Proceedings of the Twelfth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 8, Paper 1.1
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.8.1.1

Automated Machine Learning Workflows for Fusion Power Plant Design

W. Smith[1], A. J. Barker[2], Z. Miao[1], O. Woolland[3], M. Omer[1] and L. Margetts[1]

[1] School of Engineering, University of Manchester, UK
[2] School of Natural Sciences, University of Manchester, UK
[3] Research IT, University of Manchester, UK

Abstract

The need to meet increasing global energy demand and also address 2050 net zero targets is placing fusion energy in the spotlight. The authors are investigating the advances in digital technology necessary to deliver a fusion power plant by 2040. We are currently evaluating the use of the Nvidia Omniverse platform for engineering, with the specific need to consider fusion power plants as a whole system. In a plant that uses a magnetically confined plasma, fusion generates neutrons which pass through the whole machine. The design process needs to consider how to protect some systems from the neutrons, whilst in other parts of the machine, the neutrons can be used to generate new fuel. It is difficult to compartmentalise the design process for these two opposing requirements. Therefore a requirement for conceptual design is the ability to carry out fast physics-informed simulations for many coupled systems at the same time. This paper describes how the authors have integrated the Galaxy workflow engine with the Omniverse, automating the execution of a suite of containerised open source software applications that can be used for training surrogate models. Automation is essential if AI and machine learning is to be leveraged in the design of complex engineering systems.

Keywords: metaverse, digital twin, workflow, machine learning, nuclear fusion, design

1 Introduction

Large sources of clean and reliable energy are needed in a world with exponentially increasing energy demand and an ever-present climate threat. Current aims to reach net zero by 2050 need new technologies to be able to achieve this goal [1]. Fusion energy is one of the solutions being investigated as a source of clean energy, intending to put power on the grid with pilot fusion power plants in the next couple of decades [2, 3]. However, fusion presents a complex, multi-faceted suite of scientific and engineering challenges, requiring many teams of experts in science and engineering to collaborate in the design, construction and operation of power plants. These teams are spread over geographically dispersed locations due to the range of knowledge and expertise needed, often requiring international collaboration [4]. Next-generation collaborative design tools are necessary for the discipline to move at the pace required to deliver fusion energy.

The proposed solution is to exploit the vision of an industrial metaverse that combines connected digital twins with processes for industrial design, manufacture, operation, and decommissioning. Digital twins connect physical facilities and virtual models through the exchange of data, leading to a twin in cyberspace that contains all the information about its physical counterpart [5], which can cover the full lifetime from cradle to grave [6]. In this paper, we propose building digital twins using the Nvidia Omniverse platform, extending their ecosystem with a range of domain specific microservices to provide links to enhanced simulation capabilities, data provenance capture and experimental facilities.

One of the extensions to the Nvidia software is the integration of a workflow engine to automate design space exploration. Workflow engines wrap simulation programs as tools. Tools do not change anything about how the simulations run; instead, they allow for the sharing, reuse, versioning and reconfiguration of tools into workflows (chains of tools with a defined flow of data) for different application scenarios. Working with reconfigurable tools prevents replication of work or bespoke automation scripts written for specific use cases. Furthermore, the workflow engine allows experts in a given field to work with unfamiliar tools where they know the science of the inputs, outputs and techniques but do not have to learn the implementational details of yet another tool or spend time configuring it. This allows the experts to focus on the interpretation of the results rather than getting held up in making simulation codes work on their machines. This replaces the set-up of tools, and not the responsibility for executing them and interpreting results.

Finally, with more complex models required to account for all the physics interactions that affect a fusion power plant, there is an ever-increasing requirement for advances in simulation speed and assistive tools for engineers. One promising approach to address this challenge is the use of surrogate models. Surrogate models offer efficient and accurate approximations of complex systems, enabling quicker simulations and analyses while maintaining precision. They encompass various data-driven algorithms, such as response surface methodology [7], Gaussian processes [8], neural networks [9], support vector machines [10], and radial basis functions [11]. Given the challenges in acquiring data for fusion reactor components, both experimentally and through simulation, physics-informed machine learning based surrogate models are particularly advantageous. These models integrate established physical laws, constraints, or principles into their framework, blending empirical data with theoretical insights to better capture the complexities of fusion reactor dynamics. To fully leverage the capabilities afforded by machine learning now and in the future, automation of training and operation of models is a necessity for continued use. Operating reconfigurable, automated workflows for executing analyses may also offer a solution, especially when combined with an extensible platform that can link to many other applications, such as those used to create drawings or CAD.

In the subsequent sections, we briefly overview the technologies used in the integration and the reasons for their selection. Then, we give an overview of the implementation and briefly describe how some of the integration steps were performed. Finally, a case study is presented, showing some of the functionalities implemented in the context of fusion power plant design.

2 Software Stack

This section briefly describes the software packages used to build an integrated digital engineering platform for fusion power plant design.

2.1 Digital twin authoring tools

Many platforms have been used to build digital twins. Some of these come from organisations primarily focused on game development, using game physics engines to drive real-time low-resolution multi-physics simulations that can be closely coupled to their physical counterpart; examples of this include Unity and Unreal Engine [12, 13]. Other vendors such as COMSOL [14] provide solutions to integrating industry standard simulation codes into a single multi-physics platform with the aim to better replicate the physical world. The NVIDIA Omniverse is another option, that provides a lightweight, extensible framework, allowing multiple 3D applications to be integrated into one connected platform.

The NVIDIA Omniverse platform [15] consists of a series of services, including SDKs, APIs, and internal simulation tools, all interconnected through a central file system — the Nucleus. Nucleus uses the Universal Scene Description (USD) file format, developed originally by Pixar for transferring files along their animation pipelines [16]. This allows 3D, connected applications to communicate via a single, common file type and take advantage of real-time updates in and from all connected applications from multiple users in geographically dispersed locations. The Omniverse has a series of tools tightly integrated via so-called connectors, such as applications from the Autodesk suite, Kitware ParaView, Trimble SketchUp, Unity and Unreal Engine, to name a few. These all communicate in real-time via the Nucleus, and updates made in one instance are instantly reflected in all others. Omniverse has been chosen as the platform to investigate for fusion engineering, due to both the features described and its ability to easily integrate other services via lightweight, less tightly coupled, extensions that can provide extra functionality in addition to what is already available.

The Omniverse also allows for oversight of project progress via its own developed applications, such as the USD Explorer, which aggregates digital assets and views data in a top-down view of the system, or the USD Presenter, which allows for reviewing projects with real-time photo-realistic rendering.

2.2 Workflow engine

Workflow engines have found the most traction in the bioinformatics field where many simulation and processing steps are required. Several workflow engines were considered for use in this project and a comparison between the most prominent and mature examples was performed. They were compared using a range of criteria such as the ability to run containerised applications, offload computation to HPC, and project support/maturity (based on GitHub statistics). Other categories, such as previous experience in the research group, cross-platform compatibility, compatibility with universal workflow and tool description standards, and graphical vs solely command-line-driven interfaces, were also considered. Figure 1 shows a table comparing the workflow engines considered. Additional criteria used for comparison have been omitted from the table for brevity as most of the compared engines had the same features: documentation, tutorials, and the use of containerised applications.

The Galaxy workflow engine [17] was selected for the work presented in this paper. Galaxy offers many benefits over its competitors, such as an easy-to-use API and a graphical interface provided via a web page. This allows subject matter experts, unskilled or uninterested in computer administration, to quickly learn how to interact with tools, create workflows and run simulations via a simple web UI that can be accessed anywhere, on any device with a browser. Developing tools for the Galaxy workflow engine is also easy with simple XML tool descriptions, allowing the execution of jobs in containerised environments. The engine also has good links to HPC with the Pulsar runner [18], a Python-based server application that sits on a remote host without the need for any shared file systems. This has been used in very large Galaxy instances such as the usegalaxy.eu site, connected to many European supercomputers through a network of Pulsar endpoints [17].

2.3 Application software

As with many areas in science and engineering, there are multiple options for simulation tools, some of which are commercial or closed-source and others that are open-source. In fusion engineering, there are many specialist simulation packages that are used to deal with coupled physical processes in and around a fusion power plant. This

	Experience	Cross Platform	GUI	API	HPC	Cloud	State Management
Snakemake	N	N	N	N	Y	Y	Y
Toil	Y	N	N	N	Y	Y	N
Galaxy	Y	Y	Y	Y	Y	Y	Y
Pegasus	N	Y	Y	Y	Y	Y	Y
Cwltool	Y	N	N	N	N	N	N
Cylc	P	N	Y	Y	P	N	Y
Nextflow	N	N	Y	Y	Y	Y	Y

Figure 1: Comparison between the available open-source workflow managers.

includes the magnetohydrodynamics of the plasma, the particle physics of the fusion process and thermo-mechanical structural analysis. For the case study presented in this paper, a particle physics simulation tool is used to predict the heating due to neutron interaction with engineering materials. There are many options to consider for both neutronics and thermal analysis.

For the neutronics simulation, commonly used tools such as MCNP and FLUKA are not generally available (and the source code is not accessible). OpenMC and GEANT4 are both open-source and freely available. OpenMC [19] was chosen for this case study due to its availability and wide adoption in the fusion research community. It is still quite new compared to other offerings but has many features constantly growing from a large contributing community. It is easy to use with a simple Python wrapper, is scalable through parallel execution and is well documented, with active forums for support.

In the case study, software for thermal analysis in solids is also required. The neutronics software outputs an estimate of heating due to neutron interaction with the engineered components, but does not compute the resulting temperature change or stress. To demonstrate the use of both standard simulation tools and the use of machine learning tools in workflows connected to the industrial metaverse, a Physics-Informed Neural Network (PINN) is used to calculate the temperature due to neutronic heating. The PINN models are set up with NVIDIA Modulus [20], an open-source framework for building physics-ML models via a simple Python interface. To compute the temperature change, there are many standard tools available, again both licensed and open-source such as ParaFEM [21]. Here, the FEA software Abaqus has been used to solve the same problem so that the results can be compared with those obtained using the PINN. The PINN trains the surrogate model using fundamental physical equations, therefore reducing the need for data to train the model. Training may take a similar amount of time as a typical FEA simulation, but subsequent predictions will be much faster using the PINN compared with the FEA.

3 Software Integration

The software packages presented exchange data using MQTT messages via a messaging broker. MQTT is a standards-based messaging protocol, or set of rules, used for machine-to-machine communication. This allows for easy integration of other packages and services without hard-coding.

Figure 2 shows the general architecture. The user has several potential interaction points, primarily the Omniverse. Using extensions, Omniverse can communicate with the messaging broker in the centre of the architecture figure, which is connected to many applications such as the Galaxy listener. The platform can also receive streams of data from physical twins and other services Finally, connections are made remote HPC and Cloud computer resources using Pulsar [18]. The rest of this section will give further detail on the specific integration of the services in the software stack.

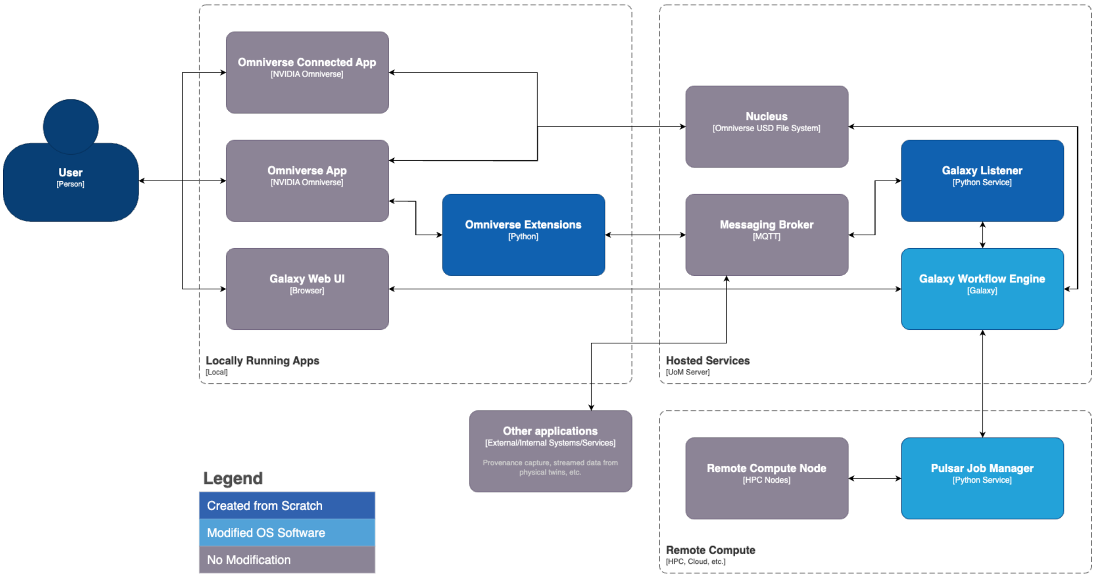

Figure 2: Architecture diagram of the connections in the software stack, and key to indicate which components are stock, modified or completely new.

3.1 Omniverse extension

The method of interaction from the Omniverse with the Galaxy workflow engine is via lightweight Python extensions/services that talk to the central messaging broker to submit jobs, receive data back, and see updates on the status of the workflow run. There are two components to enable this: a Python extension, which provides a Galaxy UI inside the Omniverse application, and the Galaxy listener, which acts as a go-between for the messages and the Galaxy instance.

The Omniverse extension is simple and follows many of the example UIs provided by NVIDIA for the graphical interface [22]. This consists of boxes to input personal authentication details, such as an API key, and to pull information about the available workflows, such as their inputs. Under the hood, this sends messages on a general topic that the Galaxy listener subscribes to send information back to the extension about workflows and inputs. When a message has been sent, a listener is created in the extension to listen for the returning message (identified by a unique ID) so incoming messages can be acted upon to update the user.

The Galaxy listener is a persistent Python service that listens to the messaging broker for instructions about what interactions need to be made with the Galaxy instance itself. Once the message has been parsed, an API request from the listener service is made to Galaxy via the BioBlend REST API Python wrapper [23]. This enables the full suite of interactions with Galaxy as a user would have via the web UI but in an object-oriented manner. Once Galaxy has fulfilled the request, the return can be sent from the listener back to the Omniverse extension, and the user can be updated on the progress made by the workflow engine.

3.2 Implementing Galaxy tools

The simulation packages mentioned earlier, OpenMC and NVIDIA Modulus, are integrated into workflow tools. Both packages have run scripts developed for them by experienced users, exposing configuration options that field experts could understand and modify to run the tools and get meaningful output. Both tools have Docker containers [24] created as lightweight virtual machines as an environment to run the package in. The tools are tested with specific scripts in containers to verify the functionality before wrapping them into a tool. Tools, as seen on the Galaxy instance, are XML files which describe the execution environment (Docker), the commands to run in the environment (the created scripts), the inputs and outputs the tool expects, and some other general data for usability when deployed on the web UI. Finally, the tools are deployed and tested against standard inputs on the Galaxy instance.

Further to the two main tools, over 10 other tools were created as part of this simulation pipeline. Tools used for pre- and post-processing are made. Most of these are reusable in many different workflows, so they avoid much of the wasted time of analysts attempting to convert between different 3D file types and run processing on data. Instead, they can just drag and drop some pre-made conversion tools and have the conversions take place automatically, freeing analysts to focus on input generation and interpretation.

3.3 Other considerations

Central to the software integration architecture, the message broker easily captures all system interactions. Who did what, when, and why? The provenance that can be captured in a system like this is essential for quality control, ensuring a trail of accountability in design and the simulations leading up to each design point. This can also ensure that only the correct people execute simulation workflows for which they have permission and relevant expertise. Also, data provenance capture over an extended period builds a knowledge base of how engineers work and their choices. The knowledge from provenance capture could eventually be used to train engineering AI assistants so humans and machines can work together in the design process; expert knowledge from humans with AI-prompted options.

Finally, the Galaxy workflow engine can be linked to multiple remote computing endpoints such as local servers, cloud computing resources or HPC systems. This works with a small service (Pulsar [18]) running on the compute resource, waiting for more IOT-style messages sent from the main Galaxy instance with run information about submitted jobs. The remote compute resource then performs the simulation and returns the results using the same method.

4 Fusion Design Case Study

This section uses the integrated engineering platform in a case study to evaluate a simple fusion power plant design concept.

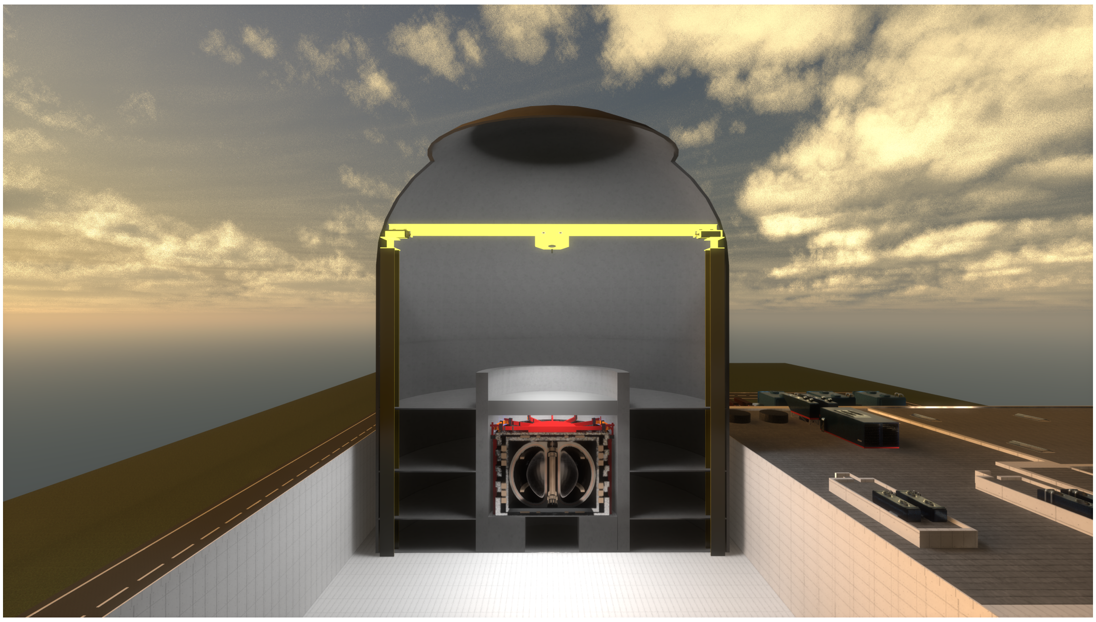

Figure 3: Parametrically generated Menard spherical tokamak, created using procedural design rules and displayed in an Omniverse power plant scene

4.1 Menard tokamak

The Menard tokamak is based on the concept of compact tokamaks, aimed at accelerating the development of fusion power [25]. The set of designs leverages the latest advances in high-temperature superconducting magnet technology, allowing for more compact spherical tokamaks with higher performance than typical larger aspect ratio (doughnut-shaped) machines. More compact devices also result in typically lower costs, faster development and build time, and therefore, faster iteration of designs and technologies used. Some test reactors currently in operation, such as MAST-U (Mega

Ampere Spherical Tokamak-Upgrade) [26], have already begun to investigate some of the confinement and performance benefits of this concept. Current plans for devices using newer magnet technology are being explored to put power on the grid soon, such as STEP (Spherical Tokamak for Energy Production) [2] and Tokamak Energy's ST80-HTS and ST-E1 prototype [3].

A general case from the Menard paper [25] is used here and converted into a parametric CAD generation code suite, allowing for geometry manipulation via simple input parameters. The CAD, generated via tools running on the Galaxy workflow engine, is then converted to a USD file, the base format for use in the Omniverse ecosystem. This can be inserted into an already authored industrial metaverse scene of the larger power plant building model as seen in Figure 3. This power plant model is created via set design rules via one of the connected 3D applications to the Omniverse - the Unity game engine and then available to both display as a rendering model and also to use the USD assets in other areas of the connected platform.

4.2 Problem description

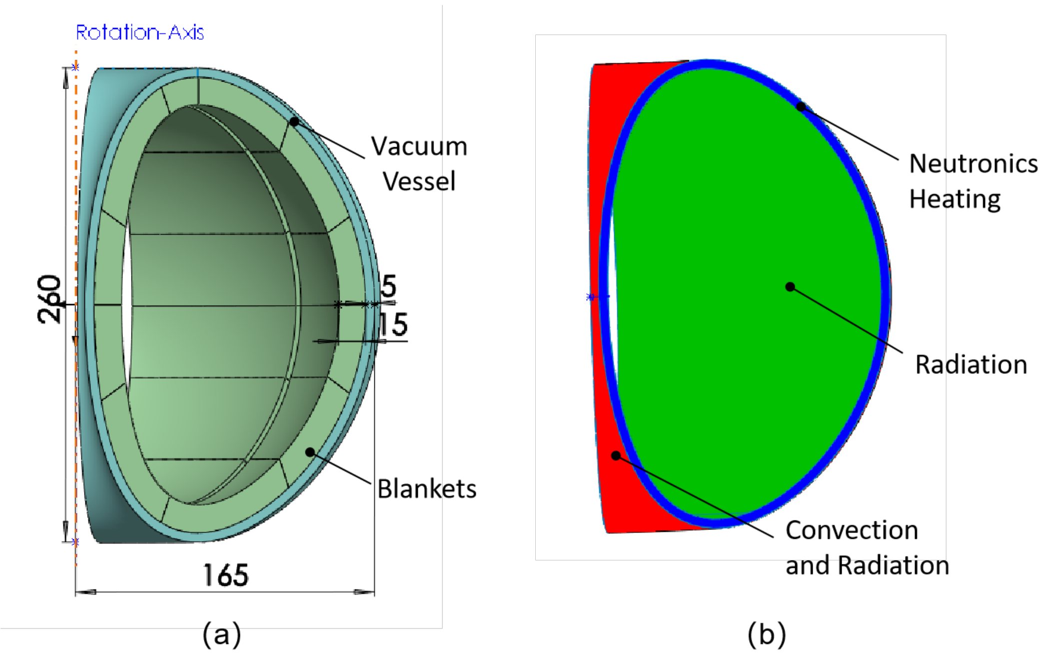

Figure 4: (a) Simplified CAD geometry of a symmetric section of a spherical tokamak with blanket and vacuum vessel wall (b) Boundary conditions for the heat transfer analysis of the vacuum vessel

The aim in this research activity is not to perform any new science but instead to develop the techniques and create the architecture of the full stack of interconnected packages, in the process of evaluating Omniverse as a platform for fusion engineering. Therefore, a simplified model has been created to reduce the computational requirement in the multiple cycles of platform iteration and testing. A simple geometry was

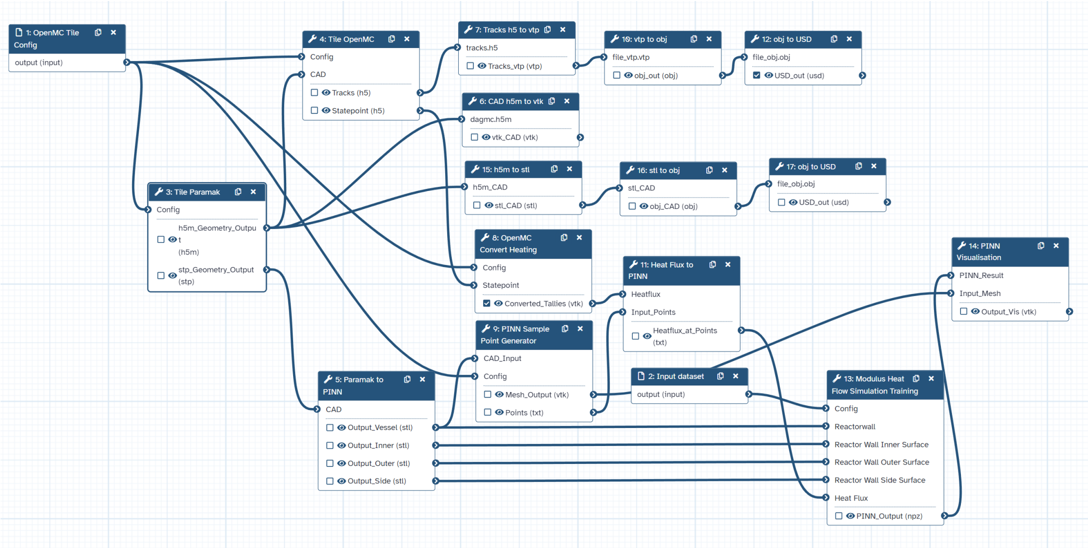

Figure 5: Galaxy workflow for the case study. The blocks are processing or simulation steps, and the lines describe the data flow between them.

Neutronics Simulation Configuration		**PINN Configuration**	
Material of Vacuum Vessel	Eurofer	Number of layers	6
Material of Blankets	Beryllium	Number of hidden layer nodes	512
Tokamak Source settings	From Fausser (2012) [27]		
Fusion power	3 MW	Activation function	SiLu
Total number of neutrons	500k	Optimizer	Adam

Table 1: Configuration of Neutronics Simulation and PINN

created using the open source parametric fusion power plant design package Paramak [28], with this geometry generation step also being integrated into the workflow.

The problem is set out that neutrons produced by fusion events occurring in the plasma would interact with both the shielding and those that made it through would also interact with the vacuum vessel. The vacuum vessel needing to retain structural integrity under the loads of vacuum could fail due to excessive heating from these neutrons. Therefore, a multi-physics simulation to first track neutron interactions in the vacuum vessel and consequently the heating flux deposited within the structure, and then subsequently calculate the steady-state temperature field due to this and set levels of cooling throughout the volume.

The specific simulation inputs can be seen in Table 1 for both the neutronic simulation and the PINN. Furthermore, the boundary conditions set on the geometry for the steady-state temperature field calculation are illustrated in Figure 4.b. Neutronics simulations were conducted on both the vacuum vessel and blanket to determine the neutronics heating within the vacuum vessel. The resulting flux was applied as an external heat source on the vacuum vessel. The inner surface of the vacuum vessel is exposed to radiation with an assumed ambient temperature of $347°C$ [29]. The outer surface of the vacuum vessel is subject to both convection and radiation, with an ambient temperature of $20°C$. The side surfaces are treated with symmetry boundary conditions. The final simulation pipeline can be seen in Figure 5 a typical disply using the Galaxy web UI. This shows the tools used as the blocks on the screen and the links between them, describing the data flow throughout the workflow.

4.3 Results

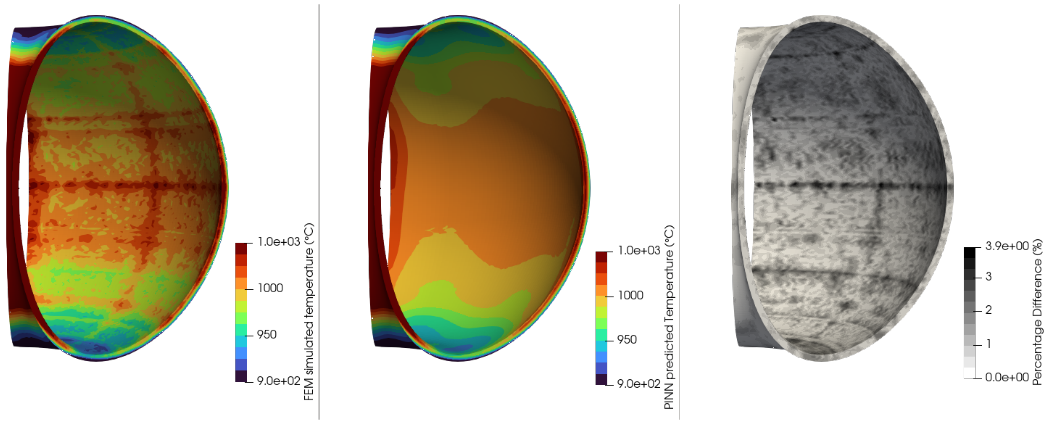

Figure 6: Plots of temperature field from the FE simulation and PINN prediction from the neutronic heating. Abaqus simulation (LEFT), the PINN prediction (CENTRE), and the percentage difference between the two (RIGHT)

The results of the PINN were compared with values obtained using the FEA package, Abaqus [30]. This used the same setup as the PINN with the same boundary conditions and 3D neutronic flux output from OpenMC to validate. The discrepancies between the results were assessed, revealing an average percentage error of 0.5% and a maximum percentage error of 3.9%.

Outside the validation of the PINN model, the results are visualised in the Omniverse as shown in Figure 6. When using the extension in the Omniverse, the launching, visualising, and subsequent work analysing the simulation results can all be done from one portal into the integrated system while allowing for real-time collaboration from colleagues or external third-party vendors.

5 Conclusions

Omniverse, as the platform for the industrial metaverse, provides engineers with reusable and reconfigurable simulation workflows via Galaxy, real-time collaboration with connected applications and services, and many more options due to its extensibility. Engineers and field experts can use unfamiliar tools with automatic data capture, focusing on interpreting results and subsequent actions rather than simulation intricacies, reducing the time spent on non-engineering tasks. ML models further accelerate design iteration with surrogate models and PINNs, giving near real-time feedback, thus enabling greater capacity for additional design optimisation and technological advances. Furthermore, the platform's automation capabilities are essential to effectively using the models, ensuring that model training and use are streamlined.

In addition to the case study presented, a fusion energy design problem, the platform is easily transferable to many application areas. This is partly due to the reconfigurability of tools in the workflow engine and the architecture's extensibility. With HPC, data provenance, and additional apps added to the platform as presented, the effectiveness and applicability can be enhanced further. Data provenance gives quality control and accountability for all tasks and decisions. Furthermore, AI engineering assistants generated by captured data could be used to enhance human-machine collaboration.

In summary, the integrated platform significantly improves time efficiency, reducing repeated, trivial, non-engineering tasks, allowing engineers to focus on engineering. NVIDIA Omniverse integrates all aspects of the industrial metaverse, and AI/ML techniques accelerate design iterations.

Acknowledgements

This project has been supported by UK Atomic Energy Authority through the Fusion Industry Programme. The Fusion Industry Programme is stimulating the growth of the UK fusion ecosystem and preparing it for future global fusion powerplant market. More information about the Fusion Industry Programme can be found online: https://ccfe.ukaea.uk/programmes/fusion-industry-programme/

References

[1] UK Gov. Net zero government initiative: UK roadmap to net zero government emissions. *DOE Roadmap*, 2023.

[2] H. Anand, O. Bardsley, D. Humphreys, M. Lennholm, A. Welander, Z. Xing, J. Barr, M. Walker, J. Mitchell, and H. Meyer. Modelling, design and simulation of plasma magnetic control for the spherical tokamak for energy production (STEP). *Fusion Engineering and Design*, 194, 2023.

[3] Melanie Windridge. Tokamak energy. In *Commercialising Fusion Energy: How small businesses are transforming big science*, pages 5–1–5–15. IOP Publishing, 2020.

[4] A. Devred, I. Backbier, D. Bessette, G. Bevillard, M. Gardner, C. Jong, F. Lillaz, N. Mitchell, G. Romano, and A. Vostner. Challenges and status of ITER conductor production. *Superconductor Science and Technology*, 27(4):044001, 2014.

[5] Michael Grieves. Digital twin: Manufacturing excellence through virtual factory replication. *Dassault Systèmes*, 2015.

[6] April Lia Hananto, Andy Tirta, Safarudin Gazali Herawan, Muhammad Idris, Manzoore Elahi M. Soudagar, Djati Wibowo Djamari, and Ibham Veza. Digital twin and 3d digital twin: Concepts, applications, and challenges in industry 4.0 for digital twin. *Computers*, 13(4):100, 2024.

[7] Raymond H Myers, Douglas C Montgomery, and Christine M Anderson-Cook. *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.

[8] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

[9] Bayya Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[10] Alex M Andrew. An introduction to support vector machines and other kernel-based learning methods by nello christianini and john shawe-taylor. *Robotica*, 18(6):687–689, 2000.

[11] Martin Dietrich Buhmann. Radial basis functions. *Acta numerica*, 9:1–38, 2000.

[12] Unity Technologies. *Unity - Manual: Unity User Manual 2022.3 (LTS)*, 2022. Software.

[13] Unreal Engine. *Unreal Engine 5.4 Documentation | Unreal Engine 5.4 Documentation*, 2024. Software.

[14] COMSOL: Multiphysics software for optimizing designs, 2024.

[15] NVIDIA omniverse, 2024.

[16] Pixar. USD home — universal scene description 24.05 documentation, 2024.

[17] The Galaxy Community. The galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update. *Nucleic Acids Research*, 2024.

[18] John Chilton. galaxyproject/pulsar, 2024.

[19] Paul K. Romano, Nicholas E. Horelik, Bryan R. Herman, Adam G. Nelson, Benoit Forget, and Kord Smith. OpenMC: A state-of-the-art monte carlo code for research and development. *Annals of Nuclear Energy*, 82:90–97, 2015.

[20] NVIDIA Modulus Team. NVIDIA modulus, 2024.

[21] Pieter Boom, Andrey Jivkov, and Lee Margetts. ParaGEMS: Integrating discrete exterior calculus (DEC) into ParaFEM for geometric analysis of solid mechanics. *SoftwareX*, 21, 2023.

[22] NVIDIA-omniverse/kit-extension-template. original-date: 2021-09-09T18:03:38Z.

[23] Clare Sloggett, Nuwan Goonasekera, and Enis Afgan. BioBlend: automating pipeline analyses within galaxy and CloudMan. *Bioinformatics*, 29(13):1685–1686, 2013.

[24] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 239:2:2, 2014.

[25] J. E. Menard. Compact steady-state tokamak performance dependence on magnet and core physics limits. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 377(2141), 2019. Publisher: Royal Society.

[26] Joe Milnes, Nizar Ben Ayed, Fahim Dhalla, Geoff Fishpool, John Hill, Ioannis Katramados, Richard Martin, Graham Naylor, Tom O'Gorman, and Rory Scan-

nell. MAST upgrade – construction status. *Fusion Engineering and Design*, 96-97:42–47, 2015.

[27] Clement Fausser, Antonella Li Puma, Franck Gabriel, and Rosaria Villari. Tokamak D-T neutron source models for different plasma physics confinement modes. *Fusion Engineering and Design*, 87(5):787–792, August 2012.

[28] Jonathan Shimwell, John Billingsley, Rémi Delaporte-Mathurin, Declan Morbey, Matthew Bluteau, Patrick Shriwise, and Andrew Davis. The paramak: automated parametric geometry construction for fusion reactor designs. *F1000Research*, 10:27, 2021.

[29] Fande Kong and Paul W. Humrickhouse. Toward a Fully Integrated Multiphysics Simulation Framework for Fusion Blanket Design. *IEEE Transactions on Plasma Science*, 50(11):4446–4452, 2022.

[30] Dassault Systèmes Simulia Corp. *Abaqus 2020 Documentation*. Dassault Systèmes, 2020. Software.