Homogenization for Multi Field Modelling

Part I: Theories and FEniCS

Yi Hu

April 16, 2016

Contents

In this student project, computational homogenization is utilized to investigate behaviour of composites under multiple fields. Many applications can be found in the field of material modelling, specifically for coupled problems of composites, e.g. electroactive polymers. As for implementation, a novel Finite Element framework, FEniCS, is investigated. FEniCS is a collection of libraries and modules, which uses Python (or C++) as its interface language. The most important feature about FEniCS is its specialization in code generation with respect to bilinear forms, linear forms and function spaces in the formulation of computational problems, which can translate the mathematical language directly into codes and accelerates the trial process of new models and new computation methods. Moreover, Python make the implementation of a problem even faster, as it is a dynamical language for prototyping. In the framework of FEniCS a unit cell module is realized, where the calculation including homogenized properties of composites in micro scale under multiple fields could be performed.

1 Introduction

Composite materials play an important role in engineering, as they combine the advantages of each components in materials. More specifically they have more specific strengths, can be tailored to their optimum to satisfy design requirements. Their advantage in material fatigue and their resistance against corrosion plus their light weight draw much attention from industry [2]. Many applications can be found in the field of Civil Engineering, Aerospace Engineering, etc. In order to understand composites thoroughly, experiments and simulations are accomplished. The current work specialized in the simulation of composites.

A multi scale model is conducive in modelling, where the morphology of composite materials in small scale affects the material behaviour in macro scale. Additionally the material components are of great importance to the overall macro behaviour as well. If an energy function is used to describe the composites, material components and geometry configuration in micro scale determines total material energy. Under this condition, an appropriate energy function is sought after. However, the exact expression for composites under different loads and with various inclusion geometries is hard to obtain. From a practical perspective, models in multiple scales would lead to full understanding of composite, which will in return be beneficial for material design. Moreover, the computation of separate scales would benefit from various computing techniques such as parrellelization [6] and model reduction [7]. These techniques would possibly accelerate the speed of computation.

There are several approaches to multi scale modelling. Theoretical results are available in simple inclusion geometry cases, e.g. circle inclusion, which go backs to Eshelby [15]. However only specific types of inclusions are handled. Another consideration is to acquire material description on the fly, i.e. to embed the result of micro scale simulation into macro scale simulation. Many contributions are credited to Miehe and coworkers [22] [19] [18] [20].

In this work computational homogenization is used to achieve multi scale modelling. The formulation of multi scale and multi field problems is motivated by [27]. The key task of homogenization is to calculate homogenized parameters for a macro scale model, which is mainly based on its micro scale results. The formulation of homogenized parameters is carried out through material energy. Using material energy in the formulation is of great importance, as it simplifies the derivation of equilibrium and unifies the calculation of homogenized parameters. This approach turns the problem into an optimization problem [24], where many tools in optimization can be utilized in this context. Moreover in the case of composite materials in multiple fields, energy formulation will result in relative simple formulation in the homogenized context [25].

To avoid the lengthy derivation of equations for coupled fields, calculation under FEniCS framework is realized. Its strength lies in code generation and convenient formulation of applied mathematics problems, especially in the field of coupled problems, which we refer to [29] in its later chapters.

The whole report will fall into two parts. This part concentrates on the theory basics of simulation. Theory of homogenization is briefly covered in the next section. We start the discussion with mathematical point of view and end with Hill-Mandel condition and its position in FE^2 scheme. The novel FE framework FEniCS is then clarified in the following. We mainly emphasize its features in multi field modelling and composite material modelling. The implementation and numerical examples are elaborated in detail in the second part of the report.

2 Computational Homogenization

The method used in the current work is computational homogenization. It was proposed in 1970s by Babuška and collaborators [1], while the conception of this method appeared already in the late 60s, where the one-dimensional example and examples of layered materials are already investigated [3]. As indicated in the name of this method, the goal is to approximate a heterogeneous problem with a homogeneous problem. The essential idea of this method is to make use of the scale separation, so that a reduced PDE for the macroscopic problem is obtained. A macroscopic problem always contains the information from its corresponding micro scale, and it is often represented as "effective parameters". These effective parameters are often calculated in the sense of an "averaging procedure" or "homogenization". In order to obtain homogenized quantities a micro scale problem needs to be solved. It could be solved with numerical methods such as Finite Element Method, Finite Volume Method or theoretical results in simple cases. An extensive review can be found in [5]. A more general framework, the Heterogeneous Multiscale Method (HMM), was proposed by Bjorn Engquist. This method extend the idea of homogenization and introduce generic methodology between macro scale and micro scale. An introductory review could be found in [28].

The convergence of this method is given by mathematicians, which are described in detail in [3]. Comparison with experiments are also accomplished, which is presented in [13]. Modelling examples are abundant in literatures, e.g. plasticity modelling [23], size effect and damage modelling [4].

In this part we first illustrate the key idea of homogenization and scale separation. Then a one dimensional example is presented. Analogously we discuss the application to the 3D elliptic PDE briefly. Hill-Mandel requirements should be fulfilled in the energy conserving problem. Brief introduction of FE^2 scheme is stated in the end of this part. We confine our discussion here mainly on materials with periodic structures.

2.1 Periodic Structures

Periodicity appears frequently in composite materials, for instance material with fiber or particle reinforcement. In these materials inclusions are arranged periodically. Material parameters can be expressed with periodic functions of coordinates. For example, Young's Modulus can be written in the following form,

$$\mathbb{C}(\mathbf{x}+\mathbf{Y}) = \mathbb{C}(\mathbf{x}). \tag{2.1}$$

It is often the case that discontinuity exists in composite materials. This characteristic comes from their different material properties such as Young's Modulus above. We want to represent this heterogeneous material with a homogenized one, where we have no discontinuity in material parameters and the original composite material can be treated as a homogeneous one with only one component. In addition this homogenized material should resemble the material behaviour of the heterogeneous one.

We denote ϵ as a periodicity parameter, or in other words "the order of homogeneity". Then material parameters such as Young's modulus $\mathbb{C}$ can be written as

$$\mathbb{C}^{\epsilon}(\mathbf{x}) = \mathbb{C}\left(\frac{\mathbf{x}}{\epsilon}\right). \tag{2.2}$$

Assumed that $\mathbb{C}$ is $\mathbf{Y}$-periodic and a reference cell of size $\mathbf{Y}$ is chosen, after transformation $\mathbb{C}^{\epsilon}$ becomes a function with period $\epsilon\mathbf{Y}$. When ϵ goes to 0, the composite material becomes more homogeneous (for a small neighbourhood the constitutes almost the same). This process is shown in the following illustration. In other words the underlying meaning of this transformation is a process of homogenization. We should notice that there are some discrepancies in the formulation of homogenization from the mathematical point of view (scale separation) and the later introduced engineering view (RVE based simulation). However the end results are the same.

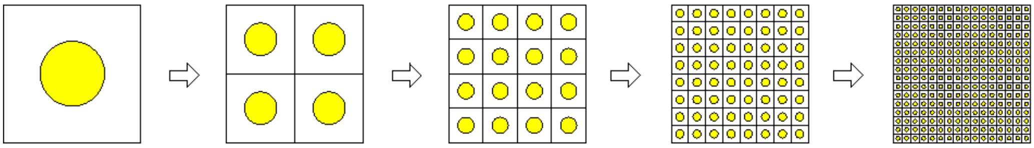

Figure 2.1: Homogenization process with $\epsilon \to 0$

2.2 Scale Separation

When a two scale problem is addressed, a corresponding field variable could be expanded as follows,

$$\mathbf{\Phi}^\epsilon(\mathbf{x}) = \mathbf{\Phi}^0(\mathbf{x},\mathbf{y}) + \epsilon\mathbf{\Phi}^1(\mathbf{x},\mathbf{y}) + \epsilon^2\mathbf{\Phi}^2(\mathbf{x},\mathbf{y}) + \cdots, \tag{2.3}$$

In the above formula, $\mathbf{x}$ is the position vector of a point, which is deemed as the *macroscopic* coordinate. $\mathbf{y} = \mathbf{x}/\epsilon$ is a *microscopic* coordinate, which stands for *rapid* oscillation. The physical nature of the right hand side is the decomposition of macro scale dependency and micro scale dependency with respect to a reference cell. The purpose of setting $\mathbf{y} = \mathbf{x}/\epsilon$ is to achieve a closed form expressed with the original coordinate. The ratio ϵ means that the micro quantity will vary $1/\epsilon$ faster than macroscopic level. When ϵ goes to 0, functions $\mathbf{\Phi}^0(\mathbf{x},\mathbf{y}), \mathbf{\Phi}^1(\mathbf{x},\mathbf{y}), \cdots$ are smooth functions with respect to macroscopic coordinate and microscopic coordinate.

We should notice that functions $\mathbf{\Phi}^i(\mathbf{x},\mathbf{y})$ have the following properties,

$$\begin{cases} \mathbf{\Phi}^i(\mathbf{x},\mathbf{y}) \text{ is defined for } \mathbf{x} \in \Omega \text{ and } \mathbf{y} \in \mathbf{Y} \\ \mathbf{\Phi}^i(\cdot,\mathbf{y}) \text{ is } \mathbf{Y}\text{-periodic} \end{cases} \tag{2.4}$$

For details we refer to chapter 6 of [3]. These two properties characterize the scale separation, since the first variable $\mathbf{x}$ defined in the whole domain is clearly the *macroscopic* coordinate, while the second variable $\mathbf{y}$ is a coordinate in the reference cell. The setting $\mathbf{y} = \mathbf{x}/\epsilon$ describes the scale size of this problem, or the "order of homogenization" as discussed in the previous section. If $\mathbf{Y}$ has the form $(l_1, \cdots, l_N)$, then $\mathbf{x}/\epsilon = \mathbf{k}_l + \mathbf{y}$, where $\mathbf{k}_l = (k_1 l_1, \cdots, k_N l_N)$ and $k_i \in \mathbb{N}$. This shows the decomposition is valid for two scale problem and the setting $\mathbf{y} = \mathbf{x}/\epsilon$ is meaningful.

The micro-macro variable decomposition is illustrated in the following figure.

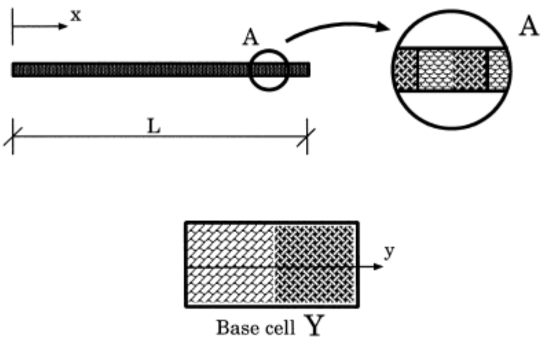

Figure 2.2: Micro and Macro Coordinate of Field Variable [10]

2.3 One Dimensional Problem

Many books and review papers begin with a one dimensional problem, such as [3]. Here we briefly go through a one dimensional elasticity problem. More detailed derivation could be found in [10].

The governing equations, i.e. the equilibrium and Hooke's law are,

$$\begin{cases} \dfrac{\mathrm{d}\sigma^\epsilon}{\mathrm{d}x} + \gamma^\epsilon = 0 \\ \sigma^\epsilon = E^\epsilon \dfrac{\mathrm{d}u^\epsilon}{\mathrm{d}x}, \end{cases} \tag{2.5}$$

Noting that ϵ in superscript represents its periodic property. γ^ϵ is the body load of the material. If E^ϵ and γ^ϵ are uniform in the macro coordinate and only differ inside each cell, then the following relation holds,

$$E^\epsilon(x, x/\epsilon) = E^\epsilon(x/\epsilon) = E(y), \tag{2.6}$$

The relation with respect to body load is likewise. Regarding the double scale expansion according to (2.3) it follows with,

$$\begin{cases} u^\epsilon(x) = u^0(x,y) + \epsilon u^1(x,y) + \epsilon^2 u^2(x,y) + \cdots \\ \sigma^\epsilon(x) = \sigma^0(x,y) + \epsilon\sigma^1(x,y) + \epsilon^2\sigma^2(x,y) + \cdots, \end{cases} \tag{2.7}$$

We employ the chain rule (similar for $\sigma^i(x,y)$),

$$\frac{\mathrm{d}u^i(x,y)}{\mathrm{d}x} = \frac{\partial u^i(x,y)}{\partial x} + \frac{\partial u^i(x,y)}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} \overset{y=x/\epsilon}{=} \frac{\partial u^i(x,y)}{\partial x} + \frac{1}{\epsilon}\frac{\partial u^i(x,y)}{\partial y} \tag{2.8}$$

After substitution and equating the correspondent terms, we have

$$\begin{cases} 0 = E(y)\left(\dfrac{\partial u^0}{\partial y}\right) \\ \sigma^0 = E(y)\left(\dfrac{\partial u^0}{\partial x} + \dfrac{\partial u^1}{\partial y}\right) \\ \sigma^1 = E(y)\left(\dfrac{\partial u^1}{\partial x} + \dfrac{\partial u^2}{\partial y}\right), \end{cases} \tag{2.9}$$

and

$$\begin{cases} \dfrac{\partial \sigma^0}{\partial y} = 0 \\ \dfrac{\partial \sigma^0}{\partial x} + \dfrac{\partial \sigma^1}{\partial y} + \gamma(y) = 0, \end{cases} \tag{2.10}$$

Combining the first two equations in (2.7) we have,

$$\sigma^0 = E(y)\left(\frac{\partial u^0}{\partial x} + \frac{1}{\epsilon}\frac{\partial u^0}{\partial y} + \frac{\partial u^1}{\partial y}\right) = E(y)\left(\frac{\mathrm{d}u^0}{\mathrm{d}x} + \frac{\partial u^1}{\partial y}\right) \tag{2.11}$$

It follows with,

$$\frac{\sigma^0}{E(y)} = \frac{\mathrm{d}u^0}{\mathrm{d}x} + \frac{\partial u^1}{\partial y}$$

Notice the periodicity for the second variable (2.4) integration of y in a reference cell yields

$$\sigma^0(x) = \left(Y/\int_Y \frac{\mathrm{d}y}{E(y)}\right)\frac{\mathrm{d}u^0(x)}{\mathrm{d}x}. \tag{2.12}$$

Define the *homogenized modulus of elasticity* as follows,

$$E^H = 1/\left(\frac{1}{Y}\int_0^Y \frac{\mathrm{d}\eta}{E(\eta)}\right). \tag{2.13}$$

Then the original problem is transformed to

$$\begin{cases} \sigma^0(x) = E^H\dfrac{\mathrm{d}u^0(x)}{\mathrm{d}x} \\ \dfrac{\mathrm{d}\sigma^0}{\mathrm{d}x} + \bar{\gamma} = 0, \end{cases} \tag{2.14}$$

where $\bar{\gamma} = 1/Y\int_Y \gamma(y)$ is the average of γ inside the cell, which is derived analogously using (2.10). From (2.14) the differential equation for displacement holds as

$$\frac{\mathrm{d}^2u^0(x)}{\mathrm{d}x^2} = -\frac{\bar{\gamma}}{E^H} \tag{2.15}$$

Taking account of the boundary condition on both ends gives the result

$$u(x) = -\frac{\bar{\gamma}}{E^H}\frac{x^2}{2} + \frac{\bar{\gamma}}{E^H}Lx$$

2.4 General Elliptical PDE

If a general PDE for three dimensional problem is taken into account, it would be more intricate, as the solution is often sought in the sense of weak form. In this circumstance a homogenized weak form is considered instead of the homogenized differential operator. Then the limit of homogenized weak form should converge to the weak form without homogenization, which is called *G-convergence* [12]. As for the case of elasticity tensor in the sense of differential operator G-convergence is expressed as

$$\lim_{\epsilon\to 0}\frac{\partial}{\partial x_i}\left[C^{\epsilon}_{ijkl}\frac{\partial u^{\epsilon}_k}{\partial x_l}\right] \to \frac{\partial}{\partial x_i}\left[\bar{C}_{ijkl}\frac{\partial u^0_k}{\partial x_l}\right] \tag{2.16}$$

It is worth noting that when $\epsilon \to 0$, $C^{\epsilon}_{ijkl} \to C^{0}_{ijkl}$ and $u^{\epsilon}_{k} \to u^{0}_{k}$. This does not mean u^{0}_{k} solves the original PDE with coefficients replaced by C^{0}_{ijkl}. The reason lies in that the convergence of C_{ijkl} and u_k is weakly convergent. In this context, the limit of product does not equal to the product of the convergent limits. We should specify a new $\bar{C}_{ijkl}$, which gives the result u^0. Thorough discussion on this point can be found in [3]. A quick overview of the general problem is given in the review [10]. Several key points of the general problem are listed here. With the notation of general elliptical operator using

$$\mathcal{A}^{\epsilon} = \frac{\partial}{\partial x_i}\left(a_{ij}(\mathbf{y})\frac{\partial}{\partial x_j}\right). \tag{2.17}$$

general problem could then be described as,

$$\begin{cases} \mathcal{A}^{\epsilon}\mathbf{u}^{\epsilon} = \mathbf{f} & \text{in } \Omega \\ \mathbf{u}^{\epsilon} = \mathbf{0} & \text{on } \partial\Omega \end{cases} \tag{2.18}$$

Employing the double scale expansion for both the field variable $\mathbf{u}^{\epsilon}$ and the differential operator $\mathcal{A}^{\epsilon}$, namely (notice that chain rule is applied when differentiating)

$$\begin{cases} \mathbf{u}^{\epsilon}(\mathbf{x}) = \mathbf{u}^0(\mathbf{x},\mathbf{y}) + \epsilon\mathbf{u}^1(\mathbf{x},\mathbf{y}) + \epsilon^2\mathbf{u}^2(\mathbf{x},\mathbf{y}) + \cdots \\ \mathcal{A}^{\epsilon} = \dfrac{1}{\epsilon^2}\mathcal{A}^1 + \dfrac{1}{\epsilon}\mathcal{A}^2 + \mathcal{A}^3 \end{cases} \tag{2.19}$$

Here $\mathcal{A}^1, \mathcal{A}^2, \mathcal{A}^3$ is defined as follows

$$\mathcal{A}^1 = \frac{\partial}{\partial y_i}\left(a_{ij}(\mathbf{y})\frac{\partial}{\partial y_j}\right);\ \mathcal{A}^2 = \frac{\partial}{\partial y_i}\left(a_{ij}(\mathbf{y})\frac{\partial}{\partial x_j}\right) + \frac{\partial}{\partial y_i}\left(a_{ij}(\mathbf{y})\frac{\partial}{\partial x_j}\right);\ \mathcal{A}^3 = \frac{\partial}{\partial x_i}\left(a_{ij}(\mathbf{y})\frac{\partial}{\partial x_j}\right).$$

Substitution with the above differential operators and comparing with the according terms give

$$\begin{cases} \mathcal{A}^1\mathbf{u}^0 = \mathbf{0} \\ \mathcal{A}^1\mathbf{u}^1 + \mathcal{A}^2\mathbf{u}^0 = \mathbf{0} \\ \mathcal{A}^1\mathbf{u}^2 + \mathcal{A}^2\mathbf{u}^1 + \mathcal{A}^3\mathbf{u}^0 = \mathbf{f}. \end{cases} \tag{2.20}$$

From [3] it is known that if a $\mathbf{Y}$-periodic function u has a unique solution in terms of $\mathcal{A}^1$ operator, i.e.

$$\mathcal{A}^1\mathbf{u} = \mathbf{F} \quad \text{in reference cell.} \tag{2.21}$$

Then the right hand side of the above equation, $\mathbf{F}$ should satisfy

$$\bar{\mathbf{F}} = \frac{1}{|Y|}\int_Y \mathbf{F}\,\mathrm{d}\mathbf{y} = \mathbf{0}. \tag{2.22}$$

Applying this proposition to (2.20) several times the field variable could be expressed with the following form,

$$\mathbf{u}^1(\mathbf{x},\mathbf{y}) = \chi^i(\mathbf{y})\frac{\partial\mathbf{u}(\mathbf{x})}{\partial x_j} + \xi(\mathbf{x}) \tag{2.23}$$

Function $\chi^i(\mathbf{y})$ is the local solution of this problem, which has $\mathbf{Y}$-periodic property. And $\xi(\mathbf{x})$ is the solution of macro scale. The local problem is

$$\mathcal{A}^1\chi^j(\mathbf{y}) = \frac{\partial a_{ij}(\mathbf{y})}{\partial y_i} \quad \text{in reference cell.} \tag{2.24}$$

Hence the macro scale problem (homogenized problem) can be written as

$$\mathcal{A}^H\mathbf{u} = \mathbf{f}, \tag{2.25}$$

with

$$\mathcal{A}^H = a^H_{ij}\frac{\partial^2}{\partial x_i\partial x_j}. \tag{2.26}$$

And the effective coefficients are related with the solution of micro scale problem, i.e.

$$a^H_{ij} = \frac{1}{|Y|}\int_Y \left(a_{ij}(\mathbf{y}) + a_{ik}(\mathbf{y})\frac{\partial\chi^j}{\partial y_k}\right)\,\mathrm{d}\mathbf{y} \tag{2.27}$$

2.5 Hill-Mandel Condition

After introducing the general mathematical concepts about homogenization methods, we move to its application in material modelling. From the perspective of material modelling a Representative Volume Element (RVE) is investigated. This concept coincides with *reference cell* in the previous section from the mathematical perspective. In such an RVE, size is not what we are concerned with, since the homogenization will lead to $\epsilon \to 0$, meaning the size of RVE going against 0, achieving a homogeneous state. This homogenization process and the underlying reason was elaborated in the previous section 2.1. Hence such RVE is often take as a unit cell, other geometry configurations are explored in [8].

In homogenization of an RVE the key step is to obtain coefficients for the macro scale PDE. For material modelling such coefficients are material parameters, i.e. tangent moduli. The behaviour of RVE should resemble the material in this scale. Therefore the model for micro scale should be able to capture specific features, where the continuum mechanical equilibrium of composites in the micro scale is always established. Such equilibrium can be formulated in various ways. The method utilized in this student project is an energy based description. The connection between micro and macro scale is called Hill-Mandel condition [11]. In a recent paper the relation of Irving-Kirkwood theory and Hill-Mandel homogenization theory is revealed [17].

The Hill-Mandel condition states that the total stress power on the micro scale should be equal to the stress power at relevant point on the macro scale. For small strain, the following equation holds,

$$\langle \boldsymbol{\sigma} \cdot \dot{\boldsymbol{\varepsilon}} \rangle = \langle \boldsymbol{\sigma} \rangle \cdot \langle \dot{\boldsymbol{\varepsilon}} \rangle \,, \tag{2.28}$$

where $\langle \cdot \rangle$ means averaging of the considered variable. In simulations certain boundary conditions are devised to fulfil Hill-Mandel condition automatically.

To solve PDEs for the micro state various issues arise. To input the information from the macro scale, averaged field variable from the macro scale was inserted in the formulation. The boundary condition of micro scale model should also be compatible within the context of homogenization. Various function spaces are devised to fulfil this point. The most common three boundary conditions for RVE simulations are Linear Displacement Boundary Condition (LDBC), Homogeneous Traction Boundary Condition (HTBC), and Periodic Boundary Condition (PBC) [8]. It can be shown that function spaces with these three common boundary conditions satisfy Hill-Mandel condition a priori. For simulations of other kinds such as thermo-mechanical or porous medium, we refer to [16], [26], and [9].

The formulation of Hill-Mandel condition in implementation aspects can be referred in [20], where Lagrangian Multiplier is used to enforce Hill-Mandel condition.

2.6 Basic Concepts of FE^2 Scheme

Embedding a micro scale computation on an RVE into a macro scale computation is the essential idea of FE^2 scheme. Related works concerning micro-macro coupling can be found in [22] [20] [19] [18] [21]. Here we present the idea from [22], which is the basics of FE^2 scheme. Although in a different notation, the following formulation can be well compared with the one presented in section 2.4.

For large strain problem the macro scale problem is,

$$\begin{cases} \operatorname{Div} \overline{\mathbf{P}} + \bar{\gamma} = \mathbf{0} & \text{in } \mathcal{B} \\ \bar{\mathbf{t}} = \bar{\mathbf{t}}_b & \text{on } \partial\mathcal{B}_f \\ \bar{\mathbf{x}} = \bar{\mathbf{x}}_b & \text{on } \partial\mathcal{B}_x \end{cases} \tag{2.29}$$

At the same time we have micro scale problem as follows

$$\begin{cases} \operatorname{Div} \mathbf{P} = \mathbf{0} & \text{in } \mathcal{RVE} \\ \widetilde{\mathbf{w}}^{+} = \widetilde{\mathbf{w}}^{-} & \text{on } \partial\mathcal{RVE} \end{cases} \tag{2.30}$$

where $\overline{(\cdot)}$ stands for macro field averages, and $\widetilde{(\cdot)}$ represents micro field fluctuations. $(\cdot)^{+}$ and $(\cdot)^{-}$ are the corresponding fluctuation pair on the boundary. The relation between micro scale and macro scale is

$$\mathbf{x} = \overline{\mathbf{F}}\mathbf{X} + \widetilde{\mathbf{w}} \text{ in } \mathcal{RVE} \tag{2.31}$$

The effective tangent moduli is derived from the averaged stress and strain,

$$\overline{\mathbf{P}} = \frac{1}{V} \int_{\mathcal{RVE}} \mathbf{P} \, \mathrm{dx}, \quad \overline{\mathbf{F}} = \frac{1}{V} \int_{\mathcal{RVE}} \mathbf{F} \, \mathrm{dx} \tag{2.32}$$

$$\Delta \overline{\mathbf{P}} = \mathbb{C}_{\text{eff}} \Delta \overline{\mathbf{F}} \tag{2.33}$$

The weak form of micro scale problem is

$$G = \int_{\mathcal{RVE}} \delta\widetilde{\mathbf{F}} : \mathbf{P} \; \mathrm{d}\mathbf{x}. \tag{2.34}$$

The linearization of the above weak form can be written as

$$G + \Delta G = G + \int_{\mathcal{RVE}} \delta\widetilde{\mathbf{F}} : \mathbb{C} : \left(\Delta\overline{\mathbf{F}} + \Delta\widetilde{\mathbf{F}}\right) \; \mathrm{d}\mathbf{x} \tag{2.35}$$

Noting $\widetilde{\mathbf{F}} = \nabla\widetilde{\mathbf{w}}$ and taking both (2.34) and (2.35) zero it follows with

$$\int_{\mathcal{RVE}} \delta\nabla\widetilde{\mathbf{w}} : \mathbb{C} : \left(\Delta\overline{\mathbf{F}} + \Delta\nabla\widetilde{\mathbf{w}}\right) \; \mathrm{d}\mathbf{x} = 0 \tag{2.36}$$

After discretization using $\widetilde{\mathbf{w}}^h = \mathbf{Nd}$, $\nabla\widetilde{\mathbf{w}}^h = \mathbf{Bd}$, and $\mathbb{C} : \nabla\widetilde{\mathbf{w}}^h = \mathbf{Ld}$. The corresponding stiffness matrix is denoted as $\mathbf{K}$. Then (2.36) is simplified as

$$\mathbf{K}\Delta\widetilde{\mathbf{d}} = -\mathbf{L}\Delta\overline{\mathbf{F}} \tag{2.37}$$

Inserting this relation into averaged stress (2.32) gives

$$\begin{aligned}
\overline{\mathbf{P}} &= \frac{1}{V}\int_{\mathcal{RVE}} \mathbf{P} \; \mathrm{d}\mathbf{x} \\
&= \frac{1}{V}\int_{\mathcal{RVE}} \mathbb{C} : \left(\Delta\overline{\mathbf{F}} + \Delta\nabla\widetilde{\mathbf{w}}^h\right) \; \mathrm{d}\mathbf{x} \\
&= \frac{1}{V}\int_{\mathcal{RVE}} \mathbb{C} : \Delta\overline{\mathbf{F}} \; \mathrm{d}\mathbf{x} + \frac{1}{V}\mathbf{L}^T\Delta\widetilde{\mathbf{d}}\Delta\overline{\mathbf{F}} \\
&= \frac{1}{V}\int_{\mathcal{RVE}} \mathbb{C} : \Delta\overline{\mathbf{F}} \; \mathrm{d}\mathbf{x} - \frac{1}{V}\mathbf{L}^T\mathbf{K}^{-1}\mathbf{L}\Delta\overline{\mathbf{F}}
\end{aligned} \tag{2.38}$$

This expression gives the end result of the effective tangent moduli

$$\mathbb{C}_{\text{eff}} = \frac{1}{V}\int_{\mathcal{RVE}} \mathbb{C} \; \mathrm{d}\mathbf{x} - \frac{1}{V}\mathbf{L}^T\mathbf{K}^{-1}\mathbf{L} \tag{2.39}$$

(2.29), (2.33), and (2.39) constitute the governing equation of macro scale, which is to be solved in the sense of common FE formulation.

To sum up in the complete FE^2 scheme micro computation and macro computation runs alternatively. When the deformation from the macro scale problem is obtained through (2.29), one can start to solve the micro scale problem with the averaged strain calculated from the macro scale. After solving the fluctuation of the micro scale problem, the current averaged tangent moduli can be determined. In the meanwhile the second term of the effective tangent moduli can be evaluated. Combining the above two leads to a full representation effective tangent moduli, which is plugged into the macro scale problem. Then the solving of a macro scale problem will start again. This is illustrated in the following diagram.

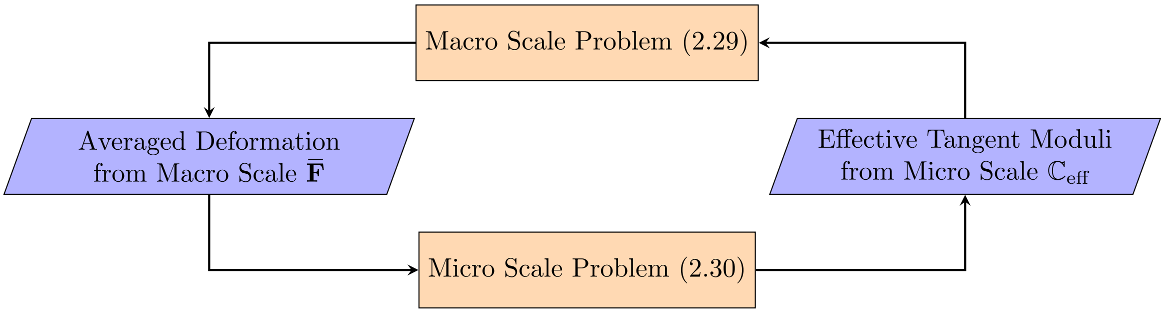
Macro Scale Problem (2.29)
Averaged Deformation from Macro Scale $\overline{\mathbf{F}}$
Effective Tangent Moduli from Micro Scale $\mathbb{C}_{\text{eff}}$
Micro Scale Problem (2.30)

Figure 2.3: FE^2 Scheme

This scheme is extended for multiple fields in the second part of report.

3 Finite Element Framework FEniCS

The FEniCS project was started at University of Chicago and Chalmers University of Technology in 2003. The name "FEniCS" consists "FE", "Finite Element" and "CS", "Computational Software". "ni" makes the name sound like "phenix". It is a collection of packages that make the best of each package to realize mathematical modelling. The main interface language of FEniCS is Python, which is a fast language for prototyping, while the most of code and libraries are implemented in C++, whose efficiency in computing stands out. Various linear algebra backends and its parallelization tools improve its efficiency further. As for modelling, UFL (Unified Form Language) and FFC (FEniCS Form Compiler) provide direct translation of mathematical formulation such as linear form and bilinear form into symbolic code. It can be regarded as the language of modelling, which simplifies the modelling process to a large extent [14].

This part concentrates on the introduction of FEniCS. Firstly the structure and components of FEniCS are presented. Then a basic example is given in order to clarify its efficiency in modelling. At last some key features of FEniCS are discussed, which demonstrates its flexibility in simulations.

3.1 The Structure of FEniCS

The structure of FEniCS is presented in the following diagram. It is shown that the structure is split into three major parts. Each part represents one or several functionalities. For form compiler UFL language is involved. With the help of form compiler (FFC), C++ code of the whole model is obtained. Till this point the mathematical derivation, which is the weak form of a PDE in most cases, is converted into pure symbolic efficient C++ code. The interface between symbolic code and numerical code is called UFC (Unified Form-assembly Code). It works as a fixed interface that achieve efficient assembling of finite elements [29].

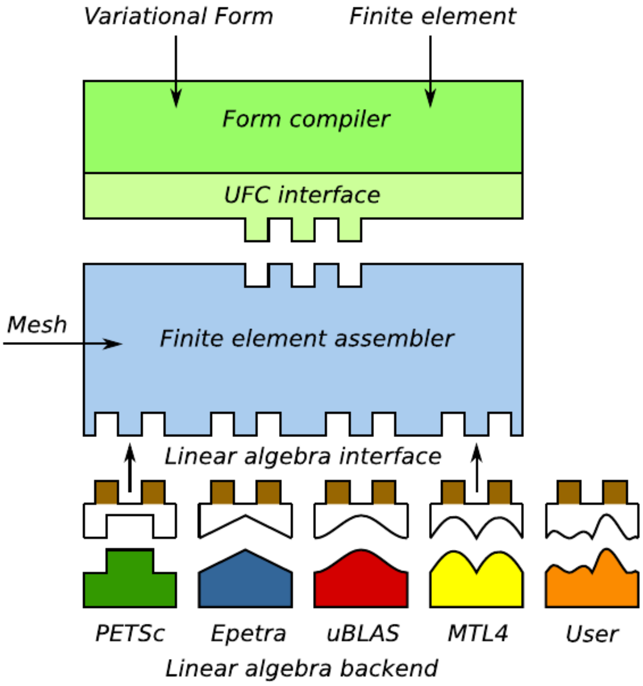

Figure 3.1: Building Blocks of FEniCS [29]

Numerical algorithms enter after the UFC interface. The main package that carries the numerical algorithm is DOLFIN. DOLFIN's task is to wrap functionalities of each components and their interactions. The user interface of DOLFIN is Python.

Other building blocks such as mshr for geometrical modelling and mesh generation, FIAT (FInite element Automatic Tabulator) for automatic finite element generation for arbitrary orders, and linear algebra backends work as add-ons for FEniCS, which provides the possibility of extension. The visualization package in use is

VTK, a substitute of the old one, Viper. The output files could be in various formats. Therefore software like ParaView can be applied in terms of visualization. The full structure of FEniCS can be viewed in the following diagram.

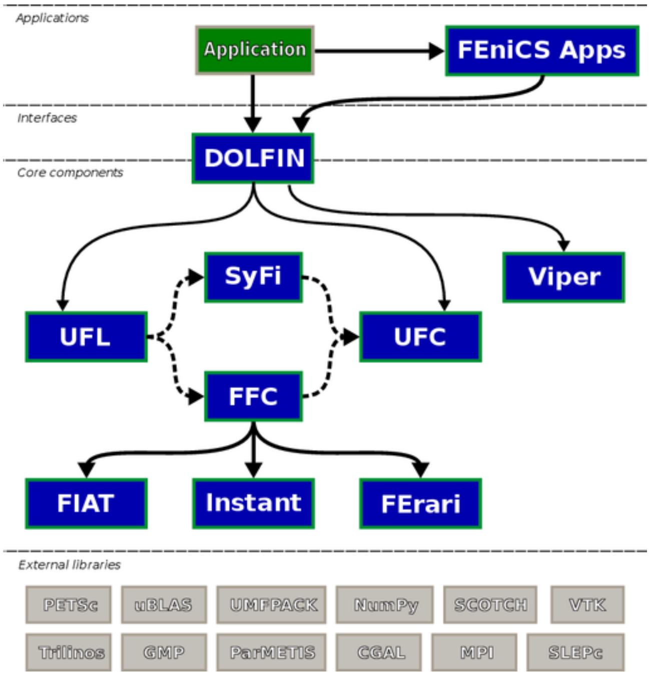

Figure 3.2: Components of FEniCS [29]

3.2 Basic Poisson Problem

An implementation of a basic Poisson problem is presented in the official tutorial of FEniCS. This example is reproduced here to give a quick overview of its usage and present its simplicity in modelling [29].

```
from dolfin import *

# Create mesh and define function space
mesh = UnitSquareMesh(6, 4)
V = FunctionSpace(mesh, ’Lagrange’, 1)

# Define boundary conditions
# Value function for boundary condition x[0]->x, x[1]->y
u0 = Expression(’1 + x[0]*x[0] + 2*x[1]*x[1]’)
# Mark boundary
def u0_boundary(x, on_boundary):
    return on_boundary
# Add Dirichlet boundary condition
bc = DirichletBC(V, u0, u0_boundary)

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Constant(-6.0)
# Bilinear Form
a = inner(nabla_grad(u), nabla_grad(v))*dx
# Linear Form
L = f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u, bc)

# Plot solution
plot(u, interactive = True)

# Dump solution to file in VTK format
file = File(’poisson.pvd’)
file << u
```

The mathematical model of this problem is

$$\begin{cases} -\Delta u(\mathbf{x}) = 0 & \text{in } \Omega \\ u(\mathbf{x}) = u_0 & \text{on } \partial\Omega. \end{cases} \tag{3.1}$$

The corresponding weak form can be written as

$$a(u, v) = L(v), \ \forall v \in \hat{V} \tag{3.2}$$

with the bilinear form and linear form defined as

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\mathbf{x}, \ L(v) = \int_\Omega f v \, \mathrm{d}\mathbf{x}.$$

The sought after function is u in function space $V = \left\{v \in H^1(\Omega) : v = u_0 \text{ on } \partial\Omega\right\}$ and the according $\hat{V}$ is defined as $\hat{V} = \left\{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega\right\}$.

As can be immediately seen in the code, every steps in finite element analysis are well preserved in FEniCS. Moreover, the name of functions and classes are consistent with mathematical language. The modelling always begins with defining its geometry and function spaces. This setting is similar with solving a PDE in weak form, where its domain and function spaces are usually given in condition. Geometry and mesh can also be imported using external files using various formats, such as `.msh` generated by `gmsh`. One thing to notice is that element type should be defined when defining a function space. As shown in the example `'Lagrange'`. The order of element is simply 1.

It follows with defining the boundary condition such as Dirichlet boundary condition. An `Expression` is a class holding the code of creating a mathematical function in the domain. It is extensible with C++ code. Here we use it to define the boundary value. As for imposing this boundary condition, one need to mark the boundary first. Marking other boundary entities such as points, lines, and facets is the same with the provided code structure. `on_boundary` is only a predefined marker that will return `True` when on the boundary.

Then the mathematical model is conveniently expressed with `a = inner(nabla_grad(u), nabla_grad(v))*dx` and `L = f*v*dx`, which is rather straightforward.

Next is the solving step, where one should redefine a new `Function` to hold the solution. Here overriding the above defined `u` is implemented. Post processing such as visualization of the result follows with `plot`. Output of the result is also standard with the listed commands.

The presentation of the basic Poisson problem serves as an introduction to the functionalities of FEniCS. To sum up, the implementation is in accordance with mathematical notations. Pre- and post-processing of the computation can be realized using handy functions supported by FEniCS. More useful features are discussed in the following section.

3.3 Relevant Key Features

When the problem becomes more complicated, there are more advanced and handy tools coming into play. Here we list only a few that appear in the material modelling and multi field problems.

First we explore the function space definition in FEniCS. There are several kinds of `FunctionSpace` in FEniCS. If a vector or tensor is used, `VectorFunctionSpace` or `TensorFunctionSpace` should be defined. More powerful structure is the `MixedFunctionSpace`, which receives a list of `FunctionSpace` and composes them together into a merged one. Due to the general representation of function spaces, arbitrary order of elements and various combinations possible. For very specific function spaces which require periodicity `constrained_type` keyword is set to be a periodic mapping. This mapping is derived from `SubDomain`. Much caution needs to be devoted into the definition of mapping, which can be observed in the corresponding code for material modelling.

```
# Useful properties and variables for FunctionSpace definition
FS = FunctionSpace(mesh=some_mesh, family='Lagrange', degree=2, constrained_domain=
                                               some_defined_periodic_mapping)

# Vector and tensor function spaces
VFS = VectorFunctionSpace(mesh, 'Lagrange', 1)
TFS = TensorFunctionSpace(mesh, 'Hermite', 1)

# Merged function Space
MFS = MixedFunctionSpace([FS, VFS, TFS])
```

When dealing with multi field modelling, not only merging functions but also splitting functions are required, since in the formulation different terms are calculated with different functions. A mixed function is generated with a `MixedFunctionSpace` and the corresponding component functions are obtained through `split` without losing dependency to the merged function. In the problem formulation it is often the case that inner product, outer product, derivative, and integral etc. are built. This is achieved easily with the help of UFL operators. Some of the operators such as `nabla_grad()` and `inner()` are already mentioned above. One strength of FEniCS is that this formulation is not restricted in the vector and matrix, but applicable on tensors of higher order. The index notation representation is also valid in the formulation, namely `i,j,k,l`, obeying Einstein's summation convention. After the derivation a stiffness matrix is assembled with `assemble`. These usages are as follows.

```
# Merged function space
MFS = MixedFunctionSpace([FS, VFS, TFS])

# Merged function and split functions
merged_f = Function(MFS)
(f, vf, tf) = split(merged_f)

# Integral
L = f*v*dx

# Derivative and its alternative
L_derivative = derivative(L, f, df)
L_diff = diff(L, f)

# Index notation
i, j, k, l = indices(4)
Energy = 0.5*F[i,j]*C_i_[i,j,k,l]*F[k,l]*dx

# Stiffness matrix, note that test function and trial function should be initiated first
f_test = TestFunction(FS)
f_trial = TrialFunction(FS)
a = f_test*C*t_trial*dx
K = assemble(a)
```

The last usage is particularly useful in composite modelling, i.e. defining different domains. Subdomains are defined through subclassing of `SubDomain`. Overriding its member method `inside()` gives the condition to distinguish whether it is in a specific subdomain or not. In integral the differentials are set according to subdomain labels.

```
# Inclusion definition
class Inclusion(SubDomain):
    def inside(self, x, on_boundary):
        d = sqrt((x[0] - 0.5)**2 + (x[1] - 0.5)**2)
        return d<0.25 or near(d,0.25)

# Initiate an instance
circle = Inclusion()

# Set cell domains for integral
domains = CellFunction("size_t", mesh)
domains.set_all(0)
# Mark inclusion
circle.mark(domains,1)

# Integrate accordingly
dx = Measure('dx', domain=mesh, subdomain_data=domains)
Pi = psi_m*dx(0) + psi_i*dx(1)
```

Bibliography

[1] Assyr Abdulle. Numerical homogenization methods. *Springer, Encyclopedia of Applied and Computational Mathematics*, 2013.

[2] F. C. Campbell. *Structural Composite Materials.* ASM International, 2010. 00151.

[3] Doina Cioranescu and Patrizia Donato. *An Introduction to Homogenization.* Oxford University Press, Oxford ; New York, 1 edition edition, February 2000.

[4] C. Dascalu, G. Bilbie, and E. K. Agiasofitou. Damage and size effects in elastic solids: A homogenization approach. *International Journal of Solids and Structures*, 45(2):409–430, January 2008.

[5] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science & Business Media, 2009.

[6] F. Feyel and J. L. Chaboche. FE2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3-4):309–330, 2000.

[7] Felix Fritzen and Thomas Böhlke. Reduced basis homogenization of viscoelastic composites. *Composites Science and Technology*, 76:84–91, March 2013.

[8] R Glüge, M Weber, and A Bertram. Comparison of spherical and cubical statistical volume elements with respect to convergence, anisotropy, and localization behavior. *Computational Materials Science*, 63:91–104, 2012.

[9] William G. Gray, Bernhard A. Schrefler, and Francesco Pesavento. The solid phase stress tensor in porous media mechanics and the Hill-Mandel condition. *Journal of the Mechanics and Physics of Solids*, 57(3):539–554, March 2009.

[10] Behrooz Hassani and Ernest Hinton. A review of homogenization and topology optimization i — homogenization theory for media with periodic structure. *Computers & Structures*, 69(6):707–717, 1998.

[11] R. Hill. The essential structure of constitutive laws for metal composites and polycrystals. *Journal of the Mechanics and Physics of Solids*, 15(2):79–95, March 1967.

[12] Scott J Hollister and Noboru Kikuchi. A comparison of homogenization and standard mechanics analyses for periodic porous composites. *Computational Mechanics*, 10(2):73–95, 1992.

[13] S. Jansson, F. A. Leckie, E. T. Onat, and M. P. Ranaweera. Materials with periodic internal structure: Computation based on homogenization and comparison with experiment. Technical report, October 1990.

[14] Robert C Kirby and Anders Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software (TOMS)*, 32(3):417–444, 2006.

[15] Shaofan Li, Roger A. Sauer, and Gang Wang. The Eshelby tensors in a finite spherical domain—part I: theoretical formulations. *Journal of applied mechanics*, 74(4):770–783, 2007.

[16] Kranthi K. Mandadapu, Arkaprabha Sengupta, and Panayiotis Papadopoulos. A homogenization method for thermomechanical continua using extensive physical quantities. *Proc. R. Soc. A*, 468(2142):1696–1715, June 2012.

[17] Brian S. Mercer, Kranthi K. Mandadapu, and Panayiotis Papadopoulos. Novel formulations of microscopic boundary-value problems in continuous multiscale finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 286:268–292, April 2015.

[18] C. Miehe. Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *International Journal for Numerical Methods in Engineering*, 55(11):1285–1322, December 2002.

[19] C. Miehe. Computational micro-to-macro transitions for discretized micro-structures of heterogeneous materials at finite strains based on the minimization of averaged incremental energy. *Computer Methods in Applied Mechanics and Engineering*, 192(5-6):559–591, 2003.

[20] C. Miehe and A. Koch. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. *Archive of Applied Mechanics*, 72(4-5):300–317, July 2002.

[21] C. Miehe, J. Schotte, and M. Lambrecht. Homogenization of inelastic solid materials at finite strains based on incremental minimization principles. Application to the texture analysis of polycrystals. *Journal of the Mechanics and Physics of Solids*, 50(10):2123–2167, October 2002.

[22] C. Miehe, J. Schotte, and J. Schroder. Computational micro-macro transitions and overall moduli in the analysis of polycrystals at Large strains. *Computational Materials Science*, 16(1-4):372–382, December 1999.

[23] C. Miehe, J. Schroder, and J. Schotte. Computational homogenization analysis in finite plasticity - Simulation of texture development in polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering*, 171(3-4):387–418, April 1999.

[24] C. Miehe, D. Vallicotti, and D. Zaeh. Computational structural and material stability analysis in finite electro-elasto-statics of electro-active materials. *International Journal for Numerical Methods in Engineering*, 102(10):1605–1637, June 2015.

[25] Christian Miehe, Daniel Vallicotti, and Stephan Teichtmeister. Homogenization and multiscale stability analysis in finite magneto-electro-elasticity. Application to soft matter EE, ME and MEE composites. *Computer Methods in Applied Mechanics and Engineering*, 300:294–346, March 2016.

[26] I. Ozdemir, W. a. M. Brekelmans, and M. G. D. Geers. Computational homogenization for heat conduction in heterogeneous solids. *International Journal for Numerical Methods in Engineering*, 73(2):185–204, January 2008.

[27] J. Schröder and M.-A. Keip. Two-scale homogenization of electromechanically coupled boundary value problems consistent linearization and applications. *Computational Mechanics*, 50(2):229–244, 2012.

[28] E Weinan, Bjorn Engquist, Xiantao Li, Weiqing Ren, and Eric Vanden-Eijnden. Heterogeneous multiscale methods: a review. *Commun. Comput. Phys*, 2(3):367–450, 2007.

[29] Garth Wells, Kent-Andre Mardal, and Anders Logg. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer, 2012.