USE OF THE DUNE ENVIRONMENT FOR THE DEVELOPMENT OF A VAPOR EXPLOSION CALCULATION TOOL

Magali ZABIEGO – Christophe FOCHESATO

Severe Accident Physics and Modeling Laboratory
CEA (French Alternative Energies and Atomic Energy Commission)
Cadarache
FRANCE
• **Generation IV international forum** → specifications:

  - improve safety
  - improve nuclear non-proliferation
  - minimize the waste production (nuclear fuel recycling)
  - optimize the use of natural resources
  - decrease the reactor building and operating costs

  Among other things, take into account, from the very conception of the reactor, the possible occurrence of a severe accident

  Severe accident sequence = Series of events leading to the ruin of the reactor core

• **CEA (+ partners) → conception of a Sodium-cooled Fast neutron Reactor (SFR)**

  designed to match the Gen IV forum specifications

• **Severe accident study → Core degradation**

  - Local heat-up (various initiators considered)
  - Material melting → corium (molten core) formation (~3000°C)
  - Corium relocation into the reactor lower plenum

  - Corium jet in contact with « cold » sodium (FCI)
    → violent interaction (vapor explosion) that may threaten the reactor structures
    → fine corium debris generation, sedimentation on the core catcher: cooling? recriticality?

  ⇒ To be evaluated for the reactor conception
VAPOR EXPLOSION

- Example of vapor explosion: molten material representing corium injected into water

  ![Molten oxide droplet surrounded by a vapor film](image1)
  ![Vapor film destabilisation, droplet explosion](image2)

  L.S. Nelson experiments, SANDIA ~1980
  1-10g droplets of molten alloy injected in water

- SFR accident scenarios → hot liquid corium (~3000°C) flow into more volatile cold liquid sodium (~500°C)

  Process similar to corium-water interaction (deeply studied for water-cooled reactors):
  - heavy corium fragmentation
  - coolant heating and vaporisation
  - vapor explosion

  ... but different:
  - sodium thermal behaviour
  - interaction time scale
  - debris size

Corium-sodium interaction
Physics not entirely understood → knowledge needs to be improved:
- dedicated experimental programs
  Data for closure laws, model development and validation
- development of a detailed computer code
  Storage of knowledge
Corium-sodium interaction → development of a new computer code

We need to be able to:
- locate the corium in the system at each time-step and have access to its physical state
- follow the vapor production and the pressure build-up within the system
- calculate the pressure wave expansion

Physics to model

- Mass exchanges
  - corium fragmentation \( \rightarrow \) coarse particles \( \rightarrow \) fine particles
  - corium solidification
  - sodium bubble fragmentation + coalescence
  - sodium vaporisation/ condensation

- Heat exchanges
  - internal conduction within the sodium bulk
  - convection / conduction / radiation between corium and sodium for the different flow regimes

- Momentum exchanges
  - phase and wall frictions, virtual mass, solid pressure
• Development of the \textbf{SCONE}** software

\begin{itemize}
  \item Main features:
  \begin{itemize}
    \item Fast transient phenomenon: $\sim$ms $\rightarrow$ $\sim$s
    \item Multi-materials: sodium and corium
    \item Multi-phases: liquid sodium and vapor, continuous and dispersed corium
  \end{itemize}
\end{itemize}

• First elements of reflection:

\begin{itemize}
  \item Mass, momentum and energy conservation equations written for each phase (phase volume fractions)
    \begin{itemize}
      \item liquid sodium $\rightarrow$ Eulerian field
      \item sodium vapor $\rightarrow$ Eulerian field
      \item dispersed corium $\rightarrow$ Lagrangian or Eulerian with a multifluid description \ldots under investigation
      \item continuous corium $\rightarrow$ Eulerian with an interface tracking method \ldots under investigation
    \end{itemize}
  \item Numerical scheme based on the \textbf{ICE}** scheme $\rightarrow$ structured meshing $=$ first step
    $\rightarrow$ will evolve towards unstructured meshings to represent complex geometries
  \item Space discretisation
    Mass and energy balance equations discretised using the finite-volume method
    Momentum equations discretised using the finite-difference method
  \item Time discretisation: first order Euler scheme
\end{itemize}

\begin{itemize}
  \item Implicit \textbf{Continuous-fluid Eulerian method}
  \item Harlow, 1968
\end{itemize}

\*Software for \textbf{CO}rium-\textbf{Na} interaction Evaluation
We have chosen to **reduce the effort on numerials** by using an open source platform (under LGPL or LGPL-like licence)
- pre-selection of **available tools** according to criteria established by our team
- study 2 of them in more details
  - Only DUNE remains: more tests to perform

**Step by step approach:**
- The first tests started in 2014, we followed the version evolutions (from 2.3.0 to 2.4)
- We had a look at the DuMu\(^x\) module → too application-dependent for our needs
- The US-NRC water/steam tables were implemented (~ defining a new material)
- The dedicated DUNE-FCI module was created
- We played with the PDElab:
  - we modified the cgstokes test case:
    - Non-stationary term added (→ time loop, definition of the initial state)
    - Gravity added
    - New boundary class to distinguish velocity and pressure boundary conditions (driving force: pressure gradient)
- we tried to implement our **ICE-based scheme**
  
  - P, T, e and ρ located at the mesh centers
  
  - Normal velocities at the mesh faces → delicate with PDELab, too finite-element oriented
    
    ⇒ We decided to rely on the basic components of DUNE (DUNE-GRID) to build our own library for our ICE scheme (an interface such as PDElab will be needed)

**Work underway**

Implementation of our ICE scheme with DUNE-GRID for a very simple case:
- model the system with the Euler equations
- discretize the equations using our ICE-based scheme
- already think about the data structure → convenient access to data to code the physics and use the future code

**What keeps us busy these days**: attach the normal velocities at the mesh faces
  
  ⇨ establish the connectivity between the cell face global numbering and the cell numbering in DUNE-GRID

**On going…**