Interactive Web Apps for Visualizations in Mathematics and Engineering

Francesca Taddei1,∗, Benjamin Rüth2, Christian Karpfinger3, and Gerhard Müller1

1 Chair of Structural Mechanics, Technical University of Munich, Arcisstr. 21, 80333 München
2 Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching
3 Research Group Algebra, Technical University of Munich, Boltzmannstr. 3, 85748 Garching

Depicting numbers in a graphical way has always been crucial to understand and explain the physical phenomena behind. Interactive graphical applications bring the understanding to a higher level, where the users get actively engaged. Moreover, interactive visualization of data, equations etc... help to motivate students and strengthen the sharing of knowledge. At the Technical University of Munich, the Chair of Structural Mechanics and the Research Group Algebra developed interactive applications in the fields of mechanics and higher mathematics. Students can access the applications online at any time, without installation or special requirements, and, thus, can actively explore the content with a very low entry barrier. For example, they can experiment with a spring-mass-damper system and let it enter the resonance by changing the frequency of the excitation; or they can search for local maxima and minima of a function with side conditions with the Lagrange multipliers in an interactive and graphical way. The interactive process enhances the understanding, even if the user does not interact with a strategy but just uses his intuition. These applications can be also used to explain complex concepts to the general public. The source code is publicly available on https://github.com/ChairOfStructuralMechanicsTUM/Mechanics_Apps.

© 2018 Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim

1 Introduction

The diversification of the media in teaching has a large impact on the quality and the outcome of the courses. In particular, several studies have shown the positive effects of interactive visualization in the learning process [1, 2]. Such visualization applications have been first used in the computer science education [3, 4], but also in several other disciplines e.g. mechanical [5] or biomedical [6] engineering. In most of the existing projects of interactive apps, the user has to download either a browser plugin or to download and execute an installation file. This can constitute a problem for many students who work from computer rooms at the university, due to usage restrictions, or simply, because they do not want to install anything on the computer. Moreover, these applications do not run on mobile devices, which can be used by the students during class for experimenting simultaneously with the lecturer or in other circumstances, e.g. while commuting by public transportation. On the contrary, up-to-date web browsers offer the possibility to run interactive applications, without additional installations and can be accessed from a smartphone, tablet, or personal computer regardless of the operating system. The only requirement is an internet connection. In this contribution we present the programming environment chosen at the Technical University of Munich for the development of interactive web applications for visualizations in mathematics and engineering.

2 Programming environment and examples

As programming framework, we use the open-source Python library, Bokeh [7]. Bokeh is an interactive visualization library that targets interactive plotting in modern web browsers. It has a concise approach to the construction of novel graphics with high-performance interactivity over very large or streaming datasets. The Bokeh library provides many essential GUI components for building interactive apps, such as sliders or buttons and well-known 2d plotting styles, including line, scatter plots and many more. Further features can be included with custom extensions. For example, JavaScript can be embedded using the package nodejs to create a 3d surface plot. The architecture of Bokeh is such that the code for creation of plots, ranges, axes, glyphs, etc. is written in Python and then converted to a format that is consumed by the client library, BokehJS. The application code runs on a web-server with a Bokeh installation. The translation from Python to the code that can be understood by the client browser is performed directly via the Bokeh framework and, as soon as the server is up and running, a Bokeh server uses the application code to create sessions and documents for all clients that connect using a browser. This means, that developers with a scientific background can write all code in Python and do not have to be familiar with web technology, while users can access the apps without having to understand the code behind. A wide range of visualization methods is possible in the proposed environment. This should not prevent a developer from keeping the application simple. Students will not use an application, if it takes them too long to understand what they are expected to do with the application. Moreover, the applications shall motivate the students to play around. Therefore, immediate interaction - without reading of instructions - is important. Schweitzer and Brown [3] have already described, how important the mouse-driven interaction (e.g. with sliders and buttons) is. The developer should prevent the user from using the keyboard.

∗ Corresponding author: e-mail francesca.taddei@tum.de, phone +0049 89 289 28326, fax +0049 89 289 28665
Each app is accompanied by a description and a short summary over the features of the app, e.g. which parameters can be changed. The apps are accessible via internet at the website of the the Chair of Structural Mechanics [8] and of the Research Group Algebra [9]. Additionally, links to the apps can also be included in the lecture notes, to facilitate access while studying. All source code of the application can be downloaded from Github [4].

Example 2.1 Figure 1 shows a screen-shot of the app for the visualization of the oscillation of a single-degree-of-freedom system. The app is structured in three main parts: On the left, the system with its spring, damper, and mass is shown. Below, the system properties can be modified by the user. The graphs show the deflection of the system (in time domain in the middle, in frequency domain with amplitude and phase on the right.). All presented elements are interactive. The system is animated and the plot in the middle is drawn in real time. Clicking on the buttons on the left starts, pauses, or stops the animation. Using the sliders on the bottom, all system properties like spring stiffness, damping ratio, or load frequency ratio can be modified. All changes here have an immediate effect on the response of the system. One can observe for example that a shift of the frequency ratio to one (to the eigenfrequency of the system) will immediately increase the response of the system.

Example 2.2 Figure 2 shows the direction field and the critical points and lines \((u(x, y) = v(x, y) = 0)\) of a system of ordinary differential equations (ODE). Furthermore, the solution curve is drawn for a given starting value \((x_0, y_0)\). The start value can be changed by a mouse click in the plot window. Other 2D ODE systems can be investigated by selecting a template or providing a symbolic representation of the system.

3 Summary

We presented a development environment for interactive web-based visualization applications in mathematics and engineering education. The applications are accessible from everywhere in the internet with an ordinary browser. The users do not need to perform installations on their devices. Some effort is necessary for the development and the maintenance of the applications and of the running server. Experiences about the frequency of usage and the feedback of students of are not yet available and will be collected during the next semesters.

Acknowledgements We thank the Study Quality Commission of the Faculty of Civil Engineering of the Technical University of Munich for funding this project.

References