict the ε_{yy} component of the strain field, effectively illustrating the uniform nature of strain (and stress) within the ellipsoid. This corroborates the findings derived by Eshelby. A notable observation is the remarkable concurrence between the numerical outcomes and the semi-analytical solutions derived for an infinitely extensive matrix, despite the presence of stress-free boundaries.
In the ultimate assessment, the case involving dimensions $2 \times 1 \times 3$ is explored, featuring a random eigen strain ε_{ij}^0 coupled with a fully anisotropic tensor of elastic constants C_{ijkl}. Importantly, when generating a random elastic tensor matrix, it's imperative to verify its positive definiteness. For instance

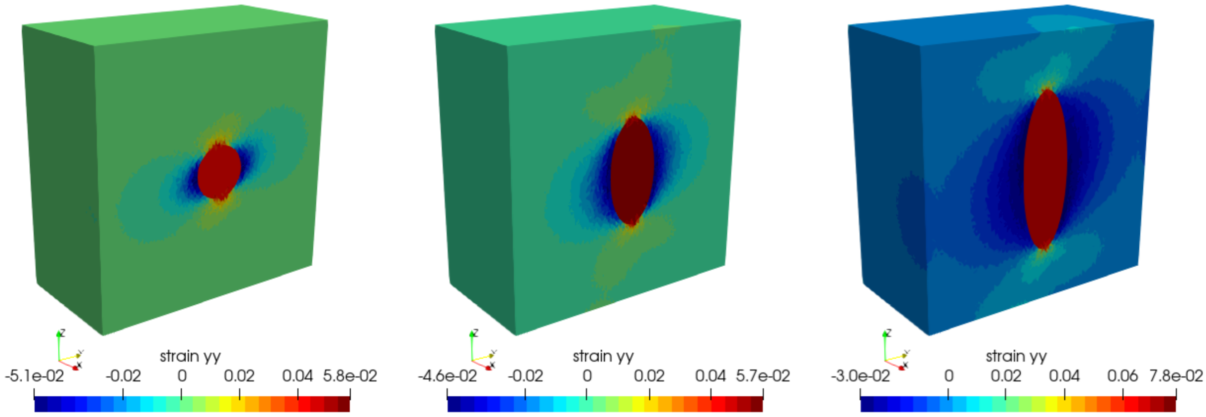

Figure 3: Strain component ε_{yy} for the three ellipsoid cases. Notice that the strain field within the ellipsoid is uniform.

using Matlab, this can be accomplished by employing the `chol(C)` function, which would issue an error message if matrix ‘C‘ lacks positive definiteness.
For the following case of eigen strain and tensor of elastic constants

$$\varepsilon_0 = \begin{bmatrix}0.023522 & 0.076558 & 0.019838 & 0.068208 & 0.059107 & 0.031905\end{bmatrix}^T \tag{11}$$

$$\boldsymbol{C} = \begin{bmatrix} 1.9627 & 0.71094 & 1.2973 & 2.1394 & 1.908 & 1.1079 \\ & 0.88304 & 0.57145 & 1.129 & 0.85012 & 0.76032 \\ & & 1.7383 & 1.7461 & 1.5757 & 0.97334 \\ & & & 2.8236 & 2.6732 & 1.5282 \\ & & & & 2.9695 & 1.4951 \\ & & & & & 1.1461 \end{bmatrix} \tag{12}$$

the results of the semi-analytical and numerical solutions are collated in the figure below. Also, strain field within and around the ellipsoidal inclusion is presented.
A noteworthy observation is the heightened complexity of the strain field surrounding the precipitate and reaching the domain boundaries. In terms of the semi-analytical solution, it becomes evident that an increased number of Gauss integration points would facilitate even stronger convergence. Nevertheless, it is evident that the semi-analytical results and the finite element simulation outcomes align remarkably well also for a case of fully anisotropic material.

3 Special cases - closed form solutions

The previous section explored an arbitrary ellipsoid within an anisotropic material. However, by imposing specific assumptions about the ellipsoid’s shape and considering simplified anisotropic conditions, it is possible to derive closed-form analytical solutions. Several of such examples are considered below:

3.1 Isotropic sphere

For the case of a spherical inclusion and isotropic material, the following expression can be derived [3]:

$$S_{ijkl} = \left(1 - \frac{5}{3}\beta\right) I_{ij} I_{kl} + \frac{1}{2}\beta \left(I_{ik} I_{jl} + I_{il} I_{jk} - \frac{2}{3} I_{ij} I_{kl}\right) \tag{13}$$

where $\beta = \frac{10\nu - 8}{15(\nu - 1)}$
Below is a Matlab code. You can easily verify that the outcomes of both the general code and this specific code will align when dealing with an isotropic spherical inclusion.

Listing 2: Matlab example

```
% Eigen strain:
```

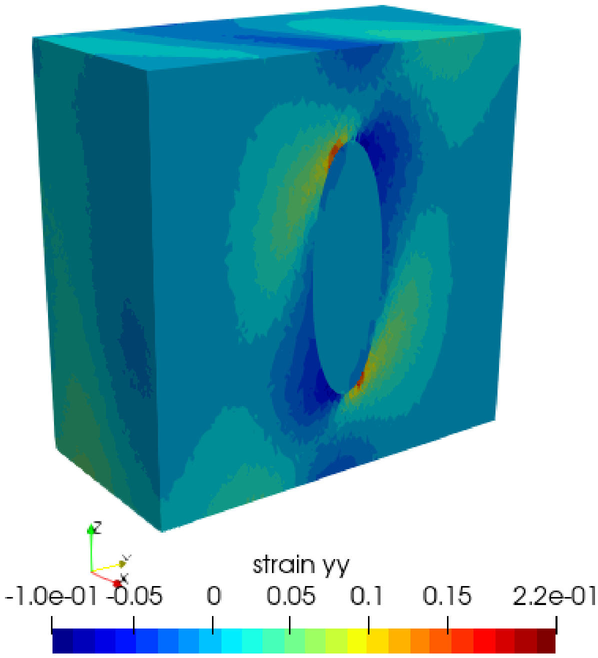

Figure 4: Strain component ε_{yy} for the fully anisotropic case.

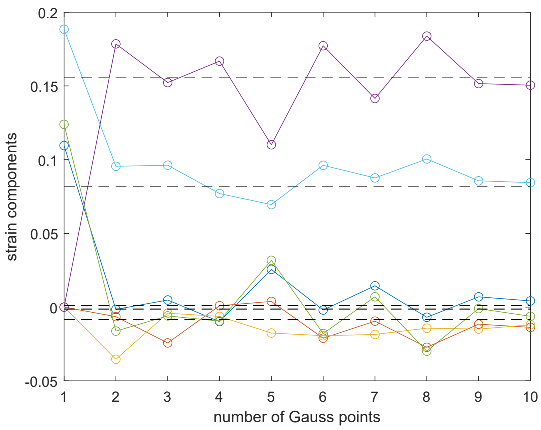

Figure 5: Convergence analysis of the Gauss method.

```
e_0 = [0.023522, 0.076558, 0.019838, 0.068208, 0.059107, 0.031905];

% Poisson's ratio:
v = 0.3;

  beta = (10.0*v-8.0)/(15.0*v-15.0)

  I3 = eye(3);
  for i=1:3
  for j=1:3
  for k=1:3
  for l=1:3
    s(i,j,k,l) = (1.0-5.0/3.0*beta)*I3(i,j)*I3(k,l) + 0.5*beta*(I3(i,k)*I3(j,l)
         + I3(i,l)*I3(j,k) -2.0/3.0*I3(i,j)*I3(k,l));
  end
  end
```

```
  end
  end

  S = [
    s(1, 1, 1, 1),   s(1, 1, 2, 2),   s(1, 1, 3, 3),   s(1, 1, 2, 3),   s(1, 1,
        3, 1),   s(1, 1, 1, 2)
    s(2, 2, 1, 1),   s(2, 2, 2, 2),   s(2, 2, 3, 3),   s(2, 2, 2, 3),   s(2, 2,
        3, 1),   s(2, 2, 1, 2)
    s(3, 3, 1, 1),   s(3, 3, 2, 2),   s(3, 3, 3, 3),   s(3, 3, 2, 3),   s(3, 3,
        3, 1),   s(3, 3, 1, 2)
  2*s(2, 3, 1, 1), 2*s(2, 3, 2, 2), 2*s(2, 3, 3, 3), 2*s(2, 3, 2, 3), 2*s(2, 3,
      3, 1), 2*s(2, 3, 1, 2)
  2*s(3, 1, 1, 1), 2*s(3, 1, 2, 2), 2*s(3, 1, 3, 3), 2*s(3, 1, 2, 3), 2*s(3, 1,
      3, 1), 2*s(3, 1, 1, 2)
  2*s(1, 2, 1, 1), 2*s(1, 2, 2, 2), 2*s(1, 2, 3, 3), 2*s(1, 2, 2, 3), 2*s(1, 2,
      3, 1), 2*s(1, 2, 1, 2)
  ];
S

% Strain within the inclusion:
S*e_0'
```

3.2 2D elliptical isotropic inclusion

For a 2D case of an elliptical inclusion (or a cylinder in 3D) and isotropic material, the following can be derived. As highlighted earlier, in the isotropic case, the Eshelby's tensor depends only on the Poisson's ratio ν:

$$\begin{aligned}
S_{1111} &= \frac{a_2\left[2(1-\nu)(a_1+a_2)+a_1\right]}{2(1-\nu)(a_1+a_2)^2} \\
S_{2222} &= \frac{a_1\left[2(1-\nu)(a_1+a_2)+a_2\right]}{2(1-\nu)(a_1+a_2)^2} \\
S_{3333} &= 0 \\
S_{1122} &= \frac{a_2\left[-(1-2\nu)a_1+2\nu a_2\right]}{2(1-\nu)(a_1+a_2)^2} \\
S_{2211} &= \frac{a_1\left[-(1-2\nu)a_2+2\nu a_1\right]}{2(1-\nu)(a_1+a_2)^2} \\
S_{1133} &= \frac{\nu a_2}{(1-\nu)(a_1+a_2)} \\
S_{3311} &= 0 \\
S_{1313} &= \frac{a_2(2-\nu)}{2(1-\nu)(a_1+a_2)} \\
S_{2323} &= \frac{a_1(2-\nu)}{2(1-\nu)(a_1+a_2)}
\end{aligned} \tag{14}$$

Finally, symmetries are applied:

$$S_{ijkl} = S_{ijkl} = S_{ijlk} = S_{jilk}$$

A Matlab code is provided below:

Listing 3: Matlab example

```
% Ellipse:
a1=1
a2=2

% Poisson's ratio:
nu=0.3

```

```
% Transformation strain:
e_0 = [1.1  0 0;0 -1.1 0;0 0 0]

% Eshelby' strain tensor:
S(1,1,1,1) = a2*(2*(1-nu)*(a1+a2)+a1)/2/(1-nu)/(a1+a2)^2;
S(2,2,2,2) = a1*(2*(1-nu)*(a1+a2)+a2)/2/(1-nu)/(a1+a2)^2;
S(1,1,2,2) = a2*(-(1-2*nu)*a1+2*nu*a2)/2/(1-nu)/(a1+a2)^2;
S(2,2,1,1) = a1*(-(1-2*nu)*a2+2*nu*a1)/2/(1-nu)/(a1+a2)^2;
S(1,1,3,3) = nu*a2/(1-nu)/(a1+a2);
S(1,3,1,3) = a2*(2-nu)/2/(1-nu)/(a1+a2);
S(2,3,2,3) = a1*(2-nu)/2/(1-nu)/(a1+a2);

S(3,1,1,3) = a2*(2-nu)/2/(1-nu)/(a1+a2);
S(3,2,2,3) = a1*(2-nu)/2/(1-nu)/(a1+a2);

S(1,3,3,1) = a2*(2-nu)/2/(1-nu)/(a1+a2);
S(2,3,3,2) = a1*(2-nu)/2/(1-nu)/(a1+a2);

S(3,1,3,1) = a2*(2-nu)/2/(1-nu)/(a1+a2);
S(3,2,3,2) = a1*(2-nu)/2/(1-nu)/(a1+a2);

e=zeros(3,3);

for i=1:3
for j=1:3
for k=1:3
for l=1:3
    e(i,j) = e(i,j) + S(i,j,k,l)*e_0(k,l);
end
end
end
end

% Strain within the inclusion:
e
```

References

[1] Renaud Masson. New explicit expressions of the hill polarization tensor for general anisotropic elastic solids. *International Journal of Solids and Structures*, 45(3):757–769, 2008.

[2] Alexander P. Suvorov and George J. Dvorak. Rate form of the eshelby and hill tensors. *International Journal of Solids and Structures*, 39(21):5659–5678, 2002.

[3] Elastic-plastic behaviour of polycrystalline metals and composites. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 319:247–272, 10 1970.