

"Common challenges in large-scale code development"

Abstracts

1. Peter Munch

deal.II: challenges and opportunities of developing an open-source FEM library

deal.II is an open-source general-purpose finite-element C++ library with origins at the University of Heidelberg in 1997. Since then, it has evolved into a mature and worldwide developed and used mathematical software library. It is the basis of applications ranging from traditional application fields like computational solid mechanics, fluid mechanics, biomechanics, and geosciences to more “exotic” fields of research such as quantum and plasma physics. The scalability of deal.II has been shown by solving partial differential equations with trillions of unknowns on 300k processes. For its outstanding contributions to the development and use of mathematical and computational tools and methods for the solution of scientific and engineering problems, deal.II was awarded the prestigious “SIAM/ACM Prize in Computational Science and Engineering 2025”.

In this talk, we will give a short introduction to deal.II and discuss the challenges and opportunities of developing an open-source (finite element) library by scientists spread all over the world.

2. Luca Ferranti

Overview of Nordic-RSE

This talk will give an overview of Nordic-RSE, a non-profit organization with the purpose to bring together, support, and promote Research Software Engineers and their activities in the Nordics. RSE as a role is relatively new and often not well captured in academic career paths. As such, several researchers often work "de facto" as RSEs, without knowing of other peers being in the same boat. The purpose of the organization is to bring together everyone interested in RSE and support their growth and career progression. The talk will give an overview of Nordic-RSE story and activities and how everyone can get involved.

3. Jonas Lindemann

From Code to Classroom: Practical Applications of Large Language Models in Software Development

This talk presents practical experiences of using Large Language Models (LLMs) in software development and teaching. Examples include C++ and Python programming, MATLAB-to-Python translation, and creating interactive visualisation tools, as well as generating documentation and improving lecture material. The focus is on concrete use cases, lessons learned, and how LLMs are reshaping workflows in research software and education.