ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB024

EFFICIENT COUPLING OF HYDRODYNAMIC AND ENERGY-DEPOSITION CODES FOR HYDRODYNAMIC-TUNNELLING STUDIES ON HIGH-ENERGY PARTICLE ACCELERATORS

C. Wiesner*, F. Carra, J. Kruse-Hansen, M. Masci, D. Wollmann, CERN, Geneva, Switzerland
Y. Nie, KIT, Karlsruhe, Germany

Abstract

The machine-protection evaluation of high-energy accelerators comprises the study of beyond-design failures, including the direct beam impact onto machine elements. In case of a direct impact, the nominal beam of the Large Hadron Collider (LHC) would penetrate more than 30 meters into a solid copper target. The penetration depth due to the time structure of the particle beam is, thus, significantly longer than predicted from purely static energy-deposition simulations with 7 TeV protons. This effect, known as hydrodynamic tunnelling, is caused by the beam-induced density depletion of the material at the target axis, which allows subsequent bunches to penetrate deeper into the target. Its proper simulation requires, therefore, to sequentially couple an energy-deposition code and a hydrodynamic code for the different target densities.

This paper describes a method to efficiently couple the simulations codes Autodyn and FLUKA based on automatic density assignment and input file generation, and presents the results achieved for a sample case.

INTRODUCTION

The design and assessment of machine-protection systems for existing and future high-energy accelerators includes the study of accidental beam impact on machine elements. In case of the impact of a large number of particle bunches in one location, the energy deposited by the first bunches can cause a rapid temperature rise, leading to a radial pressure wave that depletes the target density along the beam path [1, 2]. This allows the following bunches and the produced particle shower to penetrate deeper into the target. This effect, known as hydrodynamic tunnelling, can lead to a significant increase of the damage range in the material. For example, the static range of a 7 TeV proton and its shower in solid copper is around 1 m, while the full nominal LHC beam is expected to penetrate up to 35 m into the target when the hydrodynamic effects are considered [1].

To account for the changing material density during the beam impact, the state-of-the-art solution consists in the sequential coupling of an energy-deposition code as FLUKA [3, 4] and a hydrodynamic code as Autodyn [5] or BIG2 [6]. This requires running the two codes in a loop, recomputing the energy deposition and target densities for each simulation step [1, 2]. It, thus, implies a time-consuming simulation procedure, typically taking several months for one case study.

* christoph.wiesner@cern.ch

The total run time is currently dominated by the energy-deposition code. However, in practical terms, the required manual interventions during each of the simulation steps are a critical bottleneck for future hydrodynamic-tunnelling studies. Therefore, a coupling script, written in Python, has been developed. It reads in the material densities from the Autodyn output file, interpolates and assigns them to the correct FLUKA regions, and then automatically generates a new FLUKA input file [7]. While these steps were performed half-manually in the previously established workflow, the new approach allows time-efficient coupling by minimising the required human action and reducing the probability of human errors.

WORKFLOW & SCRIPT STRUCTURE

A diagram of the required work flow for the coupling simulations is shown in Fig. 1.

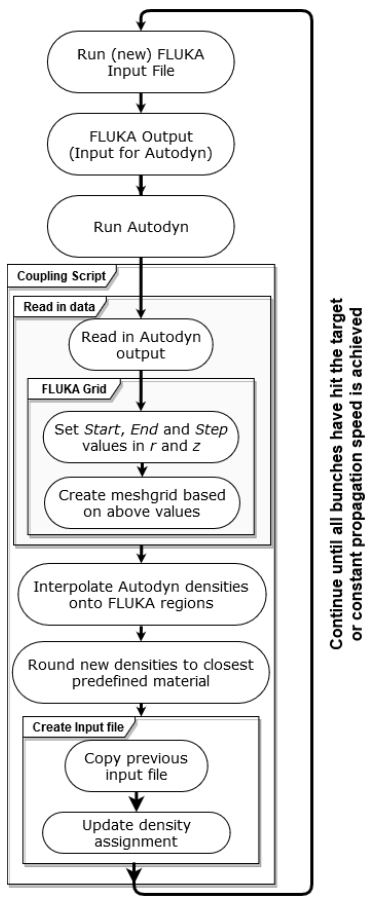

Figure 1: Work flow diagram for the coupling simulations.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB024

The new coupling script covers the steps from reading in the Autodyn output file to generating a new FLUKA input file with the updated material densities. The required input for the script includes a) a user-generated JSON input file, b) an Autodyn output file providing the material densities at given grid points, c) a FLUKA input file specifying the target geometry and the simulation parameters. The FLUKA input file has to be updated for each successive energy-deposition simulation with the new densities obtained from the corresponding Autodyn simulation.

The following main steps are taken in the script:

1. The desired input parameters are specified in a purpose made JSON file.
2. The Autodyn grid and densities are read in and post-processed to check for unphysical values and remove them, if necessary.
3. A grid, representing the FLUKA regions, is created based on the parameters specified in the JSON file.
4. The Autodyn densities are interpolated onto the centre of each FLUKA region.
5. It is verified that there are no densities with an undefined value (NaN) inside the grid domain. If any NaN values are found, they are interpolated with the nearest-neighbour method.
6. The densities at the centres of the grid regions are then rounded to the closest predefined density value. Here, the same approach as established in [2] is used: discrete density steps are predefined to reduce the total number of required FLUKA materials.
7. The new FLUKA input file is created by cloning the original input file and assigning the new densities to the FLUKA regions. Inside one FLUKA region, uniform density is assumed.

For the full simulation cycle, the updated input file is then run with FLUKA, and the resulting specific energy deposition is imported into Autodyn to simulate the impact of the following bunches. This cycle has to be repeated until all bunches have hit the target or the density-depletion front on the axis has reached a constant propagation speed, which allows for extrapolating the total tunnelling depth of the beam [1].

SAMPLE CASE

Overview

The coupling script was tested with sample data obtained during a previous study. In that study, the HiRadMat-12 experiment [8, 9] was simulated with FLUKA and Autodyn as described in [2]. During the experiment at CERN's HiRadMat facility [10], a copper target was impacted by 144 proton bunches with a bunch intensity of $1.5{\times}10^{11}$ at an energy of 440 GeV. The target comprised 15 cylinders with a 4 cm radius and a 10 cm length each, which were separated by 1 cm gaps. For the corresponding FLUKA simulations, the target was modelled as a solid copper cylinder with a radius of $r = 4$ cm and a length of $z = 1.5$ m. Fine regions were defined ranging from $r = 0$ mm to $r = 5$ mm and from $z = 0$ mm to $z = 1000$ mm. Cylindrical symmetry was assumed. Each fine region had a radial thickness of 0.1 mm and a length of 20 mm [2].

The impact of the proton bunches was successively simulated in Autodyn until the maximum density reduction along the target axis reached around 10 % to 15 %. Then, FLUKA was re-run with the updated density map. For the sample case, this was required for every 12^{th} bunch, i.e. requiring 12 FLUKA steps for the full impact. Simulation and measurement showed good agreement in terms of the melting depth of the target after the irradiation [2, 9]. To test the coupling script, the previously produced Autodyn output files from this study have been used.

Density Interpolation and Grid Data

The interpolation of the Autodyn density values onto the centres of the FLUKA regions is performed by using the `griddata()` function from the `interpolate` module of the SciPy library [11, 12].

Figure 2 shows the Autodyn grid points together with the FLUKA regions for the simulation step after the impact of 120 bunches from the sample study. As visible, in most FLUKA regions a high number of around 20 Autodyn points is reached. However, close to the beam axis, the distribution of the Autodyn points has changed compared to the original regular grid. This is due to the Langrangian mesh used for the simulations, in which the material flow is produced by the physical moving of the mesh elements or grid points. Consequently, the Autodyn point density in the critical area close to the beam axis is significantly reduced.

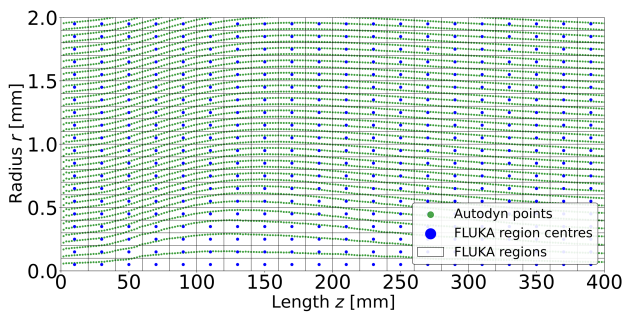

Figure 2: Autodyn grid points (green) for the simulation step after the impact of 120 bunches from the sample study, plotted on top of the FLUKA regions (black rectangles). The centres of the FLUKA regions are marked with blue dots.

This feature makes the density interpolation in these regions more challenging and more sensitive to different interpolation methods, with the result depending, e.g., on how many surrounding data points are considered for the interpolation. Therefore, it will be re-assessed for future studies how the Autodyn mesh can be refined in the critical area

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB024

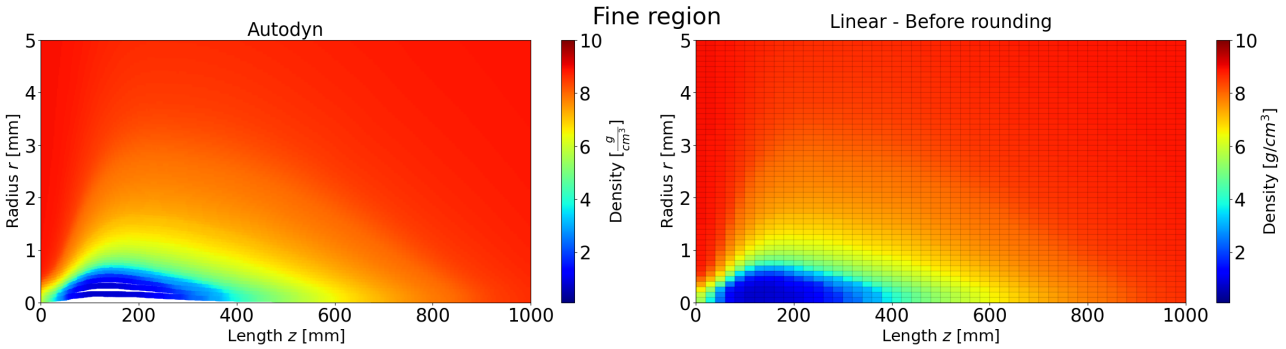

Figure 3: Comparison of the copper density as read in from the Autodyn file (left) and after assignment to the FLUKA regions (right) using a linear density interpolation. The data is taken from the sample study for the simulation step after the impact of 132 bunches. Note that the white areas close to the target axis on the scatter plot (left) are a result of the lower point density in these regions due to the Lagrangian mesh.

close to the target axis, or if an Eulerian mesh could be more beneficial.

Density Distributions

Figure 3 compares the densities as read in directly from the Autodyn output (left side) and the interpolated densities after the assignment to the FLUKA regions (right side). Overall, the shape of the density distribution is well preserved after the interpolation process. However, the structure of the FLUKA regions with their constant density is well visible.

Comparison of Interpolation Methods

Based on the `scipy.interpolate.griddata()` function, the script currently supports three different interpolation algorithms [13]:

1. Nearest: Nearest-neighbour algorithm, which returns the value of the data point that is nearest to the current point of interpolation.
2. Linear: The input data is triangulated using Delaunay triangulation. After the triangulation a linear barycentric interpolation is performed on each triangle.
3. Cubic: The input data is triangulated, after which a piecewise cubic interpolating Bezier polynomial is applied on each triangle.

Figure 4 compares the resulting densities for the three algorithms. For reference, the original density from the Autodyn output file is plotted. The radial density evolution is shown at a longitudinal position of $z = 210$ mm.

In general, all three algorithms provide similar interpolation results and approximate the original Autodyn values well. However, the density steps at the boundary of the FLUKA regions are clearly visible. Not surprisingly, the largest differences between the three methods are visible in the range between $r \approx 0.4$ mm and $r \approx 1$ mm, where the density gradient is largest. A more accurate representation would, thus, require the reduction of the radial width of the FLUKA regions for this area, implying an increased simulation time.

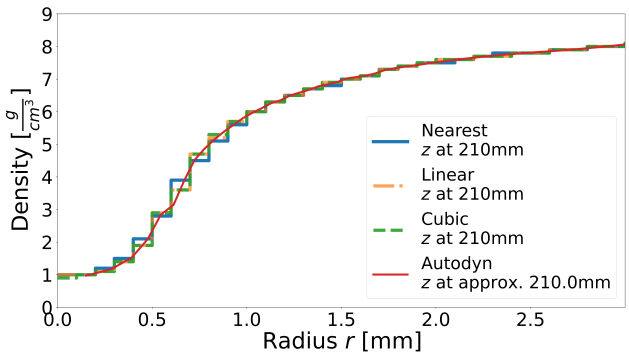

Figure 4: Comparison between the Autodyn density after the impact of 132 bunches and the nearest-neighbour, linear and cubic interpolation algorithms for $z = 210$ mm.

CONCLUSIONS

The simulation of hydrodynamic-tunnelling effects requires the sequential coupling of an energy-deposition code, as FLUKA, and a hydrodynamic code, as Autodyn. In order to increase the coupling efficiency between the two codes, a script has been developed that automatically reads in the material densities from an Autodyn output file, assigns them to pre-defined FLUKA regions, and then generates a new FLUKA input file using the updated densities. One can choose between three different density interpolation methods (nearest neighbour, linear and cubic). The script will be used for future studies simulating the impact of high-energy beams on accelerator materials. It minimises the required human action and provides the basis to further automatise the coupling of the two codes, with the long-term goal of achieving a fully automatised coupling.

ACKNOWLEDGEMENTS

The authors would like to thank R. Schmidt and N. Tahir for providing helpful insights on the physics of hydrodynamic tunnelling, as well as M. Maciejewski and C. Hernalsteens for their help in reviewing the Python code.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-MOPAB024

REFERENCES

[1] N. Tahir, F. Burkart, R. Schmidt, A. Shutov, and A. Piriz, "Review of hydrodynamic tunneling issues in high power particle accelerators", *Nucl. Instrum. Methods Phys. Res., Sect. B*, vol. 427, pp. 70–86, 2018. doi:10.1016/j.nimb.2018.04.009

[2] Y. Nie *et al.*, "Simulation of hydrodynamic tunneling induced by high-energy proton beam in copper by coupling computer codes", *Phys. Rev. Accel. Beams*, vol. 22, p. 014501, 2019. doi:10.1103/PhysRevAccelBeams.22.014501

[3] T. T. Böhlen *et al.*, "The FLUKA Code: Developments and Challenges for High Energy and Medical Applications", *Nucl. Data Sheets*, vol. 120, pp. 211–214, 2014. doi:10.1016/j.nds.2014.07.049

[4] A. Ferrari, P. R. Sala, A. Fassò, and J. Ranft, "FLUKA: A multi-particle transport code (program version 2005)", CERN, Geneva, Switzerland, Rep. CERN-2005-010, 2005.

[5] Ansys autodyn, https://www.ansys.com/products/structures/ansys-autodyn

[6] V. E. Fortov, B. Goel, C.-D. Munz, A. L. Ni, A. V. Shutov, and O. Y. Vorobiev, "Numerical Simulations of Nonstationary Fronts and Interfaces by the Godunov Method in Moving Grids", *Nucl. Sci. Eng.*, vol. 123, no. 2, pp. 169–189, 1996. doi:10.13182/NSE96-A24181

[7] J. Kruse-Hansen and C. Wiesner, "Automatic Density Assignment and Generation of FLUKA Input Files for Hydrodynamic-Tunnelling Studies", CERN, Geneva, Switzerland, Rep. EDMS 2579002, Apr. 2021. https://edms.cern.ch/document/2579002

[8] J. B. Sancho, "Machine Protection and High Energy Density States in Matter for High Energy Hadron Accelerators", Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2014.

[9] F. Burkart, "Expected damage to accelerator equipment due to the impact of the full LHC beam: beam instrumentation, experiments and simulations", Ph.D. Thesis, Goethe-Universität, Frankfurt am Main, Germany, 2016.

[10] I. Efthymiopoulos *et al.*, "HiRadMat: A New Irradiation Facility for Material Testing at CERN", CERN, Geneva, Switzerland, Rep. CERN-ATS-2011-232, Nov. 2011.

[11] ndgriddata, https://github.com/scipy/scipy/blob/v1.6.0/scipy/interpolate/ndgriddata.py

[12] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, vol. 17, pp. 261–272, 2020. doi:10.1038/s41592-019-0686-2

[13] Scipy.interpolate.griddata, https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html