

# A computer program for lifecycle flooding risk assessment according to the FLARE Framework process

Romain Michalec, Francesco Mauro, Donald Paterson,  
Mujeeb Ahmed Mughadar Palliparambil, Evangelos Boulougouris, Dracos Vassalos  
*Maritime Safety Research Centre, University of Strathclyde, Glasgow, Scotland, UK*

[romain.michalec@strath.ac.uk](mailto:romain.michalec@strath.ac.uk), [francesco.mauro@strath.ac.uk](mailto:francesco.mauro@strath.ac.uk),  
[d.paterson@strath.ac.uk](mailto:d.paterson@strath.ac.uk), [mujeeb.mughadar-palliparambil@strath.ac.uk](mailto:mujeeb.mughadar-palliparambil@strath.ac.uk),  
[evangelos.boulougouris@strath.ac.uk](mailto:evangelos.boulougouris@strath.ac.uk), [d.vassalos@strath.ac.uk](mailto:d.vassalos@strath.ac.uk)

## ABSTRACT

We present a computer program designed to assist and guide its user through the process of conducting a comprehensive and quantitative assessment of ship vulnerability to flooding, with consideration to the full lifecycle of the ship, that is to say from its design phase to its operational phase to the emergency response phase. The program implements a flooding risk assessment process known as the FLARE Framework, which is described in another paper at this conference.

This paper covers installation and usage of the program, with a focus on its input and on the flow of information between it and the external assessment programs it works with. We describe the expected ship geometry input and damage information input, explain the different calculation stages, and show how to import, display, filter, and plot the results of those stages to gain insights into the survivability of a particular ship design.

This demonstrates the FLARE Framework process in a concrete manner, and shows that by using the software to follow the process, the user conducts a comprehensive and quantitative evaluation of ship design on the basis of flooding risk.

**Keywords:** *Risk assessment, flooding, methodology, software*

## 1. INTRODUCTION

### 1.1 From process to program

The FLARE Framework is a three-stage methodology for a comprehensive assessment of ship vulnerability to flooding over the vessel's lifecycle, including design, operation and emergency response phases. It involves using different software tools, catering for different determinants of survivability, at the different stages of the assessment process. It culminates in the identification of risk control options and quantitative risk measures (Vassalos et al., 2021).

To make it easy to follow this methodology, we are developing a computer program that implements it, that is to say that the program assists and guides its user through the assessment process.

The aim of the program is to organize the user's workflow through a common platform. It does this by providing a means to specify and access the pertinent software tools and to display, filter, plot, and exchange data with these tools, regarding ship design features, damage cases, stability, and evacuability. The result is that the process is properly followed, and also easier to understand for people who are new to it.

In this paper, we describe the program and how to use it. We go through the geometry input and damage input that the program expects, explain the different calculation stages, and show how to import and explore the results of those stages, thereby demonstrating the FLARE Framework process in a concrete manner.

However, we do not explain the methodology again. A paper by Vassalos et al. (2021) at this conference provides a comprehensive overview of the methodology. The present paper complements it.

The rest of this section gives a short general description of the software. Some basic information regarding installation and licensing is then provided. The article is then set out in the form of step-by-step guidance through the program, and therefore through the assessment process.

## 1.2 General description

The program, called FLARE for short, consists in a tabbed graphical interface whose successive tabs walk the user through the FLARE Framework process. It is illustrated in figure 1.

The first two tabs are concerned with the input of the geometrical model of the ship and the definition of the damage cases that will be explored through the process (a damage case is the combination of a breach, a loading condition, and a sea state). The next three tabs walk the user through the three numerical analyses that form the pillars of the assessment, namely the static damage stability analysis, the dynamic damage stability analysis, and the evacuability analysis.

The software does not perform those analyses itself. Instead, the user is free to carry out each analysis with the program they normally use for this task, such as NAPA or FREDYN. This ensures that the software remains a flexible platform that can be used with a variety of external assessment programs, which the program only links to rather than integrates or emulates.

The user can import the assessment results produced by each of the numerical analyses as

a sortable table into the FLARE program. They can then use the filtering options that the program provides to isolate damage cases of interest that are worth progressing to the following analysis.

Each of the three numerical analyses is indeed more computationally expensive and higher-fidelity in detail than the previous one. In order to facilitate a time and resource efficient process, it is therefore advantageous to rule out damage cases where there is nothing to be gained from further investigation, and focus instead on the marginal and severe cases, where a greater wealth of knowledge and information would work to the benefit of the ship designer.

## 2. TECHNICAL INFORMATION

### 2.1 Dependencies

FLARE is a Python application. It is structured as a Python package. It is developed and tested on Linux and Windows using Python 3.8.

The graphical user interface is built with the Qt 5 application framework, used through its official Python bindings, PySide2 (The Qt Company, 2020). Qt was chosen because it is a major, comprehensive, time-tested, industry-standard widget toolkit for creating graphical user interfaces and cross-platform applications, that is well-supported, sufficiently documented, and available under a suitable license.

The program also uses the Pandas (McKinney, 2010; Reback et al., 2021) and Matplotlib (Hunter, 2007; Caswell et al., 2021) libraries to manipulate data sets and plot two-dimensional data, respectively. Those dependencies were chosen because they are the state of the art in data manipulation and scientific plotting in Python. They feature ubiquitously in data science.

### 2.2 Installation

Like any Python package, FLARE can be installed and run from its source code on any system with a Python distribution. Linux systems include a Python distribution. Windows users

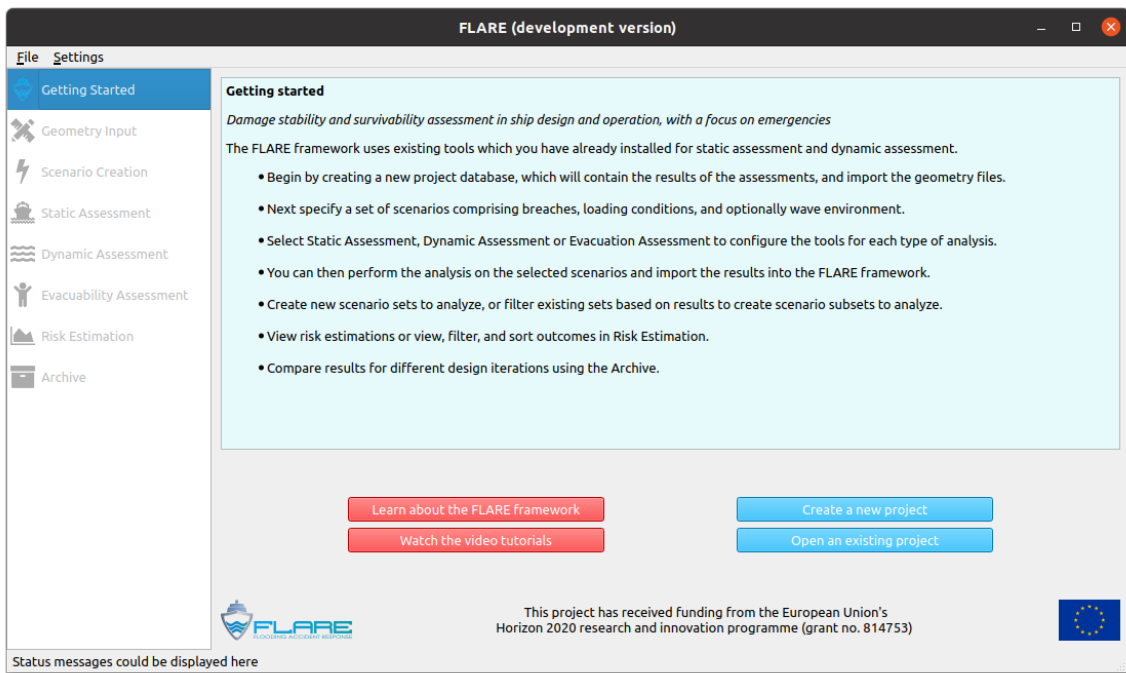


Figure 1: Start screen. The main steps through the FLARE process are visible as tabs on the left. The tabs are grayed out until a project database is created or loaded by clicking on one of the two blue buttons. Help and tutorials are available via the two red buttons.

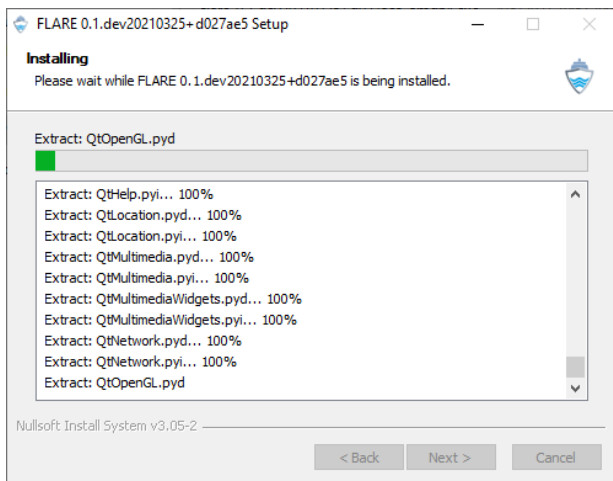


Figure 2: Installing FLARE with the graphical installer is easy and user-friendly

can download one from the Python website or the Microsoft Store.

The following command, run in the directory containing the source code, will download the dependencies of FLARE from the Python Package Index and install both the program and its dependencies on the user's computer:

```
python3 -m pip install ./ # Linux
py -m pip install .\      # Windows
```

We have also created a graphical installer for Windows that bundles FLARE and everything FLARE needs to run. It is illustrated in figure 2. Running this installer installs Python, FLARE, and its dependencies in a directory on the user's computer and puts an application shortcut in the Start menu.

## 2.3 Licensing

FLARE is proprietary non-commercial software developed in the context of a European Union's Horizon 2020 project. It is currently only available to the members of the project consortium.

## 2.4 Help

Extensive and comprehensive help is available throughout the program. This is visible in many of the screenshots in this article.

- Every screen in the program features a short explanatory text on a blue background. The text provides a quick description of the step of the FLARE process that the screen is about.
- Every screen also features a red *Guideline* button that opens the corresponding FLARE guideline. The FLARE guidelines are documents which comprehensively describe each step of the FLARE process.
- Every screen except the start screen features a gray *Help* button that opens a new window containing a shorter description of the process step, with a focus on the input that must be prepared or imported into the program at that point. These help texts are meant as a memo. They do not replace the corresponding guidelines.
- The start screen features a red *Tutorials* button that leads to video tutorials on the FLARE YouTube channel. A single video tutorial has been produced for now, but more will follow.

### 3. PROJECT DATABASES

The first step in using the FLARE program to follow the FLARE process is to create a project database.

Project databases, or projects for short, are files that contain information about a particular ship and its vulnerability assessment. In particular, project files contain references to the various files describing the ship's geometry, containing damage cases, and containing assessment results. They are central to using the FLARE program, because they record all the relevant information of a particular assessment.

To create a new project database, the user must click on the *Create* button in the program's start screen, illustrated in figure 1. The button opens up a dialog box for the user to enter a name for the project and where to save it on the computer.

Instead of creating a new project database, the user can of course open an existing one to continue working with it.

The file format of project databases is JSON. This is a very common, plain-text, open-standard file format for storing and exchanging data which is organized in lists of values and sets of name/value pairs, two universal data structures (Ecma International, 2017). This data format was chosen for its simplicity, ubiquity, and the fact that it is equally easy for machines to parse and for humans to read, thereby avoiding the pitfalls of more optimized, but also more complex and more opaque database formats.

### 4. INPUT OF SHIP GEOMETRY

The first stage in the FLARE process is to prepare the geometrical models of the ship to assess.

The models must be prepared with the software tools that the user normally uses for that task, as such preparation is out of the scope of the FLARE program. The models should then be imported into the program, as illustrated in figure 3, in the sense that references to the various files describing the ship's design information will be stored in the active project database.

FLARE currently does not use that information itself, but it is intended that it will use it in the future. The information should be saved in the project database anyway to record which geometry the assessment results relate to.

Some of the expected geometry information is three-dimensional (3D models), while the rest is tabular in nature. The program can record a reference to tabular information in any format, but it expects and will only display CSV tabular data. CSV stands for comma-separated values and is a very common, widely-supported, plain-text, open-standard file format for storing and exchanging tabular data (Shafranovich, 2005). It was chosen for its simplicity, ubiquity, and because it can be edited by hand.

In fact, the program expects not just geometry

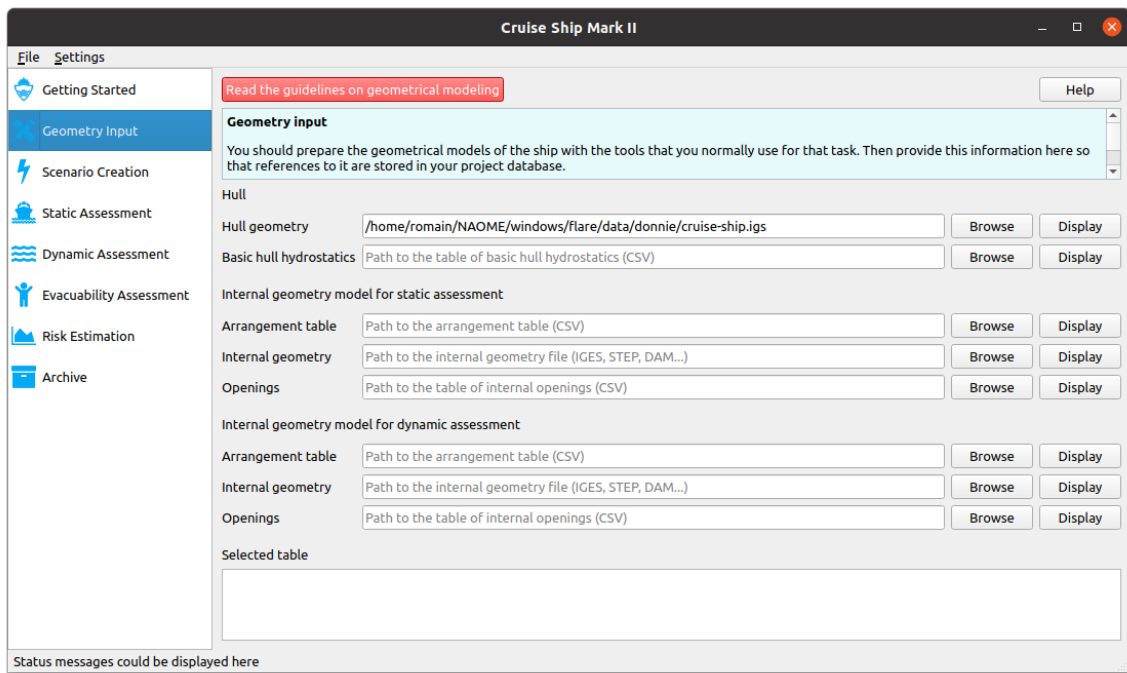


Figure 3: Input of ship design information into FLARE. Information that is tabular in nature is displayed at the bottom when the user clicks the corresponding *Display* button. Information that is three-dimensional is opened with the system’s default 3D application. We suggest using a CAD program for that: either a full-featured modeler such as Rhinoceros 3D or a lightweight viewer such as Open Cascade’s CAD Assistant.

information, but all tabular information to be provided in CSV format.

#### 4.1 Hull geometry

The user should provide a hull geometry file in the 3D format of their choice, for instance IGES or STEP. We recommend to also provide a table of basic hull hydrostatics to allow comparison between geometries created using different tools.

#### 4.2 Internal geometry

This is the geometry as prepared for statutory calculations in line with the probabilistic approach to damage stability assessment. The geometry model prepared for static assessment may be different from the model prepared for dynamic assessment.

In either case, static and dynamic, we recommend to provide the ship model’s arrangement table, which describes the properties of each

space such as volume and center of gravity, as indicated in table 1. Geometric information of the internal compartmentation, in the form of IGES or STEP surfaces describing the geometry of each space, should also be imported into the project.

#### 4.3 Openings

The user should also import all pertinent opening information into the project.

Ideally, the provided information should include values such as area, area ratio, collapse height, and leak height, in addition to those generally included in the table of openings for static stability assessment. However, if only a static assessment is being conducted, not all of this information will be necessary and, therefore, the information requirements here are dependent on the intended assessment. The information we suggest to provide is listed in tables 2 and 3.

Table 1: Suggested information to provide in the ship model's arrangement table for each room and internal space

Property	Type/Unit
Purpose of the room	text
Coordinates of the center of gravity $X_{CG}, Y_{CG}, Z_{CG}$	m
Permeability	non-dimensional
Volume	m <sup>3</sup>

Table 2: Suggested information to provide about each opening for static stability assessment

Property	Type/Unit
Deck the opening is located on	text
Opening type	text
Watertight rating	text
Frame location	text
Coordinates of the lower centroid of the opening	m
Rooms connected by the opening	text

Table 3: Suggested information to provide about each opening for dynamic stability assessment

Property	Type/Unit
Opening area	m <sup>2</sup>
Discharge coefficient of the opening	non-dimensional
Collapse height	m
Leak height	m
Area ratio (assumed percentage of the submerged opening area that leaks)	non-dimensional
Opening angle	deg
Door status	open/closed
Height of opening	m
Width of opening	m
Plane on which the opening is situated (transverse, longitudinal, horizontal)	non-dimensional



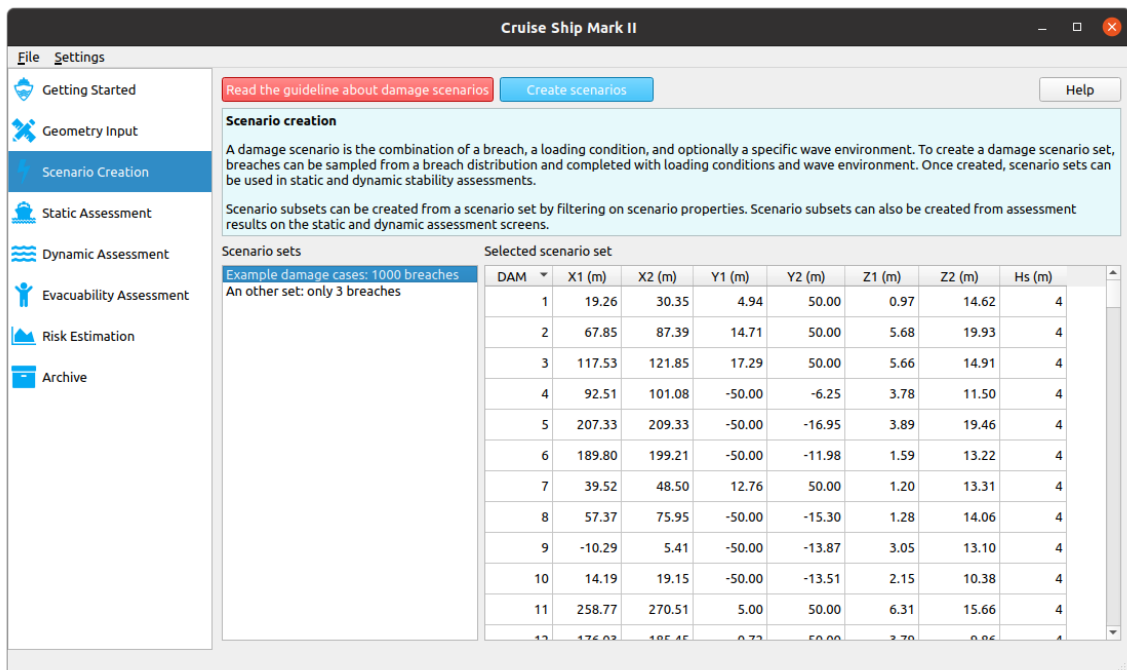


Figure 4: Input of damage information into FLARE. An arbitrary number of damage scenario sets can be loaded into the program and saved into the project database by clicking on the *Create scenarios* button. The selected set is displayed as a sortable table.

## 5. INPUT OF DAMAGE INFORMATION

The second stage in the FLARE process is to prepare the damage cases that will be investigated during the following stages of static stability, dynamic stability, and evacuability assessment. A damage case, or damage scenario, is the combination of a specific breach in the hull, a specific loading condition, and optionally a specific wave environment.

FLARE currently does not include a means to create damage scenarios from scratch, but it may provide this feature in the future, in the form of a breach generation tool that would sample from a specified breach distribution and a wave environment tool that would sample from a specified sea state distribution.

In the meantime, the framework requires the user to prepare damage scenarios outside of FLARE, as with the geometrical models of the ship in the previous stage. The user can sample a probabilistic distribution of breaches and complete the sample with loading conditions and wave environments, for instance. They can

use the software tools of their choice and the method of their choice for that task. A paper by Mauro et al. (2021) at this conference, for instance, describes a damage sampling method to reduce *A*-index standard deviation.

We suggest a specific format for the breaches, loading conditions, and optional wave environments that make up a set of damage cases. It is described below. If the user prepares their damage cases in the suggested format, the program will be able to read their scenario files, that is to say that the user will be able to import their scenario files into the program, display each scenario set as a table, and record a reference to the scenario files in the active project database, as illustrated in figure 4.

The rest of this section describes what information we suggest that a scenario set contains.

### 5.1 Breaches (damages)

The program currently expects breaches defined in the conventional “damage cube” format

and provided in CSV format, one breach per line, with the following suggested CSV headers: damage case ID,  $X_1$ ,  $X_2$ ,  $Y_1$ ,  $Y_2$ ,  $Z_1$ ,  $Z_2$  (visible in figure 4).

We intend to support breaches in non-dimensional EMSA-3 format in the future.

## 5.2 Loading conditions (initial conditions)

The program expects a loading condition provided by a single line of data in CSV format, with the following suggested information: initial condition ID, description of the initial condition, draft, trim, GM, KG, mass displacement.

The same loading condition applies to all the breaches and sea states.

## 5.3 Wave environment (sea state)

If the user intends to progress to dynamic assessment, they must provide environmental data for each breach. Their breach file must contain an additional CSV header for the significant wave height:  $H_s$  (visible in figure 4).

## 6. STATIC STABILITY ASSESSMENT

The stages that follow the preparation of the geometrical models of the ship to assess and the damage cases to evaluate are the backbone of the FLARE process: numerical assessments of the static damage stability, the dynamic damage stability, and the evacuability of the ship. Each of those three analyses has its own tab in the program's interface, as one can see in the screenshots.

FLARE is not meant to replace existing programs for static damage stability analysis, dynamic damage stability analysis, and evacuability analysis. Instead, the user should use such programs to perform the corresponding analysis. They should then record in FLARE which program they used, the damage scenarios they

assessed with it, and the assessment results it produced.

The rest of this section describes how the user is guided through conducting the static stability assessment by the three numbered buttons at the top of the corresponding screen illustrated in figure 5: *Configure tools*, *Run tool*, and *Import results*. Those are indeed the buttons that the user should click to record in FLARE the program that they will use for static stability assessment, select the damage cases that they will evaluate, and import the assessment results. The assessment results can then be displayed as a table and as a bar graph and filtered to create damage case subsets of interest.

Each of those buttons opens up a series of dialog windows that prompt the user for information such as the name and location of the assessment program, the set of damage cases to evaluate (chosen from the scenario sets defined on the previous screen), and the location of the assessment results (expected in CSV format).

### 6.1 Configure tools

The user should first click the *Configure tools* button to define the static damage stability programs that they have. The user needs to give a name and the path where each program that they intend to use is located. That information is saved into the active project database.

### 6.2 Run tool

The user should then click the *Run tools* button to specify which static damage stability program (among those defined in the previous step) they will use in their next assessment, and which set of damage cases (among those defined on the previous screen) they will evaluate. That information is saved into the active project database.

The user should then run their chosen program to perform the static stability calculations with their chosen damage cases. We suggest to calculate some or all of the properties indicated



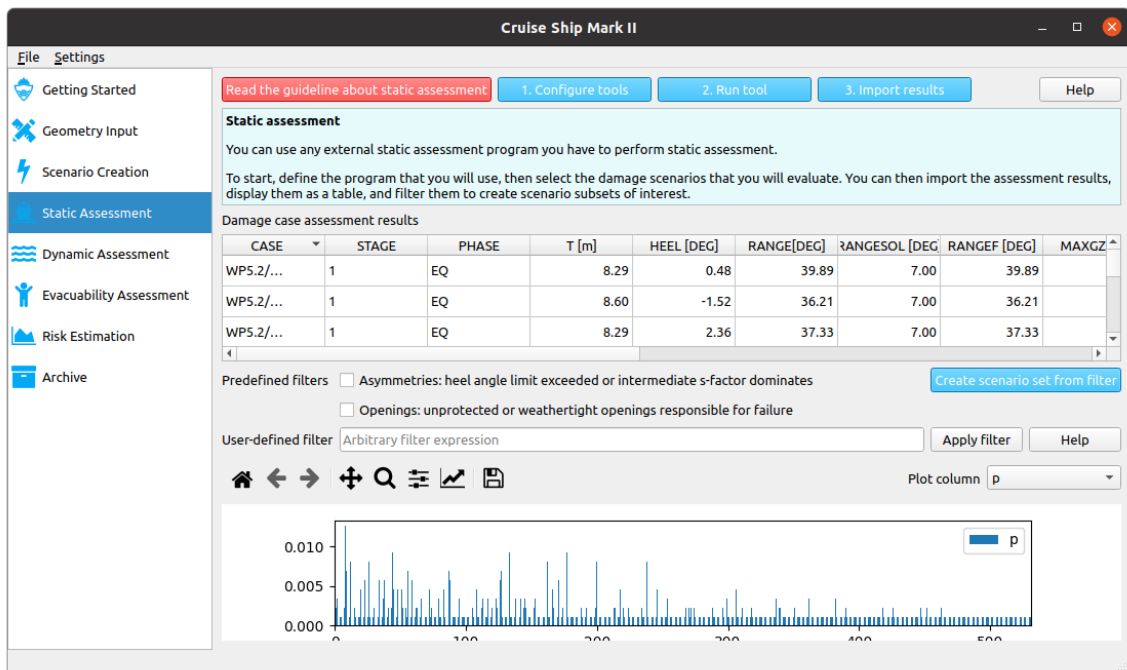


Figure 5: Static stability assessment screen, with assessment results loaded and displayed as a sortable table and as a bar graph. The bar graph displays the data in the column chosen in the drop-down menu.

in table 4 for each damage case. Those properties reflect the calculation parameters used in the SOLAS  $s$ -factor determination.

### 6.3 Import results

Once the static damage stability calculations are over, the assessment results can be imported into FLARE by clicking the *Import results* button. A reference to the file that contains the assessment results is saved into the active project database.

FLARE expects the assessment results to be in CSV format, one damage case per line. If the output of the assessment program is in a different form, it will need to be put into the expected format before being imported into FLARE. The complexity of this post-processing step that follows the assessment is dependent on the format of the output of the chosen assessment program.

We suggest that the CSV headers in the assessment result file are the same as the properties mentioned in table 4, plus a header for the damage case ID.

### 6.4 Filter results

Finally, FLARE provides filtering options to isolate damage cases of interest that are worth progressing to the following numerical analysis in the FLARE process, which is the dynamic assessment of the damage stability of the ship.

That next stage is indeed more computationally expensive than the static assessment. To make the whole process more time and resource efficient, it is advantageous to rule out damage cases where there is nothing to be gained from further investigation, and focus instead on the marginal cases.

The filtering options are currently in development. As one can see in figure 5, we intend to provide predefined filters to identify specific damage cases, and a text input field for the user to define a custom filter using a syntax such as “Heel > 30 and Max height of GZ curve > 0.5”. The feasibility of this last option remains under examination at the moment.

Table 4: Suggested outputs of the static stability assessment for each damage case

Property	Type/Unit
Flooding stage	text
Flooding phase	text
Draft in equilibrium condition	m
Equilibrium heeling angle	deg
Range of the greatest hump	deg
Range of the GZ curve as defined for the calculation of $s$ in chapter II-1, regulation 7	deg
Range of the greatest hump between equilibrium and angle of unprotected flooding	deg
Maximum height of GZ curve	m
Maximum righting lever $GZ_{max}$ as defined for the calculation of $s$ in ch. II-1, reg. 7	m
p-factor value	non-dimensional
s-factor value	non-dimensional
Heeling angle at which the first weathertight opening immerses	deg
Name of the weathertight opening first immersing	text
Heeling angle at which the first unprotected opening immerses	deg
Name of the unprotected opening first immersing	text
Minimum reserve to immersion of unprotected or weathertight openings at equilibrium	m

## 7. DYNAMIC STABILITY ASSESSMENT

The interface for the dynamic assessment of the damage stability of the ship, illustrated in figure 6, is identical to the interface for the previous stage of static stability assessment.

Likewise, the approach is similar. The user is guided through conducting the dynamic stability assessment by the three numbered buttons at the top of the screen, which they should click to record in FLARE the program that they will use for the assessment, select the damage cases that they will evaluate, and import the assessment results (expected in CSV format). At this stage, the selected damage cases should be a subset of the previously evaluated scenario set, created by filtering the static stability assessment results. As in the case of static assessment, the dynamic assessment results can be displayed as a table and as a bar graph and filtered further to create damage case subsets of interest.

### 7.1 Configure tools

This step is identical to the corresponding step on the static stability screen, with the only difference that it pertains to dynamic damage

stability programs of course.

### 7.2 Run tool

This step is also mostly identical to the corresponding step in static stability. As far as dynamic stability assessment results are concerned, we suggest to calculate at least the properties indicated in table 5 for each damage case.

### 7.3 Import results

Again, this step is mostly identical to the corresponding step in static stability. The assessment results are imported the same way and similarly subject to post-processing after the assessment if they are not already in CSV format.

We suggest that the CSV headers in the assessment result file are the same as the properties mentioned in table 5, plus a header for the damage case ID.

### 7.4 Filter results

Here too, FLARE provides filtering options to isolate damage cases of interest that are worth progressing to the last and most computationally

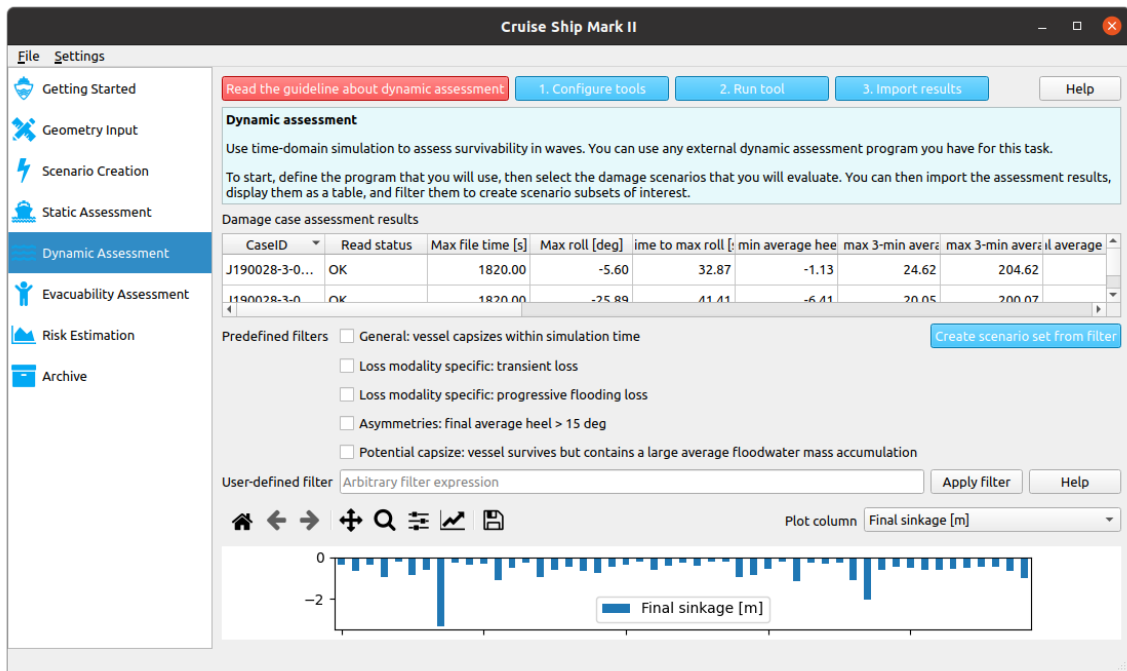


Figure 6: Dynamic stability assessment results screen, with assessment results loaded and displayed as a sortable table and as a bar graph. The bar graph displays the data in the column chosen in the drop-down menu.

Table 5: Suggested outputs of the dynamic stability assessment for each damage case

Property	Type/Unit
Maximum file time	s
Maximum roll	deg
Time to maximum roll	s
Maximum 3-minute average heel	deg
Time (1) to maximum 3-minute average heel	s
Time (2) to maximum 3-minute average heel	s
Final average heel	deg
Final heel rate	deg/h
Final average trim	deg
Final trim rate	deg/h
Final sinkage	m
Final sinkage rate	m/h
Maximum floodwater mass	t
Time to maximum floodwater mass	s
Final average floodwater mass (actual)	t
Final floodwater mass rate (actual)	t/h
Final average floodwater mass (envelope)	t
Final floodwater mass rate (envelope)	t/h
Immersion time of specific openings	list of times in s
Names of the corresponding openings	list of names

The (1) and (2) times are the start time and end time of the 3-minute time window the average is taken

expensive numerical analysis in the FLARE process, which is the assessment of the evacuability of the ship.

Again, we intend to provide predefined filters to identify those critical damage cases, and a text input field for the user to define a custom filter. These filtering options, which are critical steps in the FLARE process, are under development.

## 8. FUTURE WORK

As it is the last numerical analysis in the FLARE process, the evacuability assessment will be the next focus of our work on the FLARE program.

We then intend to implement the computation of key indicators of vulnerability, such as the attained subdivision index, the attained flooding risk index, the time to capsized, and the time to evacuate, along with other important parameters such as the critical openings and the distribution of loss modalities.

This final information constitutes a high-level summary of risks and risk features, as well as decision-support information for the user, on the basis of which they can select the most appropriate risk control options and modify the ship's design accordingly.

## 9. CONCLUSION

This paper has described a step-by-step walk-through in the FLARE program, and therefore also in the flooding risk assessment methodology that it implements. We started with the input of ship geometry and damage cases into the program and into the project database, and went through the three numerical analyses that form the pillars of the assessment.

This demonstrated the FLARE assessment process in a concrete manner and makes it clear that the program is an interactive guide which assists the user through the process: by using it to follow the process, the user conducts a comprehensive, quantitative, and progressively more

detailed evaluation of a ship's design on the basis of flooding risk.

## REFERENCES

- Caswell, T. A. et al., Jan. 2021. *matplotlib/matplotlib: REL: v3.3.4*. Software library. Version 3.3.4. DOI: [10.5281/zenodo.4475376](https://doi.org/10.5281/zenodo.4475376).
- Ecma International, Dec. 2017. *The JSON Data Interchange Syntax*. Standard ECMA-404. Version 2. Geneva, Switzerland. URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>.
- Hunter, J. D., 2007. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- Mauro, F., Paterson, D., Michalec, R., Boulougouris, E., and Vassalos, D., 2021. "A Damage Sampling Method to Reduce A-Index Standard Deviation in the Probabilistic Assessment of Ship Survivability While Using a Non-Zonal Approach". In: *First International Conference on the Stability and Safety of Ships and Ocean Vehicles. STAB&S 2021* (Glasgow, Scotland, UK, June 7–11, 2021). University of Strathclyde.
- McKinney, W., 2010. "Data Structures for Statistical Computing in Python". In: *Proceedings of the Ninth Python in Science Conference. SCIPY 2010* (Austin, Texas, USA, June 28–July 3, 2010). Ed. by S. van der Walt and J. Millman, pp. 56–61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).
- Reback, J. et al., Mar. 2021. *pandas-dev/pandas: Pandas 1.2.3*. Software library. Version 1.2.3. DOI: [10.5281/zenodo.4572994](https://doi.org/10.5281/zenodo.4572994).
- Shafranovich, Y., Oct. 2005. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180. DOI: [10.17487/RFC4180](https://doi.org/10.17487/RFC4180). URL: <https://datatracker.ietf.org/doc/rfc4180/>.

The Qt Company, Nov. 2020. *Qt for Python (PySide2)*. Software library. Version 5.15.2. URL: <https://www.qt.io/qt-for-python>.

Vassalos, D., Paterson, D., Boulougouris, E., and Mauro, F., 2021. “Realising Unsinkable Ships”. In: First International Conference on the Stability and Safety of Ships and Ocean Vehicles. STAB&S 2021 (Glasgow, Scotland, UK, June 7–11, 2021). University of Strathclyde.