International Conference
Nuclear Energy for New Europe

Divertor Armor Damage Estimation and Dynamics of Secondary Tungsten Plasma Simulation with TOKES in ITER

Leon Bogdanović[a], Sergey Pestchanyi[b], Leon Kos[a]
[a]Faculty of Mechanical Engineering, University of Ljubljana
Aškerčeva 6
1000 Ljubljana, Slovenia

[b]Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen, Germany
leon.bogdanovic@lecad.fs.uni-lj.si, serguei.pestchanyi@kit.edu, leon.kos@lecad.fs.uni-lj.si

ABSTRACT

The TOKES code has been extensively used for several years in specific ITER studies, as well as for JET-ILW (ITER-like wall) and the EU DEMO PFC design activities. The code is used for numerical simulations of the thermonuclear deuterium-tritium (DT) plasma dynamics in a tokamak core and in the scrape-off layer (SOL), for calculations of heat flux from SOL to the tokamak walls and for heat transport inside the solid walls. It also takes into account phase transitions of the wall material – tungsten (W), simulates the dynamics of the vaporized W in the vacuum vessel, its ionization and W-D-T plasma dynamics, including photonic radiation. The original source code of TOKES is written in Pascal and compiled in Delphi, a commercial Integrated Development Environment (IDE), under Windows on single machines. In the last year, the TOKES source code has been refactored with the open source Lazarus IDE, which uses Free Pascal as an Object Pascal dialect. Versions in Lazarus for both Windows and Linux has been developed. The Linux version also runs on the HPCFS cluster, which offers superior computing capabilities compared to single machines. This paper presents TOKES simulations of the disruptions of 250 MJ energy content in the core characteristic for ITER. The divertor armor damage due to melting and vaporization has been estimated and the dynamics of the secondary tungsten (W) plasma has been investigated. It has been shown that the secondary W plasma shields the divertor armor from hot DT plasma out of the core, and this shielding drastically decreases the divertor armor damage. Simulations were carried out both with the original TOKES code in Delphi and the refactored code in Lazarus. Comparison of the results is given to confirm the correct behaviour of the refactored code.

1 INTRODUCTION

As an integrated simulation code of plasma dynamics and surface processes in tokamaks, TOKES offers realistic estimation of the heat fluxes to the divertor armor during fast transient processes like major disruptions (MD) or Edge-Localized Modes (ELMs). The divertor armor is the most critical component of a tokamak in terms of heat load. TOKES simulates numerically the dynamics of the thermonuclear deuterium-tritium (DT) plasma in the tokamak core and in the scrape-off layer (SOL), calculates the heat flux from the SOL to the tokamak walls and the heat transport inside the solid walls. It considers phase transitions of the wall material, including melting and vaporization. After vaporization starts, TOKES simulates the dynamics of the vaporized material in the vacuum vessel, its ionization and the combined material-D-T plasma dynamics, including photonic radiation [1]. The aim of such simulations is the estimation of the tokamak walls heat load and damage. In the case of the ITER tokamak, the divertor is armoured with tungsten (W). High energy density transients in ITER operation will be sufficiently powerful to cause local melting of the divertor structures. On the other hand, W vaporized from the target at an initial stage of the disruption creates a plasma shield in front of the target, which effectively protects the target surface from the rest of heat flux, transforming it into photonic radiation. The plasma shielding effect is a complex physical phenomenon, combining magneto hydrodynamics (MHD) convection and diffusion of the W plasma shield with conversion of the transient heat flux from the core into radiation heat flux. Radiation from the plasma shield redirects the heat flux backwards and to the surrounding surfaces, including the sub-dome supporting structures ([2], [3]).

In the last year, efforts to refactor the original TOKES source code from commercial Delphi/Pascal [4] to the open source Lazarus IDE [5], which uses Free Pascal as an Object Pascal dialect, have been successful [6]. Versions in Lazarus for both Windows and Linux have been developed. The Linux version also runs on the HPCFS cluster. To test the refactored code as opposed to the original one, simulations of the disruptions of 250 MJ energy content in the core characteristic for ITER with both codes have been carried out.

Validation of the TOKES code is a rather difficult task, because the plasma parameters of modern tokamaks cannot provide the heat and plasma flux, characteristic of ITER and DEMO. Comparable plasma parameters are available in plasma guns, but for a very short time. The corresponding validation has already been performed in [7] and further validation against dedicated experiments is planned. This paper validates only the results of the refactored code in comparison with the original one.

2 TOKES SIMULATION CASE - 250 MJ INITIAL ENERGY DISRUPTIONS IN ITER

The case of 250 MJ initial energy content in the core was chosen to simulate MD in ITER tokamak with TOKES. Simulations were run with the original TOKES program (executable) on Windows and with the refactored TOKES program (executable) on Linux. The main input parameters and options of the simulation, as well as details of software/ hardware resources used, are given in Table 1 and Table 2, respectively. The former is mainly set in the input file `Start.txt`, where also input data (through input files) is defined:

- Vessel surface: `VesselSurfaceITERa.txt`
- Gases: `PlasmaITERa_W.txt`
- Beams: `DataRaysUpperInjector_W_2D.txt`
- Vessel wall: `DataWall_ITERa.txt`
- Atoms data: `DataPlasmav2.txt`

Also, a version (32-bit executable) of the refactored TOKES in Lazarus for Windows was developed but has not been used for this simulation case.

Table 1: Main input parameters and options of the simulation

TOKES simulation type		Full	
`CTcorr` (`EEtotCore`) - hardcoded		0.6815 (250 MJ)	
`EpsEE` - hardcoded		0.03 (stops at 3% of `EEtotCore`)	
Time step	$2\cdot10^{-7}$ s	Isotopes for plasma and neutrals	2H, 184W
Maximum time	$5\cdot10^{-2}$ s	Material names	184_W, 056_Fe
Save step	50	Nominal plasma current	15.0 MA
Maximum altitude	6.0 m	Loop magnetic potential	33.0 mT
Maximum radius	12.0 m	Layer magnetic width	1.5 Wb
Triangle max. size	0.3 m	Cell poloidal size	0.25 m
Wall toroidal cell max. size	6.0 m	Enlarged poloidal size	0.008 m
Plasma cross-diffusion transport		TRUE	
Plasma longitudinal transport		TRUE	

Table 2: Details of software and hardware resources used

	Original TOKES	Refactored TOKES
IDE	Delphi	Lazarus / FPC
Type of executable	32-bit	64-bit
OS	Windows 10	Rocky Linux 8.6
Machine	Personal laptop	Login node of HPCFS cluster
CPU	Intel Core i7-10510U	AMD EPYC 7302 16-Core
Max. memory available	16 GB	512 GB

3 RESULTS OF SIMULATIONS

3.1 Simulation runs

Both simulations ran for 25000 time steps (until the maximum time of 50 ms). In Table 3 details of the runs are shown. The simulation with the original TOKES executable on the laptop under Windows ran more than a day longer in total than the refactored TOKES executable on the HPCFS login node under Linux. The cause are many more crashes of the original executable, hence the simulation had to be manually resumed at the step prior to the crash every time. If the crash occurred, e.g., in the middle of the night, the simulation was halted for several hours. Most of the crashes of the original executable are due to “Out of Memory”, which is an indication that TOKES needs more memory resources than 16 GB available on a typical laptop. On the contrary, the refactored TOKES executable ran quite smoothly on a server-like machine abundant with memory (512 GB). As for the performance of the executables, the original code has a wider range of run times per step, probably due to limited memory resources available. For every 50^{th} time step (see Table 1) the simulation outputs were stored into result text files with P2DT suffix. One can see that the total size of output result files for the refactored code is lower than for the original code. Until time step 5000 (at 10 ms simulation time), when the processes in the system have finished or reached a steady state, the difference in size of the output is negligible. Both simulations show the value of rest of energy in the core (EEtotCore) at less than 3% of the initial value, hence all subsequent results after this time step are not relevant and can be ignored in the analysis.

Table 3: Simulation runs details

	Original TOKES	Refactored TOKES
Total run time	160 h	130 h
Run time per step (range)	13-53 s	17-31 s
Crashes of the simulation	6	2
Cause of crash	“Out of Memory” (4x), “Invalid floating point operation” (2x)	“Access violation”, “Electron thermal conductivity is not available”
Total size of output	8.43 GB	6.93 GB
Size of output (till step 5000)	2.01 GB	1.94 GB

3.2 Comparison of results

The results of the simulation are stored as *.P2DT files and will be analysed and displayed with the vw2DFMC viewer (the original viewer for TOKES simulation results).

On Figure 1 comparison for some parameters in the first 5000 time steps (until 10 ms) of both simulations is given. The time dependent curves agree well for maximum melt depth (Figure 1 (c)) and wall heat flux (Figure 1 (d)), while for the rest of energy in the core (Figure 1 (a)) and vaporization erosion (Figure 1 (b)) discrepancies from 6 ms and 2 ms onwards, respectively, can be observed. The lower curve on Figure 1 (a) and the upper curve on Figure 1 (b) are the result of simulation with the refactored code. These deviations can be explained by Monte Carlo procedures and rather turbulent dynamics of the whole system. To confirm this assumption another simulation run with the original code with the same parameters until time step 4000 (until 8 ms) has been performed. On Figure 2, time dependent curves for the rest of energy in the core and vaporization erosion for both runs with the same (original) code are shown. Both curves from the second simulation run (lower curves on Figure 2 (a) and Figure 2 (b)) differ from the curves of the first simulation run. Again, discrepancies from 6 ms and 2 ms onwards can be observed, respectively. It can be concluded that the refactored code gives comparable results to the original code, hence it can be used as replacement for simulations.

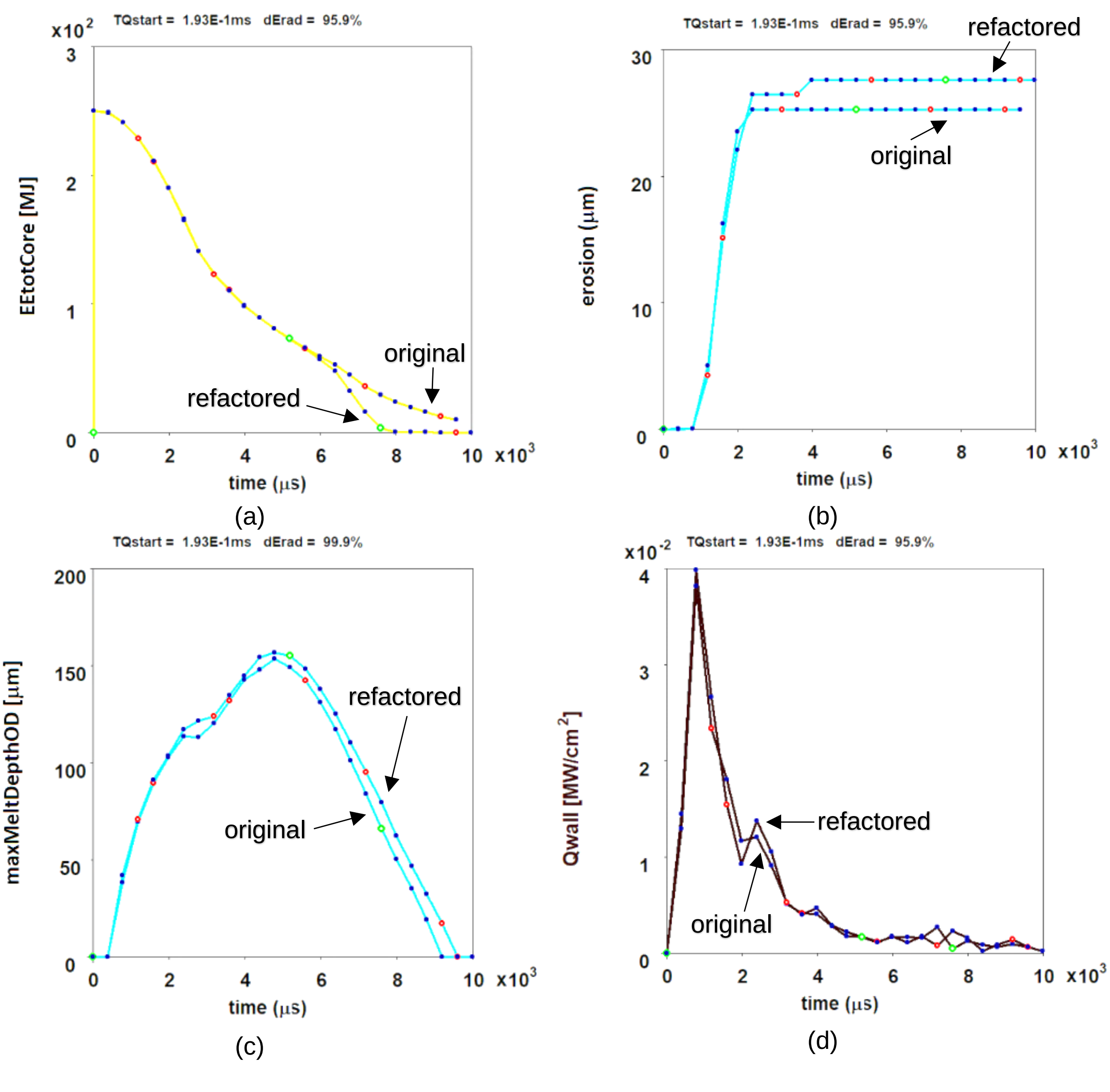
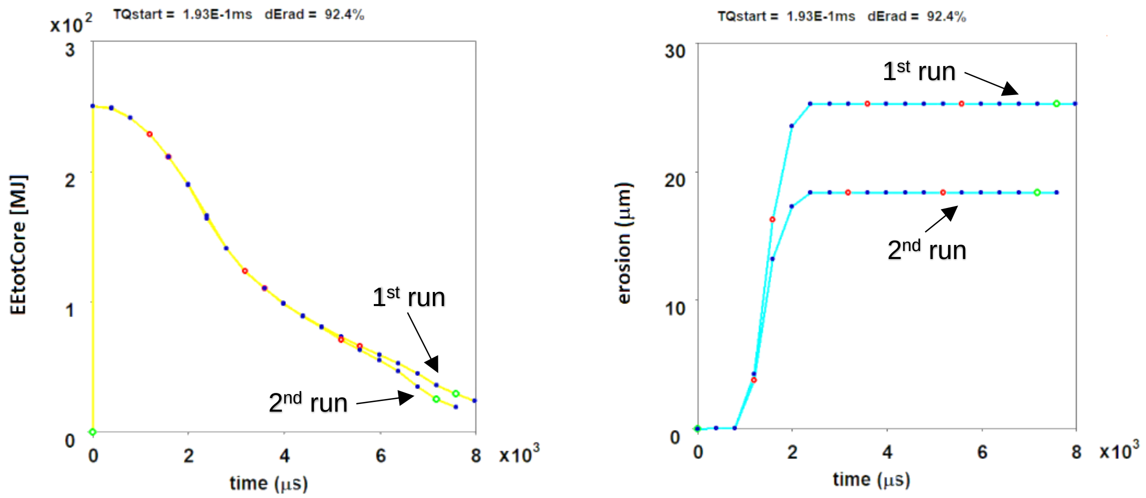


Figure 1: Comparison of time dependency for some simulated parameters from both codes.
(a) Rest of energy in the core. (b) Maximum vaporization erosion depth.
(c) Maximum melt depth. (d) Wall heat flux.

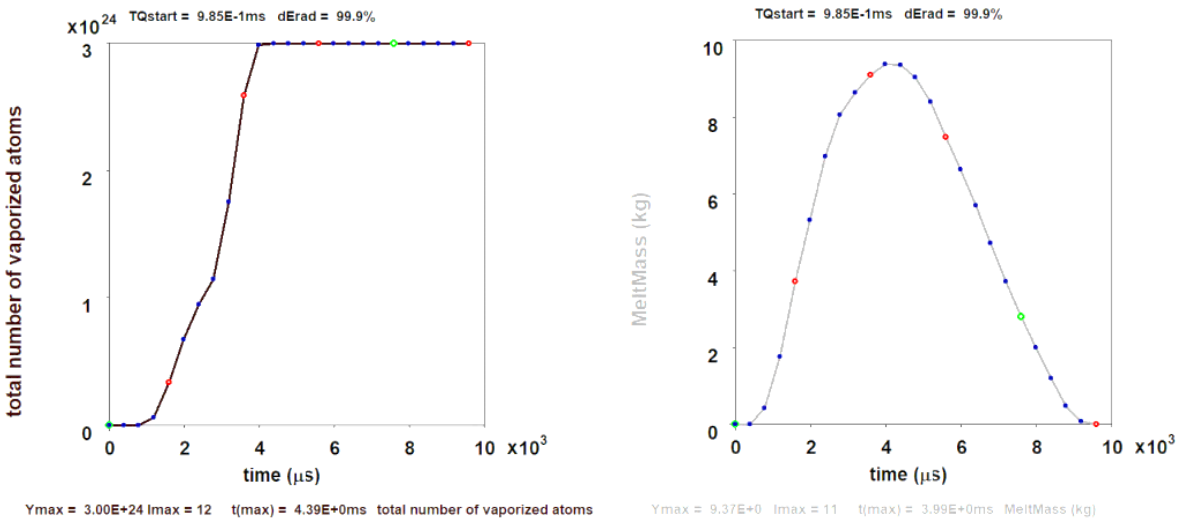


(a) (b)

Figure 2: Comparison of two simulation runs with the original code (until time step 4000). (a) Rest of energy in the core. (b) Maximum vaporization erosion depth.

3.3 Divertor armor damage estimation due to melting and vaporization

Divertor armor damage can be illustrated by plotting the diagrams of vaporized mass (total number of vaporized atoms), melt mass and maximum melt depth. Figure 3 shows the time dependency of the first two parameters. It can be deduced that vaporization stops after about 4 ms from the start of simulation (Figure 3 (a)) and melting reaches the maximum at 4 ms and then drops to zero at less than 10 ms (Figure 3 (b)). The maximum melt depth of 157 μm is reached at 4.8 ms (upper curve in Figure 1 (c)).



(a) (b)

Figure 3: Melting and vaporization of the divertor armor. (a) Vaporized mass. (b) Melt mass.

3.4 Dynamics of the secondary tungsten plasma

The sequence of the plots on Figure 4 illustrates the dynamics of the W plasma cloud. It can be seen that the W plasma cloud is expanding along the separatrix starting from both separatrix strike positions (SSP) at divertor targets to the X-point and eventually entering the core. The effect of divertor armor shielding by the W plasma can be seen when comparing the evolution of the W plasma cloud on Figure 4 with wall heat flux time dependence on Figure 1 (d). The heat flux to the wall drops abruptly from its maximum at 1 ms to 25% of the maximum value at 2 ms when sufficient amount of W has been evaporated and the W shielding is fully developed in front of the armor. At 4.0 ms, when the W plasma cloud has reached the X-point and started entering the core, the vaporization stopped, and melting started to decrease (Figure 3).

The high wall heat flux value with short time duration for the disruption of 250 MJ is qualitatively identical to the results of disruptions of 350 MJ and 280 MJ [2]. In all the three cases the W plasma cloud shields the divertor armor by effectively lowering the heat flux value at the wall surface.

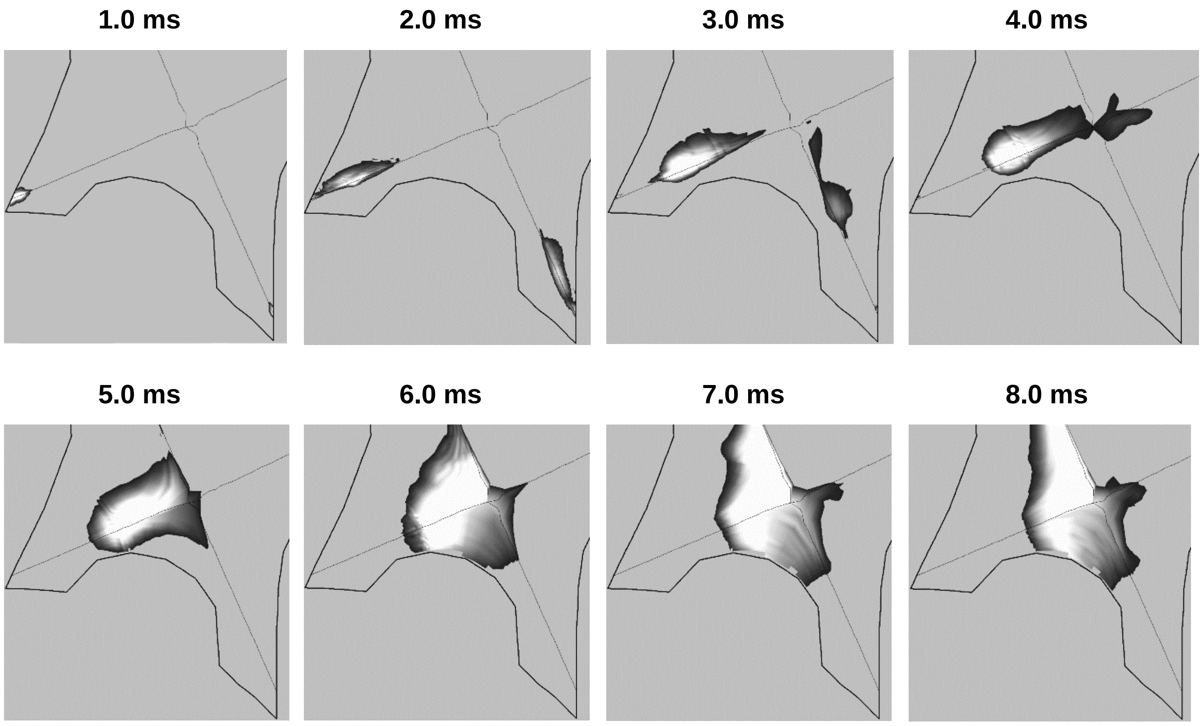


Figure 4: Dynamics of the secondary W plasma cloud. Time moments above the plots are from the start of the simulation. Shown are the W plasma densities in grey scale. Black corresponds to the minimum value and white to the maximum value of density.

4 CONCLUSIONS

The simulations of the disruptions of 250 MJ energy content in the core characteristic for ITER were used as a test for the refactored TOKES code in Lazarus running under Linux on the HPCFS cluster. The refactored code gave comparable results with more stable performance mainly due to the availability of more abundant memory resources. Observed discrepancies in the results are to be attributed to Monte Carlo procedures and rather turbulent dynamics of the TOKES code. The case of 250 MJ is qualitatively identical to more powerful disruptions (of 350 MJ and 280 MJ) with high wall heat flux values and short time duration. Simulation results with the refactored TOKES code show that the W plasma cloud shields the divertor armor by effectively lowering the flux value in the wall and hence decreases the divertor armor damage.

ACKNOWLEDGMENTS

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

REFERENCES

[1] I. S. Landman, "Tokamak code TOKES. Models and implementation", Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft Wissenschaftliche Berichte, FZKA-7496 (September 2009).

[2] S. Pestchanyi, et al., "Simulations of Energy Loads and their Mitigation during Disruptions and Runaway Electron Formation in ITER, Part-1: TOKES Simulations", Final Report of Work incl. Final Conclusions & Recommendations (incl. work on VDEs & Vapour Shielding), KIT Document Ref. No. MOD-PEW-145842-RD-2D04 (November 2020).

[3] S. Pestchanyi, R. Pitts, M. Lehnen, "Simulation of divertor targets shielding during transients in ITER", Fusion Engineering and Design, Vol. 109-111, Part A, 2016, pp. 141-145, ISSN 0920-3796, https://doi.org/10.1016/j.fusengdes.2016.02.105.

[4] Delphi: a software product that uses the Delphi dialect of the Object Pascal programming language, https://www.embarcadero.com/products/delphi, 2023. Accessed: 2023-08-29.

[5] Lazarus: a Delphi compatible cross-platform IDE for Rapid Application Development, https://www.lazarus-ide.org/, 2023. Accessed: 2023-08-29.

[6] L. Bogdanović, S. Pestchanyi, L. Kos, "Towards Refactoring Of The TOKES Tokamak Plasma Transient Code", Proceedings of the 31st International Conference Nuclear Energy for New Europe (NENE 2022), Portorož, Slovenia, September 12-15, Nuclear Society of Slovenia, 2022, pp. 1014.1-1014.8.

[7] S. Pestchanyi, R. A. Pitts, V. Safronov, "Validation of TOKES vapor shield simulations against experiments in the 2MK-200 facility", Fusion Engineering and Design, Vol. 124 (2017), pp. 401–404.