# 1 Summary and Outlook

This “Forschungsmodul” concentrates on homogenization method for multi field modelling, where the novel finite element framework FEniCS is investigated. Some key points of homogenization and basic concepts in FEniCS are presented in this repport. A unified derivation of multi field problem is given in part, derivation and implementation. The strength of FEniCS is revealed for problems of multiple fields. It is also convinced that the modelling process in FEniCS is rather straightforward, as it uses the most mathematical language. Due to Python as interface language, the period of prototyping is shortened.

It is also shown that the current module is easy to extend for other problems, as the most classes are rather generic in the formulation. This will lead to difficulty, if one is not familiar with the method and considerations of the original code. In order to alleviate this drawback, docstrings explaining the codes are well preserved and a example manual will guide user better to understand the usage of the code.

As for improvement of the code, the most relevant one is to expand the functionalities of this unit cell module, which will adapt to other numerical scheme. One of such is $\mathrm{FE}^2$. Much work could be done in this direction in implementation. The upper architecture of the problem is a macro scale problem, which will use the parameters calculated from the micro scale, i.e. homogenized problem. Since the parameters vary from element to element, the assembling might be a problem. If one want to make the best of the efficiency of FEniCS, this parameter data should be transformed into a Function object in FEniCS (or Expression object, it would be more efficient if subclassing of Expression is written in C++). This will add an extra layer of data communication. If this subclassing is achieved and linked to the unit cell module and succeed in handling parallelization nature of FEniCS. It would become a powerful tool to investigate composite behaviour in multiple fields. Another consideration is to design a new assembler that is appropriate for macro scale problem. The new assembler is based on the old one. One must then delve into the detailed implementation written in C++. This method is not generic, as with other type of PDEs or many parameters this method can not be applied. Another improvement of the current work might be plasticity or viscosity material model. For plasticity problem, a optimization problem could be formulated in micro scale and accomplish the calculation. For viscosity the time dependency needs to be included in the implementation.