Vessel.js
A web-based ship design library

Elias Hasle, M.Sc. student in Simulation & Visualization, and associated with
Ship Design and Operation Lab, NTNU in Ålesund.
Overview

• What is Vessel.js?
• Why Javascript?
• Demonstrations
• Paths forward
• Summary
What is Vessel.js?

• An open and collaborative ship design object-oriented library (*Gaspar, 2018*)
• A work in progress, living on [Github](http://github.com).
• Core classes that model sea-borne vessels with hull, parts and functionality, and perform calculations on the model.
What is Vessel.js?

• Ship specifications are in a straightforward and extensible JSON format.
• The main parts of the library are packaged in a single JS file without dependencies.
• The project aims to include a rich set of tools for manipulating, simulating and visualizing vessels.
Example calculations

• Input the outer hull surface as a table of transverse offsets, along with main dimensions and draft (how deep the ship sinks while floating).

• Output includes LCF, TCF, Awp, Ixwp, Iywp, BMt, BMI, Cwp, LWL, BWL, Ap, Cp, Vs, Cb, Cm, As, Cv, LCB and KB. (More than just a bunch of letters, if you ask a maritime engineer.)
Why Javascript?

• Web standard that lives «forever» through backwards-compatible browsers.
• Portable, open access and open source by default
  – Enables reproducible experiments and calculations, demonstrated by interactive web applications to a wider public.
• Huge developer community and software «ecosystem» to build upon.
Why Javascript?

• Performance. Really? Generally, yes!
• Javascript (1995) was slow a long time ago.
• Google’s V8 Javascript engine with JIT compilation (2008)
• The heaviest computations can be offloaded to WebGL (2011) and WebAssembly (2017), and web workers (2008?) can exploit multiple cores.
• The game industry helps drive technology.
• See Gaspar, 2017.
Demonstrations

- **Inspect ship in 3D** ¹
- **Orthographic deck views**
- **Multiple ships in configurable ocean** ²
- See more examples on [https://shiplab.github.io/vesseljs/examples/](https://shiplab.github.io/vesseljs/examples/) ³

---

¹: Block data by Killian Ledain, based on Ulstein vessel PX121. Hull data (not PX121) by Jefferson Flor.
²: Not physically based motion.
³: The examples use Three.js for graphics and dat.GUI for GUI.
Scientific calculations by Ícaro Fonseca

Input from an interactive web form (excerpt here)

Output is several graphs with simulation results

Part of Ícaro Fonseca’s Master project (Fonseca et al., 2018). Try the apps on https://icarofonseca.github.io/marsim/examples
Some possible paths forward

- More different hull parameterizations
- Specification of structural components
- Richer specification of functionality
- Standardized and better integration between model, simulations and visualization
- Visual editors for hull and objects.
- Advanced simulations with onboard equipment
- Automated design optimization
Conclusions

• Vessel.js aspires to become an open ship design platform.

• Javascript is great for science, engineering, education and communication.

• Follow web page on https://shiplab.github.io/vesseljs for new demonstrations.
References

• Gaspar, H. M.  

• Gaspar, H. M.  

• Fonseca, Í. A.; Gaspar, H. M.; Ryan, C. F.; Thomas, G. A.  
Contribute

• The project is hosted on https://github.com/shiplab/vesseljs

• Don’t hesitate to contact me or the project supervisor with questions or feedback.

Elias Hasle
Master student in Simulation & Visualization at the Department of ICT and Natural Sciences
E-mail: eliash@stud.ntnu.no, Phone: +47 90 17 11 96 (and I am here now)

Henrique Gaspar
Associate professor at the Department of Ocean Operations and Civil Engineering
E-mail: henrique.gaspar@ntnu.no
Phone: +47 70 16 12 47