Derivation and implementation in OpenFOAM of a point-kinetics model for Molten Salt Reactors

Arnaldo Samuele Mattioli,* Carlo Fiorina,† Stefano Lorenzi ,* Antonio Cammi*

*Politecnico di Milano, Department of Energy, Nuclear Engineering Division, Via La Masa, 34, I-20156 Milan, Italy
†EPFL, Laboratory for Reactor Physics and System Behaviour, Route Cantonale, 1015-Lausanne, Switzerland,
carlo.fiorina@epfl.ch

INTRODUCTION

In the last couple of decades, the nuclear community has witnessed a renewed interest in Molten Salt Reactors (MSRs) [1]. MSRs are typically claimed to overcome the drawbacks of current LWR technology thanks to their ambient operating pressure, higher efficiency, as well as several positive features associated to a liquid fuel such as: avoiding by design core melting accidents; the possibility of extracting during operation the gaseous and volatile fission products, thus decoupling radioactive and power sources; the possibility of an online reprocessing; a wide flexibility in terms of fuel composition and fuel cycle. However, this feature of a liquid fuel is also a source of concern in terms of licensing, proliferation, and technology readiness. A specific aspect in this sense is the unsuitability of most legacy nuclear safety codes to deal with a circulating fuel.

Several efforts have been dedicated to extending legacy codes to the analysis of MSRs, as well as for developing new dedicated codes (see for instance Ref. [2]). Among various approaches, several authors have opted for the use of the open-source finite-volume OpenFOAM library (see for instance Refs. [3, 4]). This choice is favoured by the availability in OpenFOAM of state-of-the-art CFD solvers, combined with the ease of code modification, which in turns allows quickly implementing and tightly coupling among them the additional equations for neutron diffusion and precursor transport.

As regards to the modelling of neutron kinetics in OpenFOAM, most of the efforts have been directed towards the development of spatial diffusion or SP3 models [3, 4, 5]. This is easily justified by the fact that the computational bottleneck is often represented by the CFD simulation, leading to only marginal computational gains associated with the use of point kinetics. However, in the authors' opinion, a point-kinetics model has the merit of requiring a very limited set of input data compared to spatial kinetics solvers, which can streamline application of a tool and make it accessible to a wider community of users. In addition, it can significantly speed up computations in cases where pressure and velocity do not necessarily need to be solved, such as in reactivity-initiated transients.

This paper presents the rigorous derivation of a point-kinetics model that is suitable for coupled CFD-neutronics tools, as well as its implementation based on OpenFOAM. The proposed model employs a lumped approach for the reactor power while allowing for a representation of delayed neutron precursors both in time and in space. A spatial representation of precursors is in fact necessary for reproducing important phenomena associated with the precursor drift and its impact on the neutron balance [6]. This is especially relevant for systems characterised by cavity or pool geometries, such as the Molten Salt Fast Reactor (MSFR) [7] where mixing and re-circulation effects prevent the use of simplified hypotheses in terms of passage and recirculation time.

Derivation of the point-kinetics model

As mentioned, the objective of this work is to derive and implement in OpenFOAM a point-kinetic model that allows for a spatial representation of delayed neutron precursors and their evolution over time. To this purpose, one can start for instance from the one-group diffusion equations for liquid-fuel nuclear systems:

$$\begin{cases} \frac{1}{v}\frac{\partial\phi(r,t)}{\partial t} = (\nabla\cdot D\nabla - \Sigma_R)\phi(r,t) + [(1-\beta)\nu\Sigma_f + \sum_{i=1}^{R}\lambda_i c_i(r,t) \\ \frac{\partial c_i(r,t)}{\partial t} + u\cdot\nabla c_i(r,t) = -\lambda_i c_i(r,t) + \beta_i\nu\Sigma_f\phi(r,t) \\ +\nabla\cdot D_{i,f}\nabla c_i(r,t) \end{cases} \tag{1}$$

Similar to the classic derivation of the point kinetics equations for solid-fuel systems (see for instance Ref. [8], from which the nomenclature has also been adopted), one can then introduce a generic weighting function $W(r)$ and define an amplitude function T(t)

$$T(t) = \int W(r)\frac{1}{v}\phi(r,t)\,dV \tag{2}$$

and a shape function S(r,t)

$$S(r,t) = \frac{\phi(r,t)}{T(t)} \tag{3}$$

where one can prove (see for instance Ref. [8]) that:

$$\int \frac{1}{v}S(r,t)W(r)\,dV = 1 \tag{4}$$

That is, the normalisation of the time-dependent shape function is such that its weighted integral is a constant.

At this point, one can multiply the first equation in the equations system 1 by the weighting function $W(r)$, integrate over the volume and rearrange to obtain:

$$\begin{cases} \frac{dT(t)}{dt} = \frac{\rho(t)-\beta_{eff}}{\Lambda}T(t) + \sum_{i=1}^{R}\lambda_i C_i(t) \\ \frac{\partial c_i(r,t)}{\partial t} + u\cdot\nabla c_i(r,t) = -\lambda_i c_i(r,t) + \beta_i\nu\Sigma_f\phi(r,t) \\ +\nabla\cdot D_{i,f}\nabla c_i(r,t) \end{cases} \tag{5}$$

where

$$C_i(t) = \int c_i(r,t)W(r) \tag{6}$$

By introducing the reactor power P(t) and by multiplying the Eqs. 5 by $P(0)/T(0)$, one obtains:

$$\begin{cases} \frac{dP(t)}{dt} = \frac{\rho(t)-\beta_{eff}}{\Lambda}P(t) + \sum_{i=1}^{R}\lambda_i \hat{C}_i(t) \\ \frac{\partial \hat{c}_i(r,t)}{\partial t} + u\cdot\nabla \hat{c}_i(r,t) = -\lambda_i \hat{c}_i(r,t) + \beta_i\nu\Sigma_f S(r,t)P(t) \\ +\nabla\cdot D_{i,f}\nabla \hat{c}_i(r,t) \end{cases} \tag{7}$$

where

$$\hat{C}_i(t) = \frac{P(0)}{T(0)} \int W(r) c_i(r,t) = \int W(r) \hat{c}_i(r,t) \tag{8}$$

The precursor equation is then divided by $\int \frac{1}{v} W(r) S(r,t) dV$ and $c^*(r,t)$ is defined as

$$c^*(r,t) = \frac{\hat{c}_i(r,t)}{\int \frac{1}{v} W(r) S(r,t) dV} \tag{9}$$

to obtain:

$$\begin{aligned} &\frac{\partial c_i^*(r,t)}{\partial t} + u \cdot \nabla c_i^*(r,t) \\ &= -\lambda_i c_i^*(r,t) + \frac{\beta_i \nu \Sigma_f S(r,0) P(t)}{\int \frac{1}{v} W(r) S(r,t) dV} + \nabla \cdot D_{i,f} \nabla c_i^*(r,t) \end{aligned} \tag{10}$$

The source term in the precursor equation can be rewritten as:

$$\begin{aligned} &\frac{\beta_i \nu \Sigma_f S(r,0) P(t)}{\int \frac{1}{v} W(r) S(r,t) dV} \\ &= \frac{\beta_i \nu \Sigma_f S(r,0) P(t)}{\int W(r) \nu \Sigma_f S(r,t) dV} \frac{\int W(r) \nu \Sigma_f S(r,t) dV}{\int \frac{1}{v} W(r) S(r,t) dV} \\ &= \frac{\beta_i P(t)}{\Lambda} \frac{\nu \Sigma_f S(r,0)}{\int W(r) \nu \Sigma_f S(r,t) dV} \end{aligned} \tag{11}$$

where the term:

$$S_f(r,t) = \frac{\nu \Sigma_f S(r,0)}{\int W(r) \nu \Sigma_f S(r,t) dV} \tag{12}$$

can be interpreted as a shape function for neutron production. However, since such a normalised shape function in often not available, one may further assume that $\nu \Sigma_f$ is uniform in space to obtain:

$$\frac{\beta_i \nu \Sigma_f S(r,t) P(t)}{\int \frac{1}{v} W(r) S(r,t) dV} = \frac{\beta_i}{\Lambda} P(t) \frac{S(r,0)}{\int W(r) S(r,t) dV} \tag{13}$$

Finally, one multiplies the precursor equation by $\int W(r) S(r,t) dV$ and redefines

$$\tilde{c}_i(r,t) = c^*(r,t) \int W(r) S(r,t) dV \tag{14}$$

to obtain

$$\begin{cases} \frac{dP(t)}{dt} = \frac{\rho(t) - \beta_{eff}}{\Lambda} P(t) + \sum_{i=1}^{R} \lambda_i \hat{C}_i(t) \\ \frac{\partial \tilde{c}_i(r,t)}{\partial t} + u \cdot \nabla \tilde{c}_i(r,t) = -\lambda_i \tilde{c}_i(r,t) + \frac{\beta_i P(t)}{\Lambda} S(0,t) \\ + \nabla \cdot D_{i,f} \nabla \tilde{c}_i(r,t) \end{cases} \tag{15}$$

where

$$\begin{aligned} \hat{C}_i(t) &= \int W(r) \hat{c}_i(r,t) \\ &= \int dV\, W(r) c_i^*(r,t) \int \frac{1}{v} W(r) S(r,t) dV \\ &= \int dV\, W(r) \tilde{c}_i(r,t) \frac{\int \frac{1}{v} W(r) S(r,t) dV}{\int W(r) S(r,t) dV} \\ &= \frac{\int W(r) \tilde{c}_i(r,t) dV}{\int W(r) S(r,t) dV} \end{aligned} \tag{16}$$

Equations 15 and 16 represent a model that can directly be implemented in a code featuring a spatial representation of delayed neutron precursors.

Implementation in OpenFOAM

The set of equations 15 and 16 has been implemented in OpenFOAM starting from the point-kinetics implementation for traditional solid-fuel reactors presented, verified and preliminarily validated in Ref. [9]. The source code, in the form of a C++ class, can be found on GitLab [10] as part of the GeN-Foam multiphysics tool [5]. In this implementation, the assumption is made that both $S(r,t)$ and $W(r)$ are equal to a normalised one-group flux that can be provided by the user.

While the details of the implementation are not reported here for brevity, it is worth pointing out at least that the solution of equations 15 and 16 requires a specific procedure of initialisation. In the case of solid-fuel reactors, the point-kinetics equations can be easily initialised by imposing a target initial power and by solving the steady state equations to find the initial values of the precursors. On the other hand, in the case of Eqs. 15, one has to first solve for the precursors equations to determine the precursors spatial distributions. Based on these distributions, one can then use equation 16 to calculate the precursor source term in the power equation. Finally, the solution of the steady-state equation for power allows deriving that, at equilibrium, the following condition must apply:

$$\rho_{eq} = \beta_{eff} - \frac{\Lambda \sum_{i=1}^{R} \lambda_i \hat{C}_i(0)}{P_0} \tag{17}$$

This means that the overall reactivity during the simulation must be expressed as the sum of the usual terms associated with feedback coefficients and control rods, with the addition ρ_{eq}:

$$\rho(t) = \rho_{feedback} + \rho_{controlRods} + \rho_{eq} \tag{18}$$

The new term ρ_{eq} represents the reactivity loss associated to the circulation of delayed neutron precursors. As an alternative formulation, one can substitute Eq. 18 into the power equation and redefine $\rho(t) = \rho_{feedback} + \rho_{controlRods}$ to obtain:

$$\frac{dP(t)}{dt} = \frac{\rho(t) - \beta_{circ}}{\Lambda} P(t) + \sum_{i=1}^{R} \lambda_i \hat{C}_i(t) \tag{19}$$

where the new term

$$\beta_{circ} = \frac{\Lambda \sum_{i=1}^{R} \lambda_i \hat{C}_i(t)(0)}{P_0} \tag{20}$$

represents the effective (reduced) β that is determined by the circulation of the delayed neutron precursors.

RESULTS

The developed point-kinetic solver has been integrated in the GeN-Foam multi-physics code [5] and applied to the analysis of the Molten Salt Fast Reactor (MSFR) [7]. For simplicity, the 2-D axial-symmetric geometry and mesh reported in Fig. 1 have been employed. Details about geometry and salt physical properties can be found in Ref. [11], where one can also find a verification of the solver based on a comparison between a simple 1-D case and a steady-state analytical solution.

A comparison has been carried out between the new point-kinetic solver and a 6-group neutron diffusion model (made

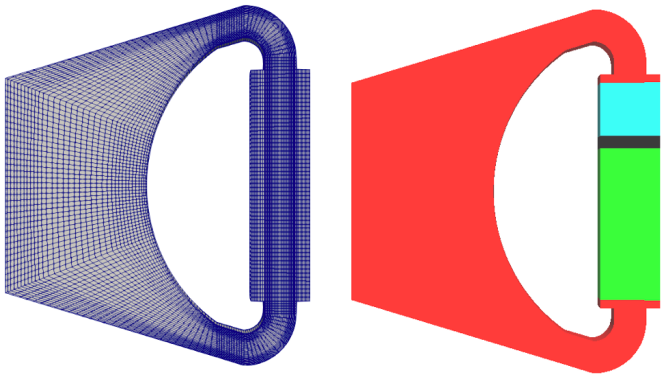

Fig. 1. Mesh and geometry of the MSFR: in red the regions only occupied by the molten salt; in light blue the region occupied by the pump (simulated as a momentum source); in green the region occupied by the heat exchanger (simulated as a constant-temperature heat sink)

available on GitLab [12]), based on the standard diffusion solver of GeN-Foam [13]. The solution of the diffusion model has also been used as shape function for the point-kinetics model.

As a first step, the capability of the point-kinetics solver to evaluate β_{circ} has been tested. To this purpose, β_{circ} has been evaluated in the diffusion case by comparing the k_{eff} with and without delayed neutron precursors. The result is 88.9 pcm and 93.0 pcm for the diffusion and point-kinetic models, respectively. The slight difference between them can be ascribed to the multi-group vs the one-group approximation for neutron energies. In fact, a similar difference is also observed for the static β, which is equal to 274.1 pcm and 285.3 pcm in the diffusion and point-kinetic models, respectively. We tend instead to exclude implementation mistakes since: the base point-kinetics solver has been carefully verified in Ref. [9]; modifications for MSRs are limited and the correct implementation has been preliminarily verified against analytic results in Ref. [11]. Of course, additional verification and validation will be the subject of future investigations for the authors.

Two transients have then been simulated, namely: a 53.4 pcm step reactivity insertion (corresponding to 0.6 $ in the diffusion case), and a 10 seconds exponential reduction of the pump momentum source with a time constant of 2 seconds leading to the fuel velocity decrease shown in Fig. 2. The power evolution in the two aforementioned cases are reported in Figs. 3 and 4, respectively.

Nearly identical solutions are observed in the case of the step-wise reactivity insertion. The overestimation in the power evolution of the diffusion model can be explained with the different reactivity variation normalised to the β_{circ}. In particular, the 53.4 pcm step reactivity insertion is equal to 0.600 $ for the diffusion-based case and 0.574 $ for the point kinetics one.

More visible differences are instead observed for the flow reduction case. However, these differences are consistent with the different prediction of β_{circ} in the two models. A lower β_{circ} in the diffusion case implies a higher reactivity loss due to the circulation of the precursors, which in turns implies

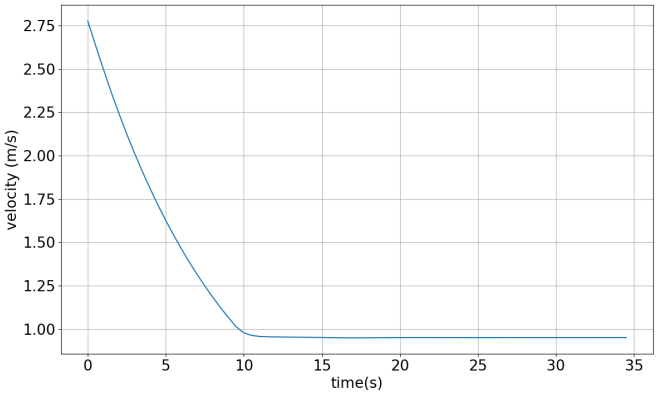

Fig. 2. Velocity evolution in the heat exchanger in case of an exponential reduction of the momentum source in the pump

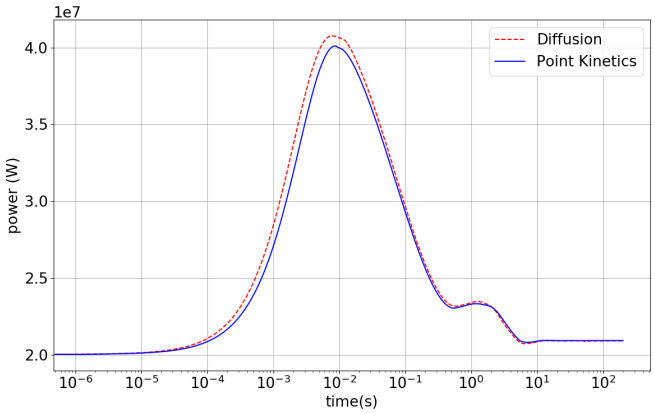

Fig. 3. Power evolution predicted by the point-kinetics and diffusion models for a step-wise reactivity insertion of 53.4 pcm

a higher reactivity insertion during a loss-of-flow transient, where the β_{circ} tends to approach the value of β_{eff} for a static fuel.

CONCLUSIONS

In the present work, a point-kinetics model has been derived for MSRs that can be used in multi-physics codes featuring a spatial representation of the delayed neutron precursors. The model allows simulating the transient behaviour of MSRs by employing standard point-kinetic parameters such as β_{eff}, Λ, and the reactivity coefficients. The only additional input compared to a traditional point kinetic model is a shape function for neutron production. However, this requirement can be relaxed by assuming a constant fission cross section and by employing the one-group flux as shape function.

The proposed model has been implemented in OpenFOAM; integrated into the GeN-Foam multi-physics code (available on GitLab [10]); and verified via a comparison against the standard diffusion solver of GeN-Foam. The comparison has provided expected results, with relatively small discrepancies that are consistent with slightly different pre-

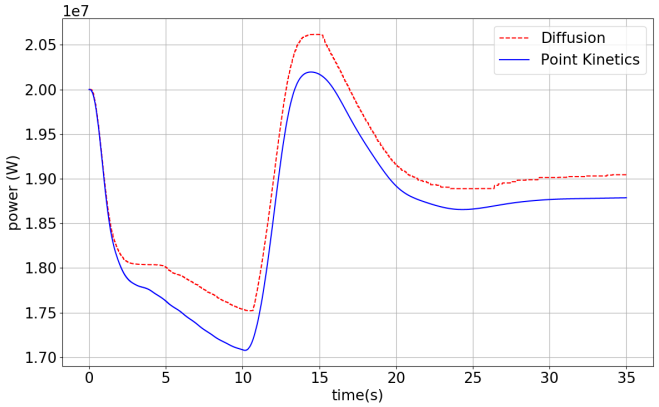

Fig. 4. Power evolution in the case of an exponential reduction of the momentum source in the pump

dictions of the reactivity associated with delayed neutrons. Additional work is required to further verify and validate the solver.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Open-source nuclear codes for reactor analysis (ONCORE) initiative[14]. ONCORE is an IAEA-facilitated international collaboration framework for the development and application of open-source multi-physics simulation tools to support research, education and training for the analysis of nuclear reactors. The work is partially motivated and supported by the endeavors under the initiative.

REFERENCES

1. T. J. DOLAN, *Molten salt reactors and thorium energy*, Woodhead Publishing (2017).
2. D. WOOTEN and J. J. POWERS, "A review of molten salt reactor kinetics models," *Nuclear Science and Engineering*, **191**, *3*, 203–230 (2018).
3. M. AUFIERO, A. CAMMI, O. GEOFFROY, M. LOSA, L. LUZZI, M. E. RICOTTI, and H. ROUCH, "Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis," *Chemical Engineering Science*, **111**, 390–401 (2014).
4. E. CERVI, S. LORENZI, A. CAMMI, and L. LUZZI, "Development of an SP3 neutron transport solver for the analysis of the Molten Salt Fast Reactor," *Nuclear Engineering and Design*, **346**, 209–219 (2019).
5. C. FIORINA, I. CLIFFORD, M. AUFIERO, and K. MIKITYUK, "GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors," *Nuclear Engineering and Design*, **294**, 24–37 (2015).
6. A. CAMMI, C. FIORINA, C. GUERRIERI, and L. LUZZI, "Dimensional effects in the modelling of MSR dynamics: moving on from simplified schemes of analysis to a multi-physics modelling approach," *Nuclear Engineering and Design*, **246**, 12–26 (2012).
7. M. BROVCHENKO, D. HEUER, E. MERLE-LUCOTTE, M. ALLIBERT, V. GHETTA, A. LAUREAU, and P. RUBIOLO, "Design-related studies for the preliminary safety assessment of the molten salt fast reactor," *Nuclear Science and Engineering*, **175**, *3*, 329–339 (2013).
8. A. F. HENRY ET AL., *Nuclear-reactor analysis*, vol. 4, MIT press Cambridge, Massachusetts (1975).
9. S. RADMAN, "A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context, PhD Thesis," Tech. rep., EPFL, Switzerland (2021).
10. EPFL, "GeN-Foam code, available at https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/develop, accessed 01/05/2021," (2021).
11. A. S. MATTIOLI, "Derivation and implementation of a point kinetics model for Molten Salt Reactors in GeN-Foam, MSc Thesis," Tech. rep., Politecnico di Milano, Italy (2021).
12. EPFL, "GeN-Foam Tutorials, available at https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/develop/Tutorials/, accessed 01/05/2021," (2021).
13. C. FIORINA, N. KERKAR, K. MIKITYUK, P. RUBIOLO, and A. PAUTZ, "Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform," *Annals of Nuclear Energy*, **96**, 212–222 (2016).
14. C. FIORINA, P. SHRIWISE, A. DUFRESNE, J. RAGUSA, K. IVANOV, T. VALENTINE, B. LINDLEY, S. KELM, E. SHWAGERAUS, S. MONTI, C. BATRA, A. PAUTZ, S. LORENZI, P. RUBIOLO, I. CLIFFORD, and B. DECHENAUX, "An initiative for the development and application of open-source multi-physics simulation in support of rd and et in nuclear science and technology," *EPJ Web of Conferences*, **247**, 02040 (2021).