DOI: 10.61782/fa.2025.0900

FORUM ACUSTICUM EURONOISE 2025

TOWARDS A COMMUNITY HUB FOR OPEN-SOURCE ROOM ACOUSTICS SOFTWARE

Silvin Willemsen[1*] **Hassan Teymoori**[2] **Ilaria Fichera**[1] **Huiqing Wang**[1]
Alexander Nolte[2] **Alexander Serebrenik**[2] **Maarten Hornikx**[1]
[1] Dept. of the Built Environment, Eindhoven University of Technology, The Netherlands
[2] Dept. of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

ABSTRACT

Open-source software plays an important role in the advancement of the development, intercomparison, and accessibility of novel and improved room acoustics simulation methods. Although the amount of open-source software in room acoustics has increased over the years, it usually remains less well-maintained and less user-friendly than their proprietary counterparts. Typically, open-source room acoustics simulation software exists as code repositories rather than ready-to-use applications and, therefore, requires technical know-how to be built into usable software. In this work, the Community Hub for Open-source Room Acoustics Software (CHORAS) is presented: a web-based user interface that provides an open-source community platform for room acoustic simulation methods. This platform acts as the front-end for back-end software that can be developed remotely. In this way, any researcher and/or institute can contribute to the platform by integrating their simulation software into the back-end of the application. Currently, users can select the diffusion equation (DE) and discontinuous Galerkin (DG) method to perform room acoustic simulations. Multiple simulations can be run consecutively, allowing for comparative analysis of results. A preliminary user evaluation indicates that CHORAS is intuitive to use, but improvements can be made.

Keywords: *room acoustics, open source software, open science*

**Corresponding author*: *s.willemsen@tue.nl.*

1. INTRODUCTION

The field of room acoustics simulation methods spans more than half a century, and during this time many different software have been created. However, the vast majority of these software is not available to the public [1]. Although there may be a lack of incentive for researchers to share their research software, open-source software can potentially lead to acceleration of science, increased impact, improved reproducibility code quality and sustainability, and a tighter-knit community.

Examples of open-source room acoustics software are Pyroomacoustics [2], RIR-Generator [3], gpuRIR [4], and SoundSpaces [5]. For an overview of the state of open source room acoustics software as of 2024, see Hornikx et al. [1]. A major issue with many of these software systems is that they are made available as code repositories rather than as ready-to-use software applications. Therefore, many systems require technical know-how to build them, and this might scare off potential users who do not have the required skills. Furthermore, as each software has their own pre- and postprocessing pipelines, this makes their simulation results difficult to compare.

In this work, we present the Community Hub for Open-source Room Acoustics Software (CHORAS), a web-based platform for room acoustics simulations. CHORAS is built using a modern web application stack designed for scalability, testability, and maintainability [6]. Furthermore, it has been created with cloud deployment in mind, so that users will not have to install the software locally or run heavy acoustics simulations on their own machines. The back-end of the platform is written in Python reducing the entrance barrier for potential future contributors.

The main contribution of CHORAS to the room acoustics community is a platform for researchers and developers to add their own room acoustics simulation back-end software and use CHORAS as the front-end. Moreover, CHORAS allows for comparison of different simulation methods, as the pre- and postprocessing stages of the simulation are identical. The idea of allowing researchers to add their own simulation techniques is shared by CHORAS and i-Simpa [7]. However, i-Simpa requires local installation, and cannot benefit from the advantages that current-day cloud computing provides.

This paper is also call to the room acoustics community to contribute to CHORAS. This can be done either by contributing to CHORAS itself (front-end / back-end), or by adding one's own room acoustics simulation back-end to the platform. The CHORAS repository is available on GitHub [8] and includes instructions on how to contribute to the platform.

The remainder of this paper is structured as follows. Section 2 describes the software architecture of CHORAS and Section 3 presents the various functionalities of the platform. A user-evaluation and results thereof are presented in Section 4. Finally, concluding remarks and future work are presented in Section 5.

2. SOFTWARE ARCHITECTURE

This section gives a high-level overview of CHORAS's software architecture (see Figure 1). Low-level details can be found in [6].

2.1 Overview

CHORAS consists of a web-based front-end, a Python-based back-end, and Python-based acoustics simulation modules. The front-end is built using *React* [1], a widely-used JavaScript library for building user interfaces for the web. To enhance type safety and catch programming errors earlier in the development process, this project uses *TypeScript* rather than JavaScript.

The back-end was developed using *Flask* [2], a lightweight and modular Python web framework. The back-end also handles the communication to a database which stores and manages simulation configurations, such as source and receiver locations and 3D geometries, and simulation results. *PostgreSQL* [3] was chosen as a database, and SQLAlchemy [4] as an Object-Relational Mapper (ORM). The back-end also communicates to the various simulation modules, which are described in more detail below. Given that acoustic simulations can be time consuming, it was imperative to implement an asynchronous processing mechanism that prevents system responsiveness from being compromised. To this end, Celery [5], a distributed task queue was used, enabling the offloading of long-running computations to background workers, ensuring that the primary system remains responsive.

2.2 Simulation Back-end

The simulation back-end is the part of CHORAS which consists of various simulation methods. Here, the room acoustics community can contribute most with their own room acoustics simulation software. Naturally, any software added to CHORAS should be open source itself. The modules currently available are the diffusion equation module [6] [9] and the discontinuous Galerkin module [7] [10] and are written in Python. Both modules use the open source meshing tool Gmsh [11] to convert the geometry to a mesh (.msh format) which can be used for the simulation.

The communication between CHORAS's back-end and the various simulation modules happens via JSON files. The relevant settings (such as source and receiver positions, geometry, simulation length, etc.) are loaded from the database into a (new) JSON file, of which the path is sent to the relevant simulation module. The module will be able to use the settings to perform a room acoustics simulation. The results of a simulation are written to the same JSON file, which can ultimately be fetched by CHORAS's front-end and be visualised.

As mentioned in Section 1, members of the community can add their own room acoustics simulation back-end to CHORAS. Although each individual simulation back-end could have a method-specific parameters, all should allow for the following inputs:

- a single source and receiver location (but preferably multiple),

[1] https://react.dev/
[2] https://flask.palletsprojects.com/
[3] https://www.postgresql.org/
[4] https://www.sqlalchemy.org/
[5] https://docs.celeryq.dev/
[6] https://github.com/Building-acoustics-TU-Eindhoven/Diffusion/
[7] https://github.com/Building-acoustics-TU-Eindhoven/edg-acoustics

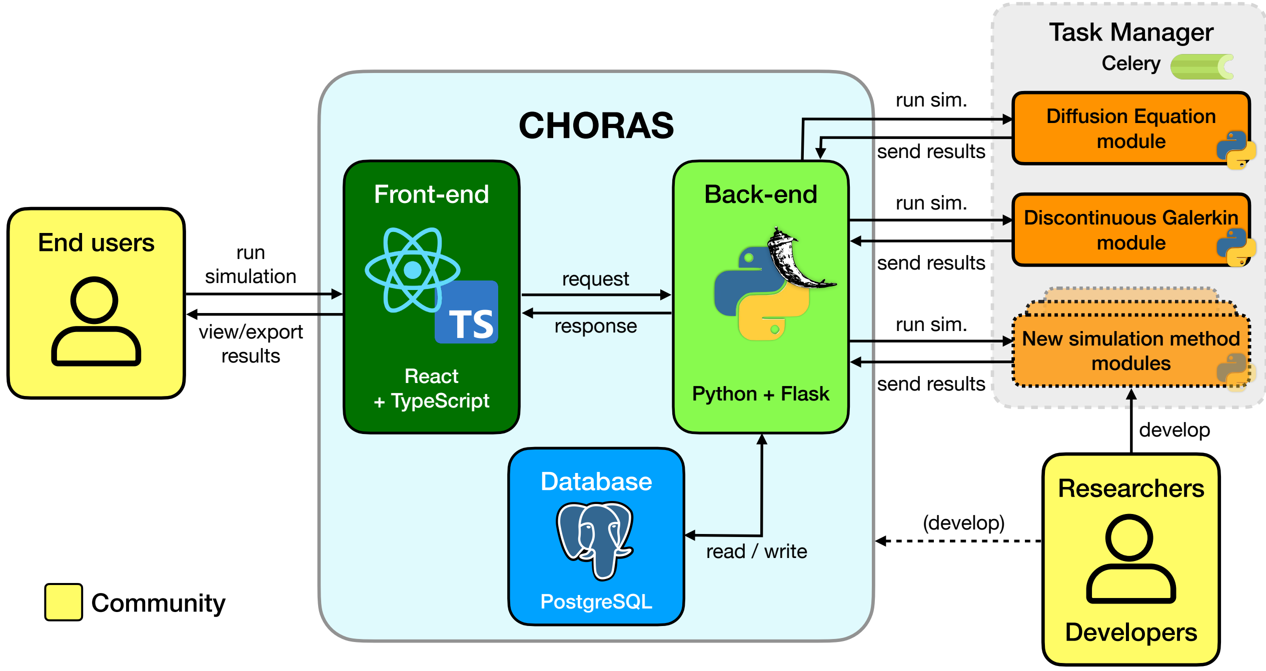


Figure 1. CHORAS's software architecture with the community (end users, researchers, and developers) highlighted in yellow. Researchers and developers can easily contribute to CHORAS by adding their own simulation methods, as well as by potentially helping to develop the platform itself. See Section 2 for a detailed description.

- geometry input (.geo / .msh / .obj),
- material absorption coefficients, and
- a path to the JSON file as an argument, so that the simulation progress can be updated while the simulation is running.

For parameters unique to the simulation method, researchers will be able to easily add this to CHORAS using a custom JSON file (see Section 3.2.3 for more information).

3. FUNCTIONALITY

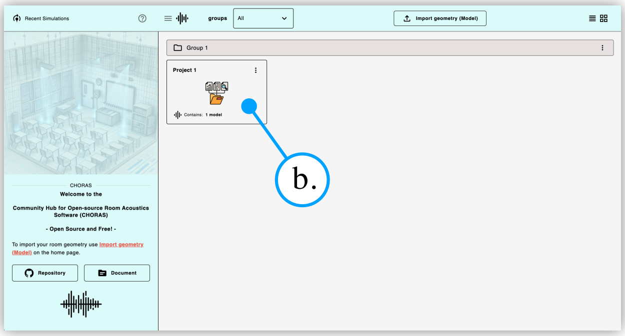
CHORAS has been designed to perform acoustic simulations while providing end users with accurate and reliable results. Additionally, the front-end has been developed with a strong emphasis on ease of use. This section presents CHORAS's functionality using a typical workflow using Figure 2 as a reference.

3.1 Front page

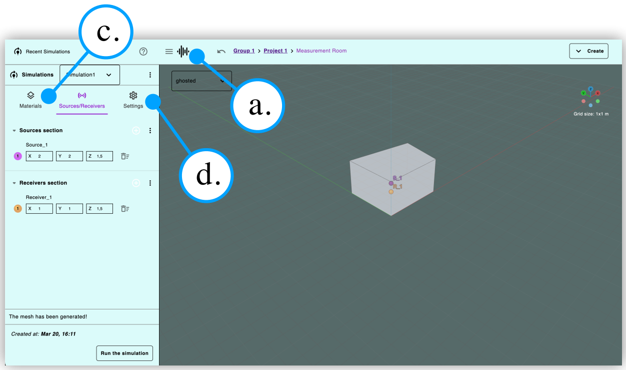
The project hierarchy in CHORAS is structured as follows: *Group* → *Project* → *Model* → *Simulation*. In other words, each *Group* can contain multiple *Projects*, each *Project* can contain multiple (3D) *Models*, and each *Model* can have several *Simulations* (with different configurations) performed on it. CHORAS's front page shows the two top levels of this hierarchy, i.e., the *Groups* and the *Projects* (see Figure 2a). By clicking the *Import Geometry* button at the top of the front page, a user can upload a 3D geometry file (currently only .obj and .dxf are supported) and assign it to a *Group* and *Project*. They will then be able to navigate to the Simulation page.

3.2 Simulation page

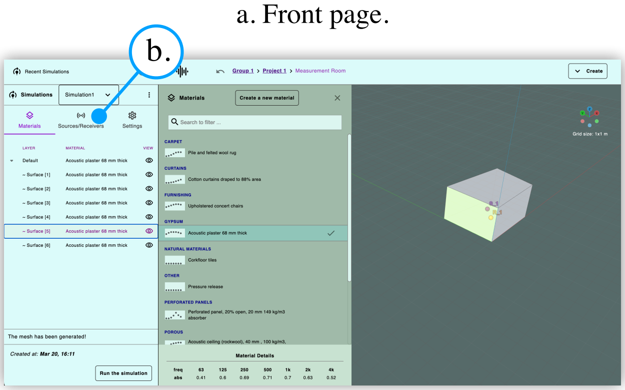
The Simulation page has two main components: a viewport showing the 3D model and a toolbar for configuration of the simulation settings (see Figures 2b-d). The 3D geometry (.obj / .dxf) file uploaded in the previous step is converted to .3dm by CHORAS's back-end, which is used



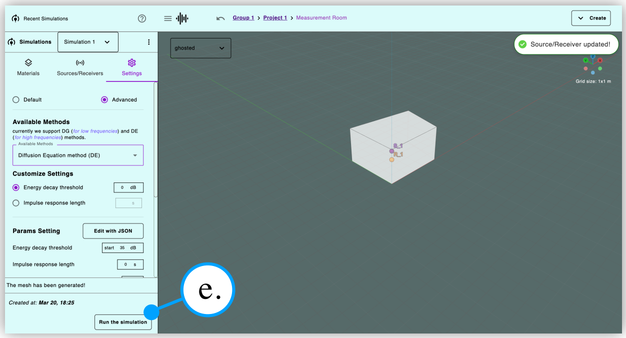
a. Front page.



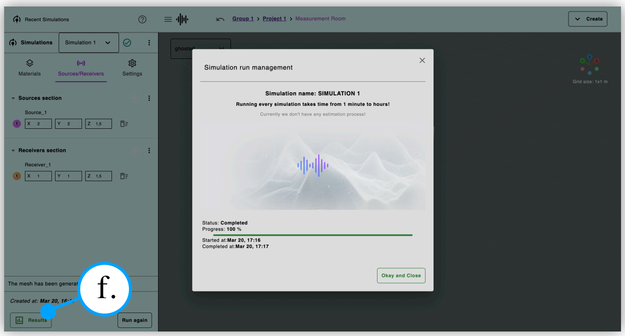
b. Simulation page (Sources/Receivers tab).



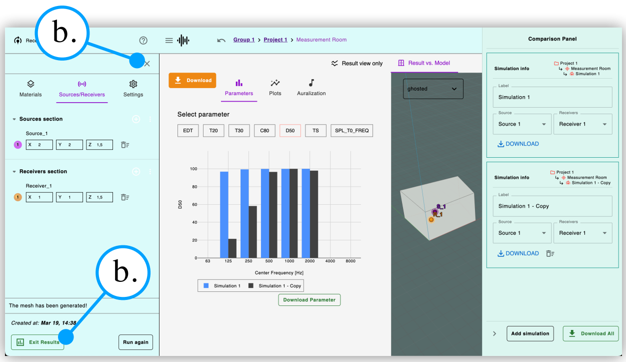
c. Simulation page (Materials tab).



d. Simulation page (Settings tab).



e. Simulation run pop-up.



f. Results page.

Figure 2. A summary of CHORAS's functionality. The bubbles show how to navigate to a different view shown in that respective subfigure 2.(a-f). See Section 3 for a detailed description.

to visualise the geometry the viewport. The following subsections describe the functionality of the three tabs in the toolbar in detail.

3.2.1 *Sources/Receivers*

The Sources/Receivers tab allows users to add sources and receivers to the simulation. See Figure 2b. Currently, only a single source or receiver is available, which will be extended to multiple sources and receivers. Users can edit the locations of sources or receivers using the textboxes in the toolbar, or (after clicking on a source or receiver in the toolbar) using the arrows in the 3D model. Finally, if a source or receiver is outside the geometry volume, it will be coloured red.

3.2.2 *Materials*

The Materials tab allows users to assign various materials to the different surfaces in the model. See Figure 2c. Selecting a surface in the dropdown will highlight this surface in the 3D model and will trigger a pane listing all available materials. Absorption coefficients for octave bands from 63Hz-4kHz are shown at the bottom of the pane and are graphically visualised next to the material names for convenience.

3.2.3 *Settings*

In the Settings tab, when a user selects *Advanced*, they can select which simulation method they would like to use (currently DE and DG). See Figure 2d. The *Edit Settings* section is dynamically generated based on the selected simulation method. This is a valuable feature, as this allows researchers and developers to easily expose their desired simulation parameters without having to write the front-end code for it. Apart from the simulation back-end, all that is needed is a JSON file listing the parameters needed for the simulation back-end. See Algorithm 1 for an example that allows the front-end to generate one text box and one slider that control the energy decay threshold and the length of the impulse response, respectively. The minimum, maximum and default values as well as the stepsize can be determined. The back-end will then be able to read the input values and use these settings for the simulation configuration.

3.3 Running a simulation

When a user is done with the configuration described above, they can click on the *Run the simulation* button at the bottom of the toolbar. If something is wrong with the configuration, such as a missing source or receiver, or a surface not having a material assigned, a notification will show on the top right informing the user how to fix the mistake. Otherwise, a pop-up will appear through which the user can start the simulation (see Figure 2e). The pop-up will show the progress of the simulation in real time, and will allow for the user to cancel the simulation as it is being performed. The pop-up can be closed so that the user can set up a new simulation if desired. The progress can still be followed at the top left of the page.

```
{
  "type":  "deSettings",
  "options":  [
  {
    "name":  "Energy decay threshold",
    "type":  "integer",
    "display":  "text",
    "min":  10,
    "max":  60,
    "default":  35,
    "step":  1
  },
  {
    "name":  "Impulse Response length",
    "type":  "float",
    "display":  "slider",
    "min":  0,
    "max":  20,
    "default":  1,
    "step":  0.1,
  }
 ]
}
```

Algorithm 1: Example of a settings file determining the parameter layout in the front-end for the DE simulation back-end (See Figure 2d and Section 3.2.3.)

3.4 Results page

Once the simulation is done, a *Results* button will show at the bottom left of the toolbar. The results of several simulations can be compared and can be exported to .xlsx and .csv (ee Figure 2f). The results page has several tabs which are described in detail below.

3.4.1 *Parameters*

The Parameters tab shows the results of the following acoustic parameters (in octave bands from 63Hz-8kHz): early decay time (EDT), reverberation time (both T_{20} and T_{30}), clarity (C_{80}), definition (D_{50}), center time (T_s) and sound pressure level (SPL).

3.4.2 *Plots*

Currently, the Plots tab only visualises the energy decay curves (relevant to DE). In the future, this tab could show different types of plots, such as frequency responses.

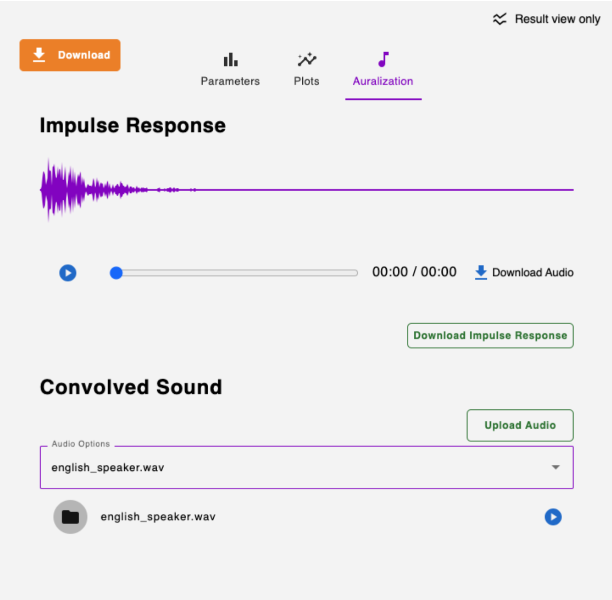


Figure 3. The auralization tab of the results page. The impulse response generated by the simulation back-end is visualised and can be auralized (see Section 3).

3.4.3 Auralization

The Auralization tab allows users to view and listen to the generated impulse response. See Figure 3. In addition, users will be able to upload sound files that will be convolved with the (resampled) impulse response at the sample rate of the input file. Both the impulse response and the convolved audio can be downloaded.

3.4.4 Results comparison panel

CHORAS features a results comparison panel with which users can compare results (Parameters and Plots) of several simulations side by side. See Figure 2f. In the future, the authors would like the auralization tab to also include functionality to easily compare between simulations.

4. EVALUATION

To assess the state of CHORAS, a workshop was organised with the platform. The main goal of the workshop was to assess CHORAS's:

- *usability*, i.e., the ease of use, and
- *usefulness*, i.e., how useful the software could be for the participant.

In total, six people participated in the workshop, among them acoustic consultants, acoustic researchers, and software engineers. This workshop was mainly to see whether the development of CHORAS was progressing in a desirable direction, as well as investigating what the priorities for future development should be. It is important to note that the version that was tested [8] neither included the auralization tab (presented in Section 3.4.3) nor the ability to export to .xlsx and .csv (see Section 3.4).

The rest of this section describes the methods and results of the evaluation, as well as a discussion of the results.

4.1 Methods

Three laptops were prepared with CHORAS installed (locally). The participants were asked to follow predefined steps in a document [9] in pairs. A walk-through video of the workshop can also be found online [10]. In summary, the participants had to run several simulations using CHORAS, explore the results, and play with the platform freely.

As the number of participants was low, it was chosen to use a variety of evaluation methods. First, some of the authors of this work observed the participants as they progressed through the workshop document. Participants were asked (and reminded) to think aloud [12]. After this, participants were asked to fill out a (slightly) modified version of the System Usability Scale (SUS) [11] [13], followed by a semi-structured interview focusing on CHORAS's usefulness, i.e., whether the participant would use CHORAS in their professional workflow. The workshop ended with a Mentimeter questionnaire [12] in which they were asked about their overall experience using CHORAS, what they liked about CHORAS, and what they thought could be better.

4.2 Results and Discussion

Due to the small number of participants, the results in this section should only be interpreted as preliminary. How-

[8] https://github.com/Building-acoustics-TU-Eindhoven/CHORAS/releases/tag/NWOworkshop
[9] https://archive.org/details/workshop_202503
[10] https://youtu.be/fIsPHmw7jaY
[11] https://forms.office.com/e/Ea6N5Dq8Nk
[12] https://www.mentimeter.com/

ever, the findings are still considered to be meaningful and can be used to determine what future steps to take.

Using the formula in [13], the final average SUS score was found to be 75.4/100, which translates to a 'good' or 'acceptable' grade. This gives an indication that the current version of CHORAS is already fairly usable.

One of the main results obtained from the observations and semi-structured interviews was that many participants mentioned that they would like to listen to the impulse response after a simulation had finished. As mentioned at the start of Section 4, this feature had not yet been implemented in the version available during the user test, but is available in the current version (also see Section 3.4.3). Other suggestions include the addition of colours in the 3D model to indicate the various materials, the removal of the *Group* level in the project hierarchy (only the *Project* level would suffice; see Section 3.1), and including a progress indication for the meshing procedure, as this can take a long time as well.

The Mentimeter questionnaire indicated that participants rated the overall experience with CHORAS as 7.3/10, which is fairly consistent with the results obtained from the SUS. The answers to the question what people liked about CHORAS included the word "intuitive" several times, as well as "all-in-one platform" and "comparing results of simulations". What participants thought could be better included "the design", "not being able to cancel the simulation," and "material visualisation" (referring to colours denoting various materials in the 3D model as mentioned above).

Overall, the authors are pleased with how the participants interacted with and rated CHORAS. With a score of 75.4/100 on the SUS and the fact that participants found CHORAS intuitive to use indicates that the development of the platform is on the right track.

5. CONCLUSION AND FUTURE WORK

This work presents CHORAS, a web-based open-source platform for room acoustics simulations. Currently, two existing acoustic simulation methods – the diffusion equation method [9] and the discontinuous Galerkin method [10] – are coupled to the platform. CHORAS allows other researchers and developers to add their own methods to the simulation back-end, and provides an easy way to expose method-specific simulation parameters. Information on the software architecture and functionality of CHORAS has been provided, as well as the results of a user evaluation.

5.1 Future work

The first step in the future development of CHORAS is its deployment to the cloud. Currently, users and developers need to go through a lengthy installation procedure and need to run three separate processes (React front-end + Python back-end + the Celery task-queueing system) to use the platform. Ultimately, CHORAS should be fully cloud-based, such that users can visit a website, log in, and run the simulations remotely. Developments to Dockerize [13] the application are in progress and are expected to be finalised soon. Cloud deployment might also open up the possibility to run simulations in parallel, because as mentioned in Section 3, simulations can only be run consecutively.

Furthermore, the coupling to the DG simulation back-end does not properly include frequency-dependent material impedances yet, which will be added as soon as possible. Looking towards the future, the authors intend to link other open-source simulation software to CHORAS, if the license of the respective software allows it, such as Pyroomacoustics [2] and other software. This offers more types of simulation methods to users and fosters a community.

In addition to the above, many small improvements can be made to CHORAS. These include, but are not limited to, the possibility to include multiple sources and receivers, to improve plotting functionality, to cancel running simulations, to show the progress for meshing, and to improve material handling (grouping surfaces into layers and colouring by material). Finally, the authors acknowledge that the CHORAS user interface is heavily inspired by the Treble software [14]. Once the current functionality is stable, efforts will be made to overhaul the design to be more original and suited to our needs.

5.2 Our vision: A stronger community

The main reason that CHORAS was initially conceived, was to build a community around open-source room acoustic software. The authors are convinced that a community around CHORAS, consisting of academic researchers, software developers, and end users, will accelerate innovation in the field of room acoustics software and will be beneficial to all parties.

To kick-start this, the authors plan to organise several workshops and hackathons surrounding CHORAS in the

[13] https://www.docker.com/
[14] https://www.treble.tech/

near future. These events could greatly improve the visibility of this project, and thereby attract those who are interested in joining the community. To reiterate what has been mentioned in Section 1, this paper invites the room acoustics community to contribute to CHORAS, either to the platform itself, or by coupling their own simulation methods. The repository is available on GitHub [8] and includes guidelines on how to contribute.

6. ACKNOWLEDGMENTS

This project is supported by the Dutch Research Council (NWO), grant No. 19430. The authors thank EngD trainees Shatabdi Barua, Baigalmaa Bayarsaikhan, Chao Fu, Almas Hilman Muhtadi, Hary Dhimas Prakoso, and Yuhang Tian for making valuable additions to CHORAS.

7. REFERENCES

[1] M. Hornikx, H. Wang, I. Fichera, L. Paganini, A. Nolte, and A. Serebrenik, "Exploring the current landscape of open research software in room acoustics," in *Proc. of 53rd Int. Congress & Exposition on Noise Control Engineering, inter.noise24*, pp. 5875–5884, 2024.

[2] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE int. conf. on acoustics, speech and signal processing (ICASSP)*, pp. 351–355, 2018.

[3] E. A. Habets, "Room impulse response generator," tech. rep., Technische Universiteit Eindhoven, 2006.

[4] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "gpuRIR: A python library for room impulse response simulation with GPU acceleration," *Multimedia Tools and Applications*, vol. 80, pp. 5653–5671, 2021.

[5] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *Computer Vision–ECCV 2020: 16th European Conference*, pp. 17–36, 2020.

[6] H. Teymoori, "Design and evaluation of a user interface for room acoustic software," tech. rep., Eindhoven University of Technology, 2024.

[7] J. Picaut and N. Fortin, "I-Simpa, a graphical user interface devoted to host 3D sound propagation numerical codes," in *Acoustics 2012*, 2012.

[8] Building acoustics group at TU Eindhoven, "CHORAS." Available at https://github.com/Building-acoustics-TU-Eindhoven/CHORAS, accessed Mar 31st, 2025.

[9] I. Fichera, C. Van hoorickx, and M. C. J. Hornikx, "Determination of a diffusion coefficient function for long rooms using a least square optimization approach," in *Proc. of 53rd Int. Congress & Exposition on Noise Control Engineering, inter.noise24*, 2024.

[10] H. Wang, A. P. da Silva Clérigo, and M. C. J. Hornikx, "An open-source time-domain wave-based room acoustic software in python based on the nodal discontinuous Galerkin method," in *Proc. of 53rd Int. Congress & Exposition on Noise Control Engineering, inter.noise24*, 2024.

[11] C. Geuzaine and J.-F. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *Int. Journal for Numerical Methods in Engineering*, vol. 79, pp. 1039–1331, 2009.

[12] M. Someren, Y. Barnard, and J. Sandberg, *The Think Aloud Method - A Practical Guide to Modelling Cognitive Processes*. London: Academic Press, 1994.

[13] J. Brooke, *Usability Evaluation in Industry*, ch. SUS – a quick and dirty usability scale. Taylor & Francis, 1996.