PX920 Workshop: Computation of the effective elasticity matrix

Łukasz Figiel

WMG, University of Warwick

Effective modulus from the theory of homogenization

The effective (**homogenized**) (**Young's modulus**) as derived from the theory of homogenization (*0th-order terms*) is given by

$$E^{eff} = \frac{1}{|\Theta|} \int_0^{\Theta} E(y) \left(\frac{\partial \chi}{\partial y} + 1 \right) \, \mathrm{d}y \tag{1}$$

Substituting the linear elastic law

$$\bar{\sigma} = E^{eff} \bar{\varepsilon} \tag{2}$$

into (1) one obtains the material law as follows

$$\bar{\sigma} = \frac{1}{|\Theta|} \int_0^{\Theta} E(y) \left(\frac{\partial \chi}{\partial y} + 1 \right) \, \mathrm{d}y \; \bar{\varepsilon} \tag{3}$$

where $\bar{\sigma}$ and $\bar{\varepsilon}$ are average stresses and strains, respectively.

Effective elastic modulus/elasticity matrix

The average stress $\bar{\sigma}$ is given by

$$\bar{\sigma} = \frac{1}{L} \int_L \sigma \, \mathrm{d}y \tag{4}$$

Then, if one assumed $\bar{\varepsilon} = 1$, then effective modulus can be given by

$$E^{eff} = \bar{\sigma} = \frac{1}{L} \int_L \sigma \, \mathrm{d}y \tag{5}$$

The above can be generalised into 3D

$$\boldsymbol{C}^{eff} = \bar{\boldsymbol{\sigma}} = \frac{1}{V} \int_V \boldsymbol{\sigma} \, \mathrm{d}V \tag{6}$$

where $\boldsymbol{C}^{eff}$ denotes the effective elasticity matrix.

Unit strain cases

The RVE problem is solved using the finite-element method, and it must be solved for as many right-hand side vectors as there are unit strain components in the problem - here (2D), there are three strain components. The unit strain vectors (Voigt notation) are applied to each element as

$$\varepsilon_{u11} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \varepsilon_{u22} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad , \quad \gamma_{u12} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{7}$$

Unit strain cases

Each strain state results in a nodal displacement field, where stresss in each element of the RVE must be calculated. The calculated element stresses represent a column in the elasticity matrix of the element as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}^e = \begin{bmatrix} C_{1111} & C_{1122} & C_{1112} \\ C_{2211} & C_{2222} & C_{2212} \\ C_{1211} & C_{1222} & C_{1212} \end{bmatrix}^e \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix}^e \quad (8)$$

Pseudo-code for the linear homogenisation - Part 1

Input and Mesh Generation

- Coordinates of the RVE boundaries: **top**,**bot**,**left**,**right**
- Material/element properties: Young's modulus **E**, Poisson's ratio **v**, thickness (**th**), **PlaneStrain/PlaneStress**
- Call **mesh**: returns arrays of nodal coordinates (**XYZ**), element node numbers (**CON**), element DOFs (**DOF**)
- Call **BCcorner_fun**: returns an array (**BCcorner**) containing IDs of corner nodes of an RVE

Pseudo-code for the linear homogenisation - Part 2

- Calculate element's constitutive matrix $\boldsymbol{C}^e$
- Call **T_matrix**: returns the transformation matrix $\boldsymbol{T}$
- Call **K_matrix**: returns the global stiffness matrix $\boldsymbol{K}$ - one needs to evaluate element contributions $\boldsymbol{B}^e$, $\det \boldsymbol{J}^e$ and $\boldsymbol{K}^e$
- Calculate the modified stiffness matrix $\boldsymbol{K}_m$
- Calculate **HomoC** (effective elasticity matrix)

for i in range(3):

$\varepsilon[i, 0] = 1$

Call **r_vector**: returns the global right-hand side vector $\boldsymbol{r}$ (see Eq. (29) in Lecture 2) - requires element right-hand side vectors $\boldsymbol{r}^e$

Calculate the modified external load vector $\boldsymbol{r}_m = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{r}$

Calculate modified nodal displacements $\boldsymbol{d}_m = \boldsymbol{K}_m^{-1}\boldsymbol{r}_m$

Calculate nodal displacements $\boldsymbol{d} = \boldsymbol{T}\boldsymbol{d}_m$

Call **sigmaHomo_vector**: returns **sigmaHomo**

HomoC[:,i] = **sigmaHomo**

Average stress calculation (**sigmaHomo_vector**) - Part 1

1. **Input**
   - ε_u : unit strain
   - $\boldsymbol{d}$: total displacement vector including corner DOFs
   - location of Gauss points: **Gauss**=[-1 1 1 -1, -1 -1 1 1]/$\sqrt{3}$
   - weight functions: **w**$= 1$
   - zero volume: **vol**$= 0$
   - create stress array: **sigma**$= np.zeros(3, 1)$

Average stress calculation (**sigmaHomo_vector**) - Part 2

2. **Loop over all elements**

   for i in range(number of elements):

   $\boldsymbol{C}^e$: constitutive (elasticity matrix) for element i

   id: DOF number for element i

   $\boldsymbol{d}^e = \boldsymbol{d}$ (**id**): nodal element displacements of element i

   for j in range(number of Gauss points):

   xi=Gauss[0,j]

   eta=Gauss[1,j]

   Call function **dispstrain_B** to calculate $\boldsymbol{B}_I$ and det $\boldsymbol{J}$

   sigmaGauss=$\boldsymbol{C}^e(\boldsymbol{B}_I\boldsymbol{d}^e + \varepsilon_u)$

   sigma=sigma + sigmaGauss det $\boldsymbol{J}$ w

   vol=vol+det $\boldsymbol{J}$

3. **sigmaHomo**=sigma/vol

Element RHS vector (**r_vector_e**)

1. **Input parameters**: xyze, Ce, th, eps

   r_vector_e=np.zeros((8,1))

   a=1/(np.sqrt(3))

   w=1

   Gauss = np.array([[-a, a, a, -a],[-a, -a, a, a]])

2. **Loop over all integration points within an element**

   for j in range(number of Gauss points):

   ξ=Gauss[0,j]

   η=Gauss[1,j]

   Call function **dispstrain_B** to calculate $\boldsymbol{B}_j$ and det $\boldsymbol{J}$

   rhs=$\boldsymbol{B}_j^T \boldsymbol{C}^e \boldsymbol{\varepsilon}_u$

   r_vector_e=r_vector_e + rhs det $\boldsymbol{J}$ w

Global RHS vector (**r_vector**)

1. **Input parameters**: XYZ, CON, DOF, C, th, eps, BCcorner

    nel=len(CON)
    ndof=2*len(XYZ)
    r_vector=np.zeros((ndof,1))

2. **Loop over all elements**

    for i in range(number of elements):

        id=DOF[i,:]
        xyze=XYZ[CON[i,:],:]
        Ce=C[3*i:3*i+3,0:3]
        Call function **r_vector_e**(xyze,Ce,th,eps_u)
        r_vector[np.ix_(id)] = r_vector[np.ix_(id)] + r_vector_e

    r_vector= - r_vector[np.ix_(BCcorner)]