Analytical solutions to a coupled fluid dynamics and neutron transport problem with application to GeN-Foam verification[☆]

Rodrigo G.G. de Oliveira[*], Konstantin Mikityuk

École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland
Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

Code verification is an important part of software development process that asserts code correctness. In this work, analytical solutions are used in order to develop a simple multiphysics problem involving coupled fluid dynamics and neutron transport to be used for verification of GeN-Foam. Results show that the analytical solutions are successfully recovered from numerical simulation, with relative errors of the order of less than 0.01%. We conclude that the fluid dynamics and neutron transport solvers of GeN-Foam are working correctly in regard to the limited functionality explored in this test case. Test coverage should be expanded to include additional models, such as turbulence, cross-section parametrization and delayed neutron precursors transport.

Keywords: Verification, Multiphysics, Analytical solution, Method of manufactured solutions

1. Introduction

GeN-Foam is a relatively new multiphysics code [1] using the OpenFOAM library for finite volume method [2] for modelling nuclear reactors. Being such a recent code, very little verification has been done [1, 3, 4] and the ones done were limited to code-to-code benchmarks, which cannot assert correctness of the code, making it a weak form of verification [5]. It is important that rigorous verification efforts start and procedures are implemented to allow a sustainable development of the code. We recognize that other projects might be in a similar position, where a new code is under development and verification will be needed to build up a reliable piece of software. This study targets codes at a stage of development similar to GeN-Foam, where the code is apparently working and returning reasonable outputs, but formal evidence of correctness is lacking.

The body of literature in scientific code verification is fairly rich, with particularly notable works done by Knupp, Oberkampf, Roache and Salari [6, 7, 8]. Although the literature has a

[☆]All files are available at `https://github.com/deOliveira-R/recirculatingCavity`.

[*]Corresponding author
Email address: `rodrigo.de-oliveira@psi.ch` (Rodrigo G.G. de Oliveira)

major focus on computational fluid dynamics, mostly done by the aerospace industry, there are examples available of verification done for radiation transport codes [9, 10, 11].

As a multiphysics code modelling different phenomena, GeN-Foam is harder to verify than a usual code that solves only radiation transport or fluid mechanics separately. As a first approach, compromises in test coverage were made in order to build up evidence that essential functions of the code are working correctly, namely the fluid dynamics and neutron diffusion solvers. In this work, the intention is to verify GeN-Foam in a simplified homogenous system, targeting reactors such as the Molten Salt Fast Reactor, undergoing studies by the SAMOFAR project [12].

Code verification can be divided into 2 recognized branches: numerical algorithm verification and software quality assurance (SQA) [6, page 47]. Numerical algorithm verification, produces evidence that numerical algorithms in the code are implemented correctly and functioning as intended, which is the main objective of this study. A secondary objective is to transform the test case into a regression test, which is part of SQA, allowing continuous testing of the code during further developments.

2. Methodology and specifications

2.1. Equations to solve

Equations 1, 2, and 3, collectively known as the Navier-Stokes equations, represent the conservation of mass, momentum and energy for an incompressible flow.

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} - \mu\nabla^2\mathbf{u} = -\nabla P + \mathbf{F} \tag{2}$$

$$\rho c_p(\mathbf{u} \cdot \nabla)T - k\nabla^2 T = \gamma\phi \tag{3}$$

The following equation (4) is the one-group neutron diffusion equation expressed in terms of the reactor buckling.

$$\nabla^2\phi + B^2\phi = 0 \tag{4}$$

These 4 equations represent the multiphysics problem that is to be verified by this work. The usual nomenclature has been used in all equations, but a detailed nomenclature definition can be found at the end of the paper.

In equation 2 a forcing term $\mathbf{F}$ was added on the right-hand side, making the equation a slightly modified version of the usual one. It is typically hard to satisfy non-trivial analytical solutions in the usual equation, therefore a simple manufactured solution was used.

The method of manufactured solutions for code verification circumvents the need for an analytical solution to satisfy the exact equations, proposing instead the use of a modified equation to satisfy the analytical solution [6, 7, 8]. In order to use this method for code verification, it is necessary that the code allows an arbitrary forcing term to be inserted into the equations. This capability is present in all standard OpenFOAM solvers as "fvOptions" and has been implemented in the GeN-Foam fluid mechanics solver during this work. Additional information on how this functionality works can be found in the GeN-Foam test case files at the GitHub repository provided and in OpenFOAM tutorials.

2.2. Solution domain specification

The proposed case is a square domain centred at point (0,0), with dimensions of π meters by π meters, as shown in figure 1.

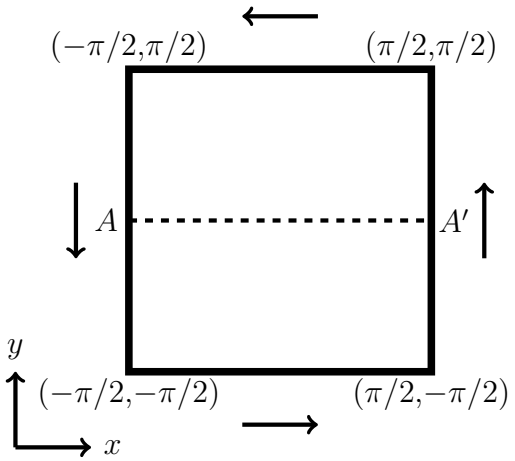

Figure 1: Solution domain

This is a very simple domain, but such simplicity is desirable for verification activities. It reduces the complexities introduced by the mesh into the verification procedure, and allows systematic refinement of the mesh.

For later reference, it is convenient to define the point at the left-bottom corner of the domain as follows:

$$(x_{min}, y_{min}) = -\pi/2, -\pi/2 \tag{5}$$

Care was taken in selecting a domain that is centered on (0,0) instead of placing the (x_{min}, y_{min}) point at (0,0). If the second option had been taken, a round-off error would be introduced because the code can compute $\sin(0)$ exactly, but not $\sin(\pi)$.

2.3. Analytical solutions

In this section, we present the analytical solutions that we would like to recover from the numerical experiments. In order to simplify the analytical solutions, to be presented further on, α and β parameters are defined as follows:

$$\begin{aligned} \alpha(x) &= \frac{\pi}{a}(x - x_{\min}) \\ \beta(y) &= \frac{\pi}{b}(y - y_{\min}) \end{aligned} \tag{6}$$

Where a and b are the dimensions of the domain on the x and y directions. Written in this manner, the problem becomes invariant to stretching (variations of a or b) or translation (variations of $x_{\min}$ or $y_{\min}$) in any direction. For the particular case proposed, $a = b = \pi$ m, resulting in a immediate simplification of equations 6.

Below, we can see the set of analytical solutions imposed for velocity, pressure, temperature and flux, parametrized by α and β previously defined. In equation 8, P_c represents the pressure at the centre of the cavity, which is the minimum pressure and also the reference one. Similarly, T_b in equation 9 represents the constant temperature at the boundaries of the domain, which will be specified later on, while ΔT_c represents an amplitude above this baseline at the centre, which depends on simulation parameters to be discussed. Equation 10 is the flux in the domain, and ϕ_c is flux in the centre, or an amplitude above a baseline along which flux is taken as 0.

$$\begin{aligned} u_x(x, y) &= \sin(\alpha)\cos(\beta) \\ u_y(x, y) &= -\cos(\alpha)\sin(\beta) \end{aligned} \tag{7}$$

$$P(x, y) = P_c + \rho\frac{\cos^2(\alpha) + \cos^2(\beta)}{2} \tag{8}$$

$$T(x, y) = T_b + \Delta T_c \sin(\alpha)\sin(\beta) \tag{9}$$

$$\phi(x, y) = \phi_c \sin(\alpha)\sin(\beta) \tag{10}$$

As previously mentioned, the set of equations is arbitrary. It does not have to be a physical problem in any sense. The only requirement is for **solutions to be continuous**. Another desirable, but not strictly necessary property is for **solutions to be infinitely differentiable**, avoiding the disappearance of any term in equations 1-4 due to differentiation.

By inserting the analytical solutions 7 and 8 into equation 2, the forcing term $\mathbf{F}$ in equation 2 is derived as:

$$\mathbf{F} = 2\mu\mathbf{u} \tag{11}$$

Equations were purposely chosen to give a very simple arbitrary source, contrary to the usual sources generated by the method of manufactured solutions, containing many terms. This was recognized as a good initial approach that allowed testing the arbitrary source functionality just implemented, which requires coding in C++ and is a relatively advanced functionality in OpenFOAM. In addition, it is simple enough to allow derivation by hand, allowing development and verification of a symbolic mathmatics script to derive the usual extensive forcing terms in the future.

2.4. Boundary conditions

Evaluating equation 7 at the domain boundaries, it is possible to specify the velocity boundary conditions as shown in equations 12. Each one is a non-uniform Dirichlet boundary condition that is relatively complex to implement in OpenFOAM, requiring the use of the "codedFixedValue" boundary condition. To better understand how this particular boundary condition is coded, the reader is again referred to the GeN-Foam test case files at the provided GitHub repository.

From the boundary conditions for velocity, it is possible to infer qualitatively that, in this problem, the fluid flows in a counter-clockwise pattern with velocity reducing to 0 at the corners of the domain.

$$\begin{aligned} u_x(x, -\pi/2) &= \sin(\alpha) \\ u_x(x, \pi/2) &= -\sin(\alpha) \\ u_y(-\pi/2, y) &= -\sin(\beta) \\ u_y(\pi/2, y) &= \sin(\beta) \end{aligned} \tag{12}$$

Since the problem is incompressible, the absolute pressure of the system is not important, a reference pressure of 10^5 Pa was given since the code requires some value. What actually matters is the pressure difference created by dynamic forces in the system, commonly known as gauge pressure or, more rigorously, as sealed gauge pressure with the reference pressure as specified. The appropriate boundary condition for this system sets the gradient of pressure normal to the wall to 0. This represents an uniform Neumann boundary condition, which also satisfies equation 8.

For temperature, a uniform Dirichlet boundary condition of $T_b = 1000$ K was used, which results from evaluating equation 9 at the boundaries. This effectively represents an infinite heat sink outside the system with the set temperature.

The neutron flux is set to 0 at the boundaries, also resulting from evaluation of equation 10. While this is a quite unusual boundary condition for the neutron diffusion equation,

it should be emphasized that the main targets of this verification are the solvers, not the boundary conditions available. An albedo boundary condition is implemented in GeN-Foam, and could be used, but this should be done only after the neutron diffusion solver is verified.

All the boundary conditions, with the possible exception of the velocity one, should be readily available in any package for solution of PDEs. The velocity boundary condition might be a bit more challenging due to its non-uniformity.

2.5. Parameters

Table 1 shows the parameters used for the fluid in this problem. All the parameters are arbitrary, but their choice requires some consideration.

Table 1: Fluid properties

Parameter	Value
Density	$1000\ \mathrm{kg \cdot m^{-3}}$
Specific heat capacity	$1\ \mathrm{J \cdot kg^{-1} \cdot K^{-1}}$
Thermal conductivity	$50\ \mathrm{W \cdot m^{-1} \cdot K^{-1}}$
Dynamic viscosity	$50\ \mathrm{Pa \cdot s}$

Dynamic viscosity is high, compared to physically sensible values, in order to force the flow regime to be laminar despite such a large domain. In addition, all the parameters in the table are related in choice in order to keep the advective and diffusive terms in the Navier-Stokes equation in balance so that both are similarly exercised. For example, the choice of specific heat is related to the choice of density and thermal conductivity in order to keep the terms of equation 3 in balance. If this balance is not taken into account in and, for example, a high value is given to specific heat while keeping density and thermal conductivity unchanged, numerical noise in the advection term will become significant. This happens because the advection term of the energy conservation equation is analytically 0, but numerically will be a very small non-zero number (of the order of 10^{-8}). This very small number (numerical noise) will be multiplied by a very high value (density $\cdot$ specific heat capacity) and the term will become significant compared to the diffusion one. This type of carelessness negatively impacts the solution.

The solution of equation 4 requires neutronics parameters, which are also arbitrary to a certain extent. Equation 13 is the geometrical buckling of a 2D square homogenous unreflected reactor with extrapolation distance 0. From the equation, and reminding that $a = b = \pi$ m, the problem has $B_g^2 = 2\ \mathrm{m^{-2}}$.

$$B_{\mathrm{g}}^2 = \left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2 \tag{13}$$

In order for the reactor to be critical, the material buckling, given by the following

equation, has to be equal to the previously calculated geometrical one ($B_g^2 = B_m^2$) [13].

$$B_{\mathrm{m}}^2 = \frac{\frac{\overline{\nu}\Sigma_{\mathrm{f}}}{k_{\mathrm{eff}}} - \Sigma_{\mathrm{a}}}{D} \tag{14}$$

The neutron diffusion coefficient D in equation 14, considering isotropic scattering, is given by:

$$D = \frac{1}{3(\Sigma_{\mathrm{s}} + \Sigma_{\mathrm{a}})} \tag{15}$$

From equations 14 and 15, we generate arbitrary one-group cross-sections considering the following criteria:

1. $k_{\mathrm{eff}} = 1$
2. $D = 2 \cdot 10^{-2}$ m
3. $\Sigma_{\mathrm{a}} = 1$ m^{-1}

Where item 1 is for convenience, 2 is to work with a diffusion coefficient of the same order of magnitude as the mesh size, and 3 is to have $\Sigma_{\mathrm{a}} << \Sigma_{\mathrm{s}}$, which is a condition for the diffusion theory to be valid. The neutronic parameters prescribed by the criteria and the calculated ones are consolidated in table 2.

Table 2: Neutronic properties

Parameter	Value
D	$2 \cdot 10^{-2}$ m
Σ_{a}	1 m^{-1}
Σ_{s}	15.666...m^{-1}
$\overline{\nu}\Sigma_{\mathrm{f}}$	1.04 m^{-1}
γ	1 J $\cdot$ m^{-1}

Inserting the analytical solutions 7, 9, and 10 into equation 3, we get:

$$\gamma\phi_{\mathrm{c}} = 2k\Delta T_{\mathrm{c}} \left(\left(\frac{\pi}{a}\right)^2 + \left(\frac{\pi}{b}\right)^2 \right) \tag{16}$$

For a flux to power conversion ratio (commonly known as kappa-fission) $\gamma = 1$ J $\cdot$ m^{-1}, $k = 50$ W $\cdot$ m^{-1} $\cdot$ K^{-1} and an arbitrary $\Delta T_{\mathrm{c}} = 100$ K, we get a $\phi_{\mathrm{c}} = 10^4$ m^{-2} $\cdot$ s^{-1}. Usually, codes normalize the flux according to reactor power, as is the case of GeN-Foam. To calculate the required power, first it is necessary to integrate the flux $\phi(x, y)$ given by equation 10 over the domain, getting the integral flux Φ. It is important to mention that although the

domain is conceptually 2D, all parameter were given with 3D units for better understanding. However, this implies that in order to achieve strict unit matching the integration has to cover 3 dimensions, where the third dimension over a pseudo-axis z measures 1 m, being numerically irrelevant.

$$\Phi = \int_0^1 \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \phi_c \sin(\alpha) \sin(\beta) dx dy dz = 4\phi_c \tag{17}$$

If $\phi_c = 10^4\ \mathrm{m}^{-2} \cdot \mathrm{s}^{-1}$, as previously mentioned, then $\Phi = 4 \cdot 10^4\ \mathrm{m} \cdot \mathrm{s}^{-1}$. Considering $\gamma = 1\ \mathrm{J} \cdot \mathrm{m}^{-1}$ once again, power will be equal to $\gamma\Phi = 4 \cdot 10^4$ W.

2.6. Post processing specification

Field data for velocity, pressure, temperature and flux are shown in figures 2-5 and used as a qualitative demonstration of the shape of the fields. The fields for the analytical solutions will not be shown because there is no visual difference from the numerical solution fields.

The absolute error field is calculated subtracting the numerical solution field from the analytical solution field taking the absolute value as $|r_n - r_{n,\mathrm{ref}}|$, where r_n is the numerical value of a response in cell n, and $r_{n,\mathrm{ref}}$ is the analytical value of the response in the same cell. The error fields will not be shown for brevity.

Plots over line AA' shown in figure 1 for all numerical, analytical and their respective relative error ($|r_n - r_{n,\mathrm{ref}}| / r_{n,\mathrm{ref}} \cdot 100\%$) will be shown, allowing a quantitative evaluation of the results.

In addition, L_1 and L_2 norms are calculated as shown in equations 18 and 19, where ω_n is the cell volume and Ω is the domain volume.

$$\|r - r_{\mathrm{ref}}\|_1 = \frac{1}{\Omega} \sum_{n=1}^{N} \omega_n |r_n - r_{n,\mathrm{ref}}| \tag{18}$$

$$\|r - r_{\mathrm{ref}}\|_2 = \left(\frac{1}{\Omega} \sum_{n=1}^{N} \omega_n |r_n - r_{n,\mathrm{ref}}|^2 \right)^{\frac{1}{2}} \tag{19}$$

As can be seen, the L_1 norm represents the mean volume-weighted absolute error, and the L_2 norm the root-mean-square of the volume-weighted absolute error. For the particular case of uniform meshes, as used in this study, the equations for the norms degenerate into simple averages over the cells. Nonetheless, the equations are coded as presented, accounting for possible future developments using a non-uniform grid. These norms are used for a sensitivity study of the mesh and to evaluate mesh convergence.

3. Results and discussion

3.1. Numerical solutions

On figure 2 we can see that the flux has the expected behaviour of a cosine shape, showing a peak at the centre and 0 at the border. The shape of the flux is not the typical "chopped cosine" since no extrapolated distance was given to the case. The relative error is very low (below 0.01%), and as we can see, the numerical solution is indistinguishable from the analytical one. The effective multiplication factor was also found with very high accuracy, showing an overprediction of 0.3 pcm for the reference mesh.

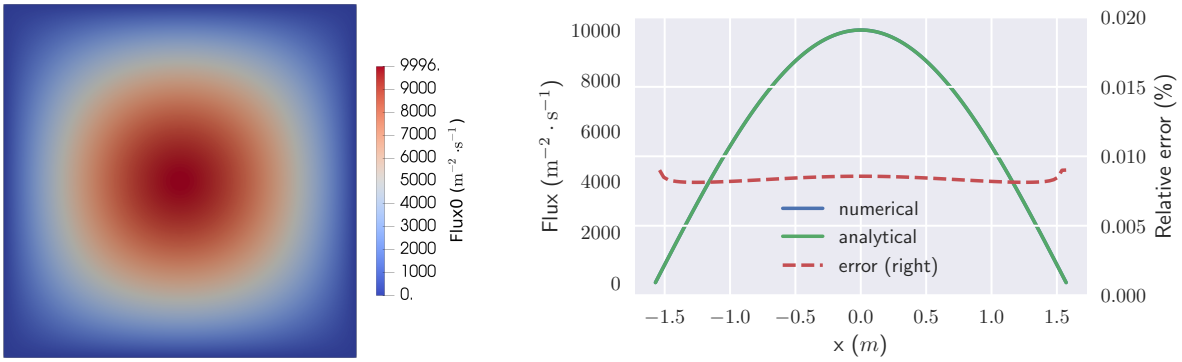

Figure 2: Flux on the domain and results over line AA'

The correct solution of the momentum equation under the conditions of this problem is quite demanding on the divergence discretization scheme of velocity. A first order upwind scheme fails to generate a correct flow and pressure fields, requiring the use of a second-order upwind scheme with results presented in figures 3 and 4. It is relevant to mention that the velocity plot in figure 3 has a blunt tip in the centre due to the mesh having an even number of cells, therefore there is no cell in the centre of the domain. A simple test was made with a cell in the centre, offering no benefit to the solution. With the lack of any benefit, it was decided not to have a cell in the centre to avoid the problem of defining a relative error in the centre of the domain, where the analytical solution to velocity is 0.

The relative error graphic for velocity shows that it peaks quickly and reduces to an approximately constant value away from the boundary. While a finer mesh at the boundary could appear to offer modest improvements, even though errors are already very low, this improvement was not observed. A mesh refinement at the walls or a mesh with refinement at the walls and in the centre did not convey any improvement over a uniform one.

Pressure has the lowest error, at approximately 0.0001%, with an unusual curve shape. Upon closer investigation of the numerical and analytical values, it is clear that the numerical value is higher than the analytical closer to the boundary while the reverse is true for the centre. Therefore, the zone of 0 error simply represents an intersection where both happen to be equal.

Solving the temperature field reveals some challenging characteristics of this case. Term dominance can easily be a problem since $\rho c_p >> k$ usually, resulting in advection strongly

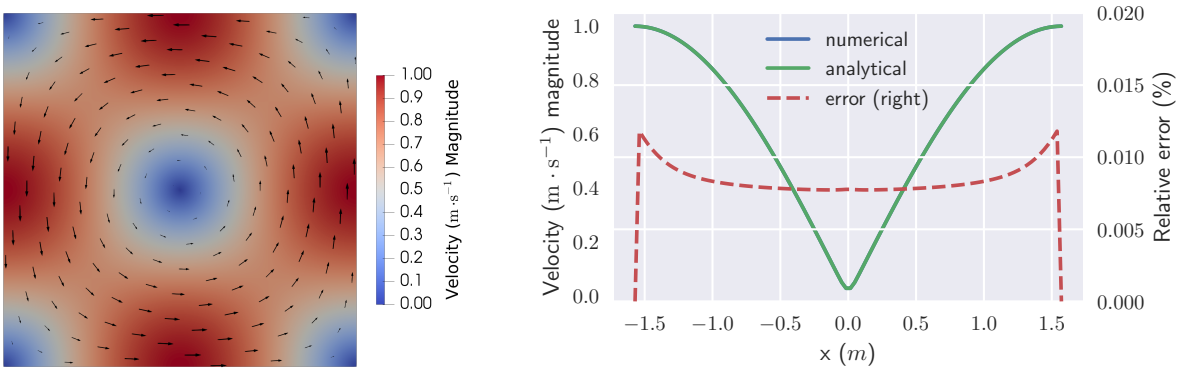

Figure 3: Velocity magnitude on the domain and results over line AA'

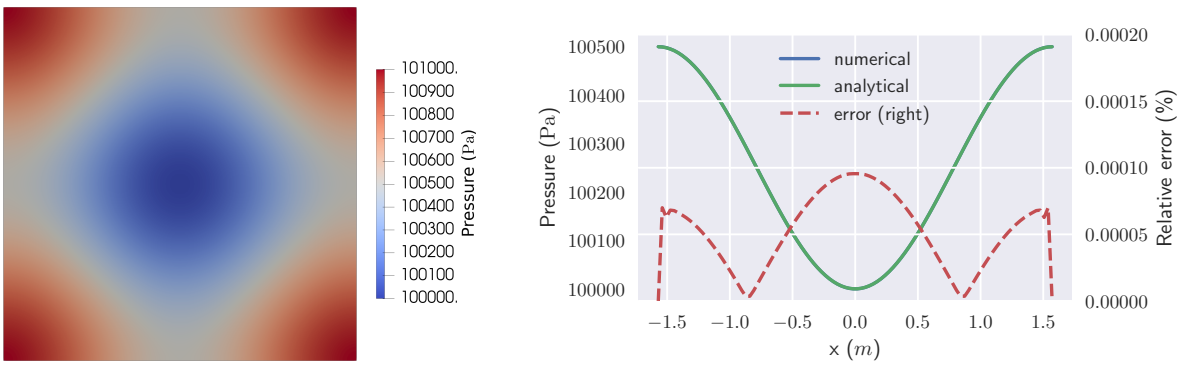

Figure 4: Pressure on the domain and results over line AA'

dominating over diffusion. Advection of energy should be analytically 0 for this problem but numerically becomes a very low non-zero number. Multiplying this numerical noise by a relatively big number results in unnaceptable noise levels that prevent the calculation of the correct temperature field. Despite the temperature field being resolved to a very low relative error as shown at figure 5, indicating convergence, the error field of this case has a unique shape that is strongly influenced by the advection noise. Investigating the best way to deal with this problem should be a priority in the future in order to increase the robustness of the proposed case.

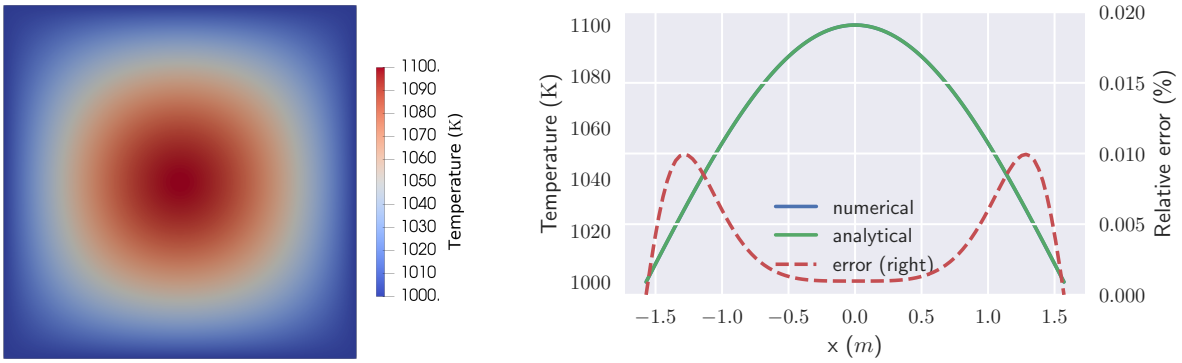

Figure 5: Temperature on the domain and results over line AA'

3.2. Mesh studies

Figure 6 shows the L_1 and L_2 norms of velocity on the domain where mesh size indicates the number of uniform cells in the x and y direction. As can be seen on the figure, the norms decrease quickly with mesh refinement, reaching an assymptotic value at 100 x 100 cells.

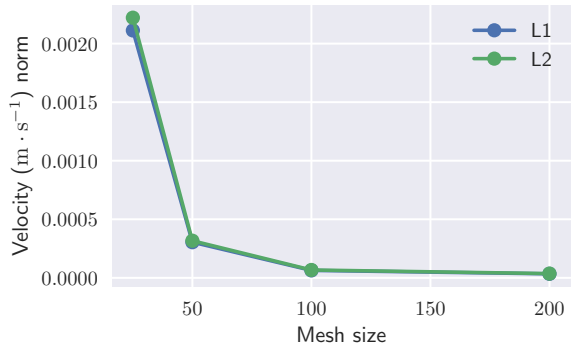

Figure 6: L_1 and L_2 norms for velocity

Norms for other variables show a similar trend, also reaching an assymptotic value at 100 cells for each dimension. Therefore, we consider the mesh converged for this problem. All graphics presented were created from results of calculation using this mesh.

The increase in wall time from a mesh refinement is significant as shown in figure 7. At mesh size 25, the minimum wall time is limited by startup operations, where a mapping of values between the fluid mechanics and neutronics meshes is established, and coded sources and coded boundary conditions are compiled. At mesh size 200 the wall time is limited by processing power. At mesh size 50, the case runs at approximately 50 seconds in a common desktop, making it a good candidate for use as a fast regression testing during code development. At mesh size 100, where mesh is converged, the problem takes only 180 seconds, making it the appropriate resolution for further developments in this problem.

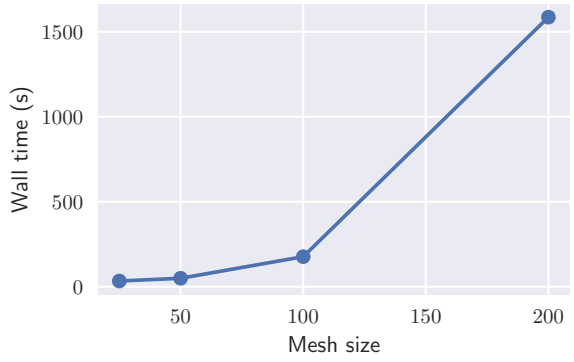

Figure 7: Wall time for different mesh sizes

4. Conclusion

The coupled fluid dynamics and neutron transport problem presented in this paper can contribute to a rigorous verification of multiphysics tools, such as GeN-Foam. To that end, we consider that the numerical experiment has been successful in demonstrating that essential functions of GeN-Foam are working correctly regarding the solvers and capabilities of the program that have been covered by this test case, which was the primary objective of this work. In addition, the presented problem has been incorporated as a regression test for GeN-Foam development by the authors, increasing the robustness of the SQA currently in place, which relies mainly on reproducing the results of reference test cases.

For future work, it would be desirable to increase the test coverage to include more parts of the program, such as: turbulence, delayed neutron precursors transport, equations of state, reactivity feedbacks and multi-group neutronics. As part of an ongoing effort, these should be added one at a time in further developments of this work, with the aim of creating a test case that can exercise the entire code selectively or simultaneously.

References

[1] C. Fiorina, I. Clifford, M. Aufiero, K. Mikityuk, GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2d/3d transient analysis of nuclear reactors, Nuclear Engineering and Design 294 (2015) 24–37. doi:10.1016/j.nucengdes.2015.05.035.
URL http://linkinghub.elsevier.com/retrieve/pii/S0029549315003829

[2] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in physics 12 (6) (1998) 620–631.
URL http://scitation.aip.org/content/aip/journal/cip/12/6/10.1063/1.168744

[3] C. Fiorina, N. Kerkar, K. Mikityuk, P. Rubiolo, A. Pautz, Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform, Annals of Nuclear Energy 96 (2016) 212–222. doi:10.1016/j.anucene.2016.05.023.
URL http://linkinghub.elsevier.com/retrieve/pii/S0306454916303309

[4] C. Fiorina, M. Hursin, A. Pautz, Extension of the GeN-Foam neutronic solver to SP3 analysis and application to the CROCUS experimental reactor, Annals of Nuclear Energy 101 (2017) 419–428. doi:10.1016/j.anucene.2016.11.042.
URL http://linkinghub.elsevier.com/retrieve/pii/S0306454916305126

[5] W. L. Oberkampf, T. G. Trucano, M. M. Pilch, On the role of code comparisons in verification and validation., Tech. Rep. SAND2003-2752, 918244 (Aug. 2003). doi:10.2172/918244.
URL http://www.osti.gov/servlets/purl/918244-tbRwVC/

[6] W. L. Oberkampf, C. J. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, Cambridge, 2010. doi:10.1017/CBO9780511760396.
URL http://ebooks.cambridge.org/ref/id/CBO9780511760396

[7] P. Knupp, K. Salari, Verification of Computer Codes in Computational Science and Engineering, Vol. 20023155 of Discrete Mathematics and Its Applications, Chapman and Hall/CRC, 2002. doi:10.1201/9781420035421.
URL https://www.taylorfrancis.com/books/9781420035421

[8] P. J. Roache, Verification and validation in computational science and engineering, Hermosa Publishers, Albuquerque, USA, 1998.

[9] S. D. Pautz, Verification of Transport Codes by the Method of Manufactured Solutions: The ATTILA Experience, Tech. Rep. LA-UR-01-1487, Los Alamos National Laboratory, Los Alamos, USA (Mar. 2001).
URL http://lib-www.lanl.gov/la-pubs/00796073.pdf

[10] S. Schunert, Y. Y. Azmy, A two-dimensional method of manufactured solutions benchmark suite based on variations of Larsen's benchmark with escalating order of smoothness of the exact solution, Rio de Janeiro, Brazil, 2011.

[11] J. Wang, W. Martin, B. Collins, Application of the method of manufactured solutions to the 1d Sn equation, in: PHYSOR 2016, Sun Valley, Idaho, USA, 2016.

[12] SAMOFAR, A Paradigm Shift in Reactor Safety with the Molten Salt Fast Reactor (2015).
URL samofar.eu

[13] W. M. Stacey, Nuclear reactor physics, 2nd Edition, Wiley-VCH, Weinheim, 2007.

Nomenclature

γ	Flux to power conversion factor
μ	Dynamic viscosity
ϕ	Neutron flux
ρ	Density
$\mathbf{F}$	Momentum source vector
$\mathbf{u}$	Velocity vector
B	Buckling

	c_p	Specific heat capacity at constant pressure
	k	Thermal conductivity
	P	Pressure
300	T	Temperature